Universal Synchronous Asynchronous Receive/Transmit USART

This section describes the serial communication interface USART. It has two functions implemented, to allow serial communication working in different ways. The first function is the well-known asynchronous communication protocol UART; the second function is the serial peripheral interface function SPI, which is also widely used. Even if all the hardware is used in common for both functions, it is described specifically for the function finally chosen, in the application environment which is normally defined to be UART or SPI. Nevertheless, with proper software and hardware design, both functions can be used, one after the other. One bit in the control register defines if the module operates as UART or SPI.

USART Peripheral Interface

The universal synchronous/asynchronous interface is a serial channel which allows a serial bit stream of 7 or 8 bits to be shifted into and out of the MSP430, at a programmed rate, or at a rate defined by an external clock. The USART peripheral interface is built to support, with one hardware configuration, two different serial protocols: the universal asynchronous protocol - often simply called RS232 - and the synchronous serial protocol - usually known as the SPI protocol.

The control bit SYNC in control register UCTL is used to select the required mode:

SYNC = 0: asynchronous - UART - mode selected
SYNC = 1: synchronous - SPI - mode selected.

The USART is connected to the CPU as a byte peripheral module. It connects the controller to the external system environment by three or four external pins.

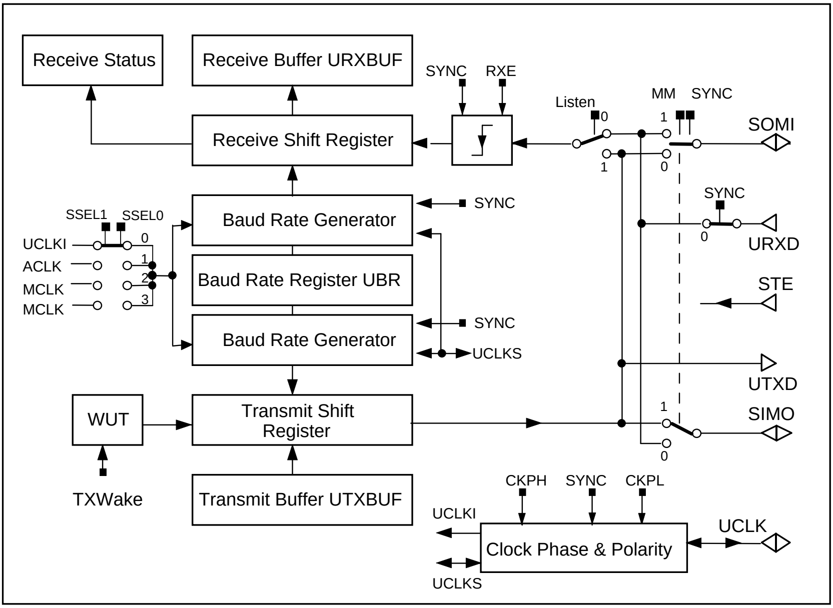


Figure 12.1: Block diagram of USART

12 USART Peripheral Interface, UART Mode

The universal synchronous/asynchronous interface is a serial channel which allows a serial bit stream of 7 or 8 bits to be shifted into and out of the MSP430 at a programmed rate. The asynchronous mode is selected when the control bit SYNC in the USART control register UCTL is reset. The USART is connected to the CPU as a byte peripheral. It connects the controller to the external system environment by three external pins.

USART's serial asynchronous communication feature:

- Asynchronous modes, including Idle line/Address bit communication protocols
- Two shift registers shift serial data stream into URXD, and out on UTXD
- Data transmitted/received with LSB first
- Programmable transmit and receive bit rate
- Status flags

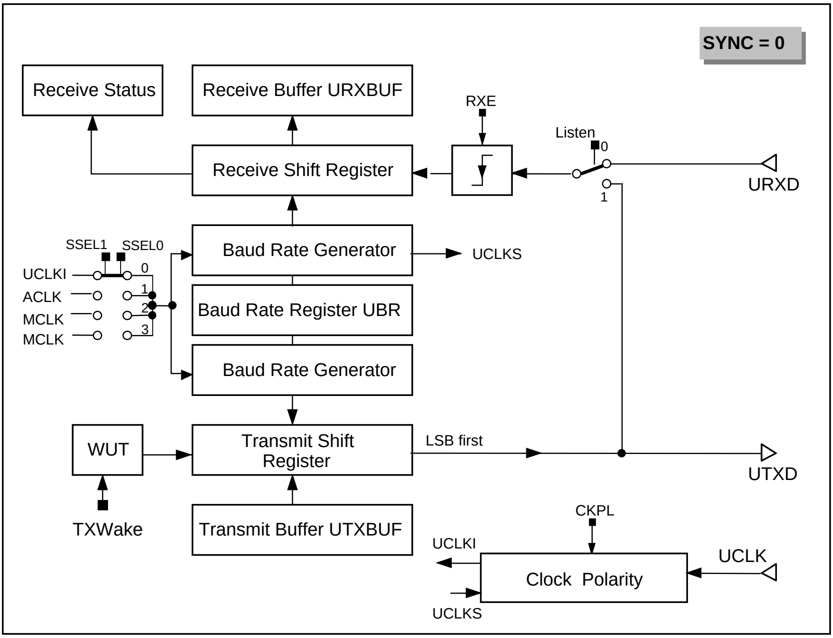


Figure 12.1: Block diagram of USART - UART mode

12.1 Asynchronous Operation

In the asynchronous mode, the receiver synchronizes itself to frames, but the external transmitting and receiving devices do not use the same clock source; the baud rate is generated locally.

12.1.1 Asynchronous Frame Format

The asynchronous frame format consists of a start bit, seven or eight data bits, even/odd/no parity bit, an address bit in Address bit mode, and one or two stop bits. The bit period is defined by the selected clock source and the data in the baud rate registers.

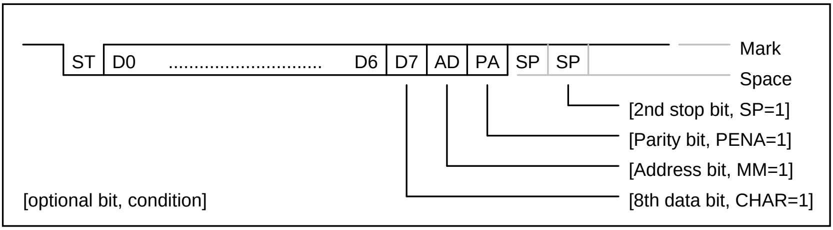
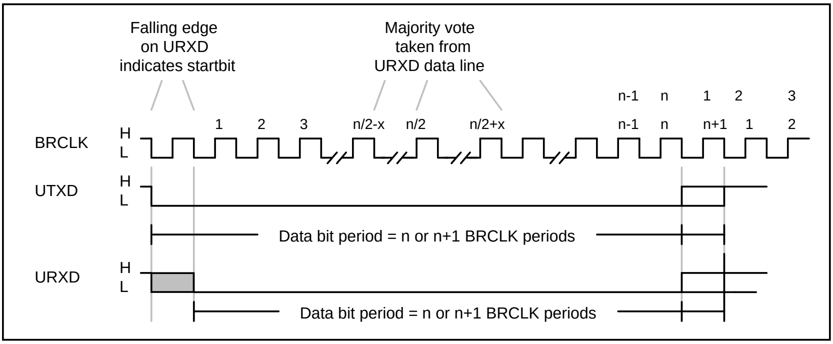


Figure 12.2: Asynchronous frame format

The receive (RX) operation is initiated by the receipt of a valid start bit. It consists of a negative edge at URXD, followed by the taking of a majority vote from three samples, where 2 of the samples must be zero. These samples occur at n/2-x, n/2 and n/2+x of BRCLK periods after the negative edge. This sequence provides false start bit rejection, and also locates the center of bits in the frame, where the bits will be read on a majority basis. The timing of x is $^1/_{32}$ to $^1/_{63}$ times of BRCLK, but at least BRCLK, depending on the division rate of the baud rate generator.



Falling edge on URXD indicates startbit
Majority vote taken from URXD data line
n-1
n
1
2
3
BRCLK
H
L
1
2
3
n/2-x
n/2
n/2+x
n-1
n
n+1
1
2
UTXD
H
L
Data bit period = n or n+1 BRCLK periods
URXD
H
L
Data bit period = n or n+1 BRCLK periods

Figure 12.3: Asynchronous bit format. Example for n or n+1 clock periods

12.1.2 Baud rate generation in asynchronous communication format

The baud rate generation in the MSP430 differs from other standard serial communication interface implementations.

Standard Baud Rate Generation
The standard implementation uses a prescaler from any clock source and a fixed second clock divider which is usually a divide by 16.

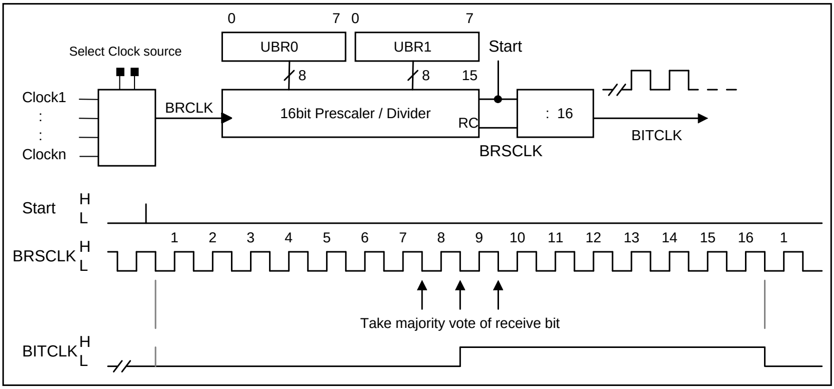


Figure 12.4: Standard baudrate generation - other than MSP430

Baudrate = **Error!**

Using this common scheme to generate the baud rate can not generate baud rates that are chosen close to the frequency of the prescaler's input frequency BRCLK. Division factors of e.g. 18 are not possible, as well as non-integer factors - for example 13.67.

Example 1
Assuming a clock frequency of 32,768Hz for the BRCLK signal, and a required baudrate of 4800 Baud, the division factor is 6.83. In a standard baud rate generator the minimum factor is 16 - the crystal's frequency and the baud rate generation can not meet the requirements.

Example 2
Assuming a clock frequency of 1.04MHz (32 x 32,768Hz) for BRCLK signal and a required baudrate of 19 200 Baud, the division factor is 54.61. In a standard baud rate generator the next factors are 48 (3x16) or 64 (4x16) - the crystal's frequency and the baud rate generation can not meet the requirements. The crystal frequency needs to be selected to meet the communication requirements. Other criteria like current consumption, simple real-time clock function or system cost constraints can not be considered to be favorable.

MSP430 Baud Rate Generation

The baud rate generator of the MSP430 uses one prescaler/divider and a modulator. This combination is used to work properly with crystals whose frequency is not a multiple of the standard baud rates, but allows the protocol to run at maximum baud rate. Using this technique, even with a watch crystal (32,768Hz) baudrates up to 4800 (9600) baud are possible. This gives power advantages, since the selection of sophisticated MSP430 operation in low power mode is possible.

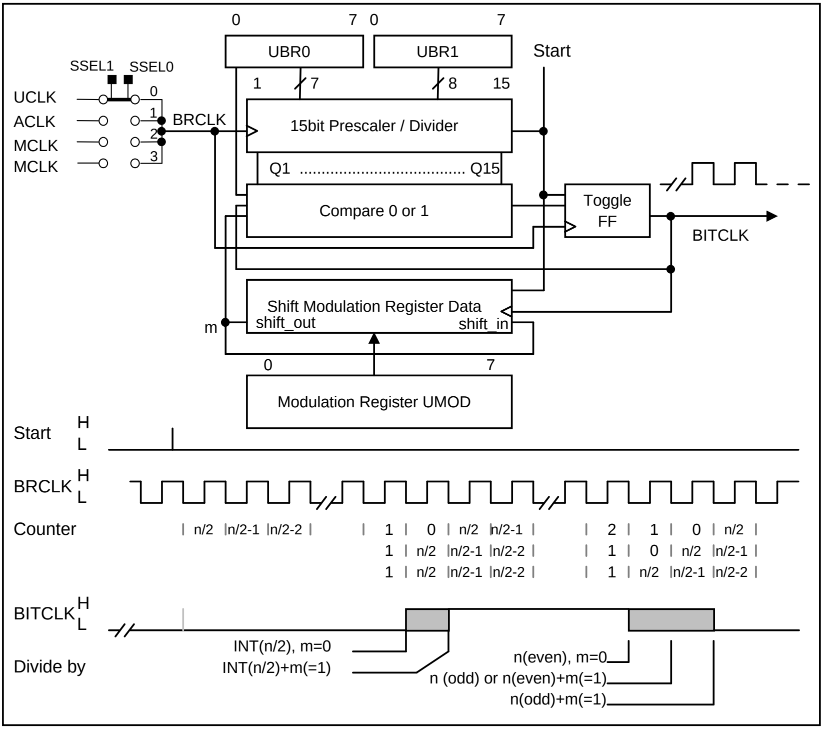


Figure 12.5: MSP430 Baud Rate Generation. Example for n or n+1 clock periods

The LSB of the modulation register is used first for modulation - it starts with the start bit. A set modulation bit increases the division factor by one.

Example 1

Assuming a clock frequency of 32,768Hz for BRCLK signal and a required baudrate of 4800 Baud, the division factor is 6.83. The baud rate generation in the MSP430's USART uses a factor of 6 plus the modulation register loaded with 6Fh (0110 1111). This means the divider runs the following sequence: 7 - 7 - 7 - 7 - 6 - 7 - 7 -6 - .... The sequence repeats after all eight bits of the modulator are used.

Example 2
Assuming a clock frequency of 1.04MHz (32 x 32,768Hz) for BRCLK signal, and a required baudrate of 19 200 Baud, the division factor is 54.61 The baud rate generation in the MSP430’s USART uses a factor of 54 (36h) plus the modulation register loaded with 0D5h. This means the divider runs the following sequence: 55 - 54 - 55 - 54 - 55 - 54 - 55 -55 - .... The sequence repeats after all eight bits of the modulator are used.

The standard baud rate data needed for the baud rate registers and the modulation register are listed for the watch crystal 32,768Hz (ACLK) and MCLK, assumed to be 32 times the ACLK frequency. The error listed is calculated for the receive path. In addition to this error, the synchronization error should also be considered.

Baud rate	Divide by		ACLK			max. error %	MCLK (= 32 x ACLK)			max. error %
	ACLK	MCLK	UBR1	UBR0	UMOD		UBR1	UBR0	UMOD	
75	436.91	13981	1	B4	FF	-.1/.3	36	9D	FF	0/.1
110	297.89	9532.51	1	29	FF	0/.5	25	3C	FF	0/.1
150	218.45	6990.5	0	DA	55	0/.4	1B	4E	FF	0/.1
300	109.23	3495.25	0	6D	22	-.3/.7	0D	A7	00	-.1/0
600	54.61	1747.63	0	36	D5	-1/1	06	D3	FF	0/.3
1200	27.31	873.81	0	1B	03	-4/3	03	69	FF	0/.3
2400	13.65	436.91	0	0D	6B	-6/3	01	B4	FF	0/.3
4800	6.83	218.45	0	06	6F	-9/11	0	DA	55	0/.4
9600	3.41	109.23	0	03	4A	-21/12	0	6D	03	-.4/1
19 200		54.61					0	36	6B	-.2/2
38 400		27.31					0	1B	03	-4/3
76 800		13.65					0	0D	6B	-6/3
115 200		9.10					0	09	08	-5/7

Table 12.1: Commonly used Baud Rates, Baudrate data and errorsCommonly

The maximum error is calculated for the receive mode and the transmit mode. The error in the receive mode is the accumulating timing error versus the ideal scanning time in the middle of each bit. The transmit error is the accumulating timing error versus the ideal time of the bit period.

The maximum frequency of MCLK is noted in the device data sheet and can exceed the example frequency.

12.1.3 Asynchronous Communication Formats

The USART module supports two multiprocessor communication modes when the asynchronous mode is used. These formats can be used to transfer information between many microcomputers on the same serial link. Information is transferred as a block of frames from a particular source to one or more destinations. The USART has features to identify the start of blocks, and to suppress interrupts and status information from the receiver, until a block start is identified. In both multiprocessor modes, the sequence of data exchange with the USART module could be based on polling of data, or using the receive interrupt features.

Both asynchronous multiprocessor protocols, the idle line and the address bit multiprocessor mode allow efficient data transfer between multiple communication systems. They also can be used to minimize activity of the system, whether to save current consumption or processing resources. The MM bit in the control register defines the address bit or idle line multiprocessor protocol mode. Both formats use the wake up on transmitting, using the address feature function (TXWake bit), and on activating the RXWake bit. The URXWIE and URXIE bits control the transmit and receive features of these modes.

12.1.4 Idle line multiprocessor mode

In this mode, blocks of data are separated by an idle time between them. An idle receive line is detected when 10 or more 1s in a row are received after the first stop bit of a character.

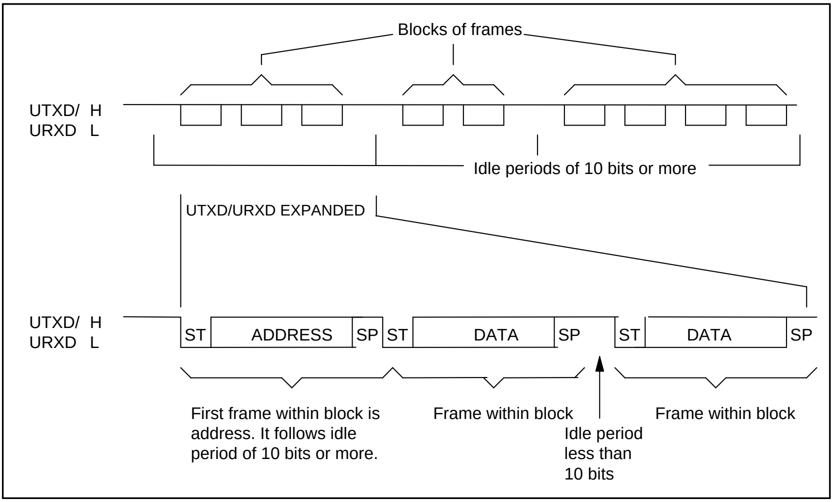

Blocks of frames
UTXD/ H
URXD L
Idle periods of 10 bits or more
UTXD/URXD EXPANDED
UTXD/ H
URXD L
ST
ADDRESS
SP
ST
DATA
SP
ST
DATA
SP
First frame within block is address. It follows idle period of 10 bits or more.
Frame within block
Idle period less than 10 bits
Frame within block

Figure 12.6: Idle line multiprocessor protocol

When two stop bits are used, the second one is counted as the first 'Mark' bit of the idle period. The first character received after an idle period is an address character. The RXWake bit can be used as an address tag for the character. In idle line multiprocessor format, RXWake bit is set when a received character is an address character and is transferred into the receive buffer.

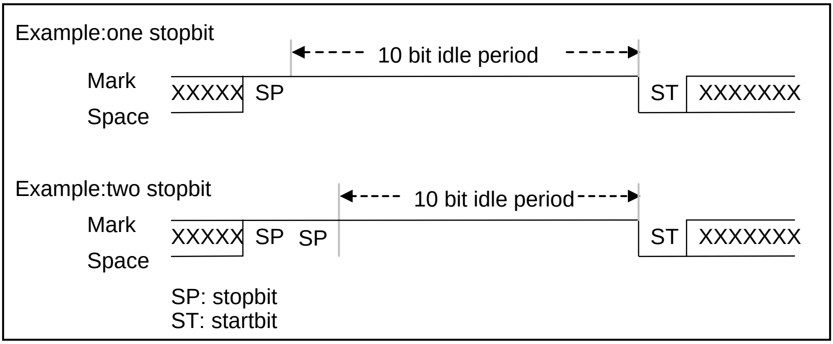


Figure 12.7: USART Receiver Idle Detect

Normally, if the USART’s URXWIE bit in the receive control register is set, characters will be assembled as usual by the receiver, but they will not be transferred to the receiver buffer, URXBUF, nor will interrupts be generated. When an address character is received, the receiver is temporarily activated to transfer the character to URXBUF and set the URXIFG interrupt flag. Appropriate error status flags will be set. The application software can validate the received address. If there is a match, the application software will handle the further data processing and execute proper operation. If not, the processor waits for the next address character to arrive. The URXWIE bit itself is not modified by the USART: it should be modified by the user in order to receive non-address characters or address characters.

In idle line multiprocessor mode, a precise idle period can be generated to create efficient address character identifiers. Associated with the TXWake bit is the wake-up temporary (WUT) flag. WUT is an internal flag, double buffered with TXWake. When the transmitter is loaded from UTXBUF, WUT is loaded from TXWake, and TXWake bit is reset.

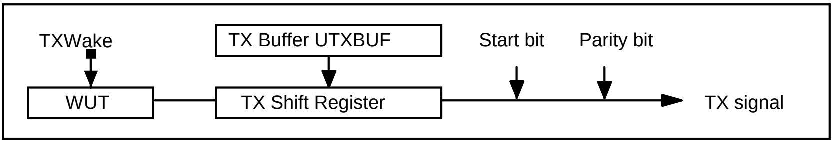


Figure 12.8: Double-Buffered WUT and TX Shift Register

Sending out an idle frame to identify an address character is accomplished as follows:

The TXWake bit should be set, and then any word (don't care) must be written to the UTXBUF (UTXIFG should be set). When the transmitter shift register is empty (TXEPT is set), the contents of the UTXBUF are shifted to the transmit shift register, and the TXWake value is shifted to WUT. When the WUT bit has been set, the start, data, and parity bits will be suppressed and an idle period of exactly 11 bits will be transmitted. The next data word, shifted out of the serial port after the address character identifying idle period, will be the second word written to the UTXBUF after TXWake bit was set. The first data word written is suppressed while the address identifier is sent out, and ignored after that. Writing the first don't care word to UTXBUF is necessary so that the TXWake bit value can be shifted to WUT.

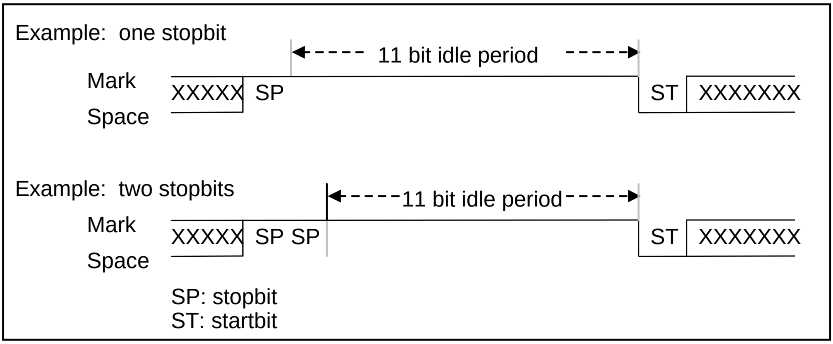


Figure 12.9: USART Transmitter Idle Generation

12.1.5 Address bit Format

In this mode, characters contain an extra bit that is used as an address indicator. The first character in a block of data carries an address bit that is set to indicate that the character is an address. The RXWake bit is set when a received character is an address character, and is transferred into the receive buffer (receive conditions are true).

Normally, if the USART's URXWIE bit is set, data characters will be assembled as usual by the receiver, but they will not be transferred to the receiver buffer URXBUF nor will interrupts be generated. When a character is received that has an address bit set, the receiver is temporarily activated to transfer the character to URXBUF and set the URXIFG. Error status flags will be set as appropriate. The application SW handles the succeeding operation for the best benefit in processing resource handling or current consumption reduction. The application software can validate the received address. If there is a match, the processor can read the remainder of the data block. If not, the processor waits for the next address character to arrive.

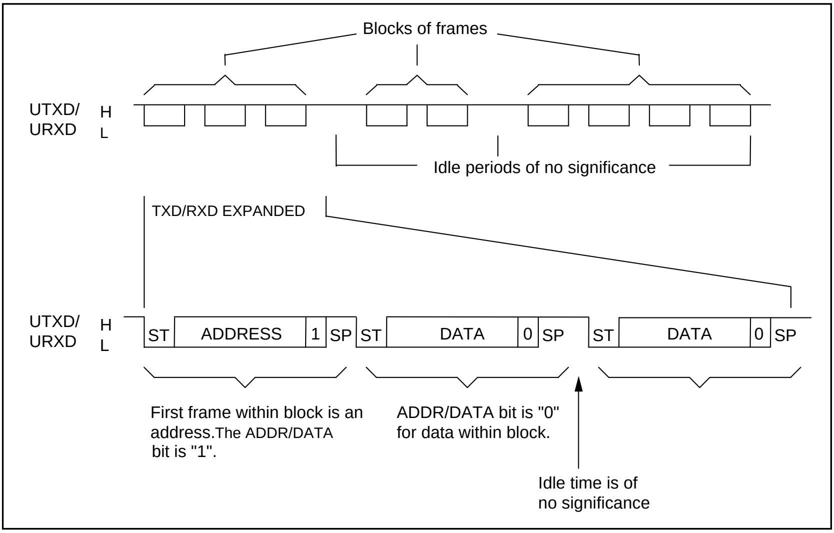

Blocks of frames
UTXD/
URXD
H
L
Idle periods of no significance
TXD/RXD EXPANDED
UTXD/
URXD
H
L
ST
ADDRESS
1
SP
ST
DATA
0
SP
ST
DATA
0
SP
First frame within block is an address.The ADDR/DATA bit is "1".
ADDR/DATA bit is "0" for data within block.
Idle time is of no significance

Figure 12.10: Address bit multiprocessor protocol

In address bit multiprocessor mode, the address bit of a character can be controlled by writing to the TXWake bit. The value of the TXWake bit is loaded into the address bit of that character each time a character is transferred from the transmit buffer UTXBUF to the transmitter. The TXWake bit is then cleared by the USART.

12.2 Interrupt and Control Function

The USART peripheral serves two main interrupt sources, for transmission and reception. Two individual interrupt vectors are available, one for receive and one for transmit interrupt events.

The control bits of the USART are located in the SFR address range:

•	Receive Interrupt Flag	URXIFG	initial state reset (by PUC/SWRST)
•	Receive Interrupt Enable	URXIE	initial state reset (by PUC/SWRST)
•	Receive Enable	URXE	initial state reset (by PUC)
•	Transmit Interrupt Flag	UTXIFG	initial state set (by PUC/SWRST)
•	Transmit Interrupt Enable	UTXIE	initial state reset (by PUC/SWRST)
•	Transmit Enable	UTXE	initial state reset (by PUC)

The receiver and transmitter of the USART operate fully independently, but use the same baud rate generator. Transmit and receive use the same baud rate.

12.2.1 USART Receive Enable

The Receiver Enable bit URXE enables or disables the receiver from collecting the bit stream on the URXD data line. Disabling the USART receiver will stop the receive operation after completing a receive operation which has been started, or stop immediately if no receive operation is active. The start bit detection is disabled.

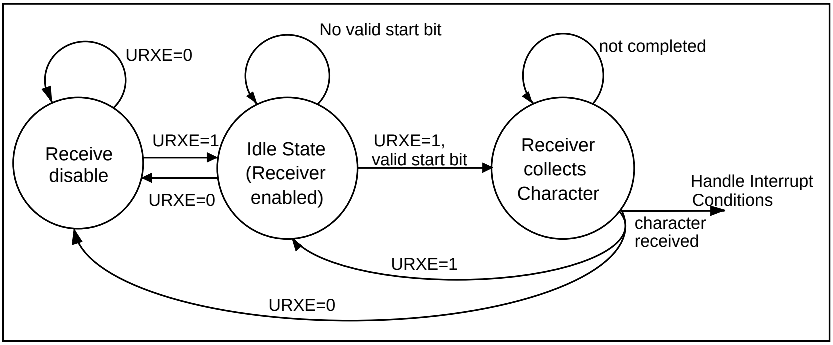


Figure 12.11: State diagram on Receiver enable URXE

Note: URXE re-enable, UART Mode

Since the receiver is completely disabled a re-enable of the receiver is asynchronous to any data stream on the communication line. Synchronization can be done by looking for an idle line condition before accepting any received character.

12.2.2 USART Transmit Enable

The transmit enable bit UTXE enables or disables a character transmission on the serial data line. If this bit is reset, the transmitter is disabled but any active transmission is not halted until all data previously written into the transmit buffer has been sent. If the transmission is completed, any further write to the transmitter buffer will not result in a data transmission.

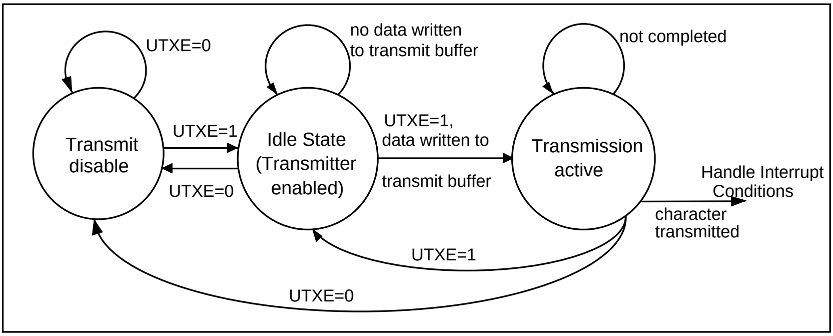


Figure 12.12: State diagram on Transmitter enable

When UTXE is reset any data can be written regularly into the transmit buffer, but no transmission is started. Once the UTXE bit is set, an immediate start of transmission of the character presently in the buffer is initiated. This character is transmitted correctly.

Note: Write to UTXBUF, UART Mode

Data should never be written into the transmit buffer UTXBUF when it is not ready and the transmitter is enabled (UTXE is set). If it is, the character shifted out can be random.

12.2.3 USART Receive Interrupt Operation

The receive interrupt flag URXIFG is set or is unchanged each time a character is received and loaded into the receive buffer:

- Erroneous characters (parity, frame or break error) will not set interrupt flag URXIFG when URXEIE is reset: URXIFG is unchanged.
- All type of characters (URXWIE=0) or only address characters (URXWIE=1) will set the interrupt flag URXIFG pending on the bit URXWIE. When URXEIE is also set, erroneous character will set the interrupt flag URXIFG.

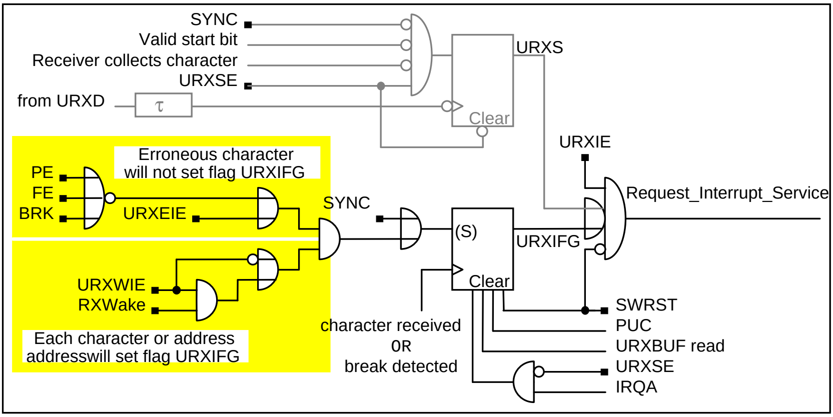


Figure 12.13: Receive Interrupt Conditions

URXIFG is reset at system reset PUC, or at a software reset SWRST. URXIFG is reset automatically if the interrupt is served (URXSE=0) or the receive buffer URXBUF is read. The Receive Interrupt Flag URXIFG indicates, if set, an interrupt event waiting to be served. The Receive Interrupt Enable bit URXIE enables, if set, serving of a waiting interrupt request. Both the receive interrupt flag URXIFG and the receive interrupt enable bit URXIE are reset with PUC and SWRST.

The signal URXIFG can be accessed by software. Signal URXS can not be accessed by software. When both interrupt events - receive start detection and character receive action - are enabled by software, the flag URXIFG indicates that a character was received and not the start detect request interrupt service. This works, since the interrupt software handler for the receive start detection will reset the URXSE bit. This clears the URXS bit and prevents further interrupt requests from URXS. The URXIFG should be already reset since no set condition was at this time at URXIFG latch.

12.2.4 USART Transmit Interrupt Operation

The transmit interrupt flag UTXIFG is set by the transmitter to indicate that the transmitter buffer UTXBUF is ready to accept another character. This bit is automatically reset if the interrupt request service is started or a character is written into the UTXBUF. This flag will assert a transmitter interrupt if the local (UTXIE) and general (GIE) interrupt enable bit are set. The UTXIFG is set after system reset PUC or SWRST are removed.

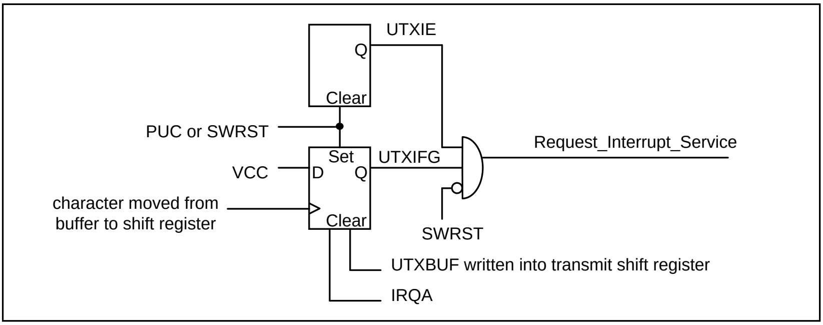

UTXIE
Q
Clear
PUC or SWRST
Set
UTXIFG
Request_Interrupt_Service
VCC
D
Q
character moved from buffer to shift register
Clear
SWRST
UTXBUF written into transmit shift register
IRQA

Figure 12.14: Transmit Interrupt Condition

The transmit interrupt enable UTXIE bit controls the ability of the UTXIFG to request an interrupt but does not prevent the flag UTXIFG from being set. The UTXIE is reset with PUC or software reset bit SWRST. The UTXIFG bit is set after system reset PUC or software reset SWRST, but the UTXIE bit is reset to ensure full interrupt control capability.

12.3 Control and Status Register

The USART module hardware is byte structured and should be accessed by byte processing instructions (suffix 'B').

Register	short form	Register type	Address	Initial state
● USART Control register	UCTL	Type of read/write	070h	See ....
● Transmit Control register	UTCTL	Type of read/write	071h	individual ...
● Receive Control register	URCTL	Type of read/write	072h	bit description
● Modulation Control reg.	UMCTL	Type of read/write	073h	unchanged
● Baud Rate register 0	UBR0	Type of read/write	074h	unchanged
● Baud Rate register 1	UBR1	Type of read/write	075h	unchanged
● Receive Buffer	URXBUF	Type of read/write	076h	unchanged
● Transmit Buffer	UTXBUF	Type of read	077h	unchanged

All bits are random after PUC, unless noted otherwise by the detailed functional description.
Reset of the USART is performed by PUC or SWRST bit. After power-up clear (PUC) the SWRST bit remains set and the USART remains in this condition until the reset is disabled by resetting the SWRST bit.
The USART module operates in asynchronous or in synchronous mode defined by the SYNC bit. The bits in the control registers may have different functions in the two modes. All bits in this section are described with their functions in the asynchronous mode - SYNC=0. Their functions in the synchronous mode are described in the USART's serial peripheral interface section.

12.3.1 USART Control register UCTL

The information stored in the control register determines the basic operation of the USART module. The register bits select the communications protocol, communication mode and parity bit. All bits should be programmed according to the selected mode before reset is disabled by resetting bit SWRST.

	7							0
UCTL 070h	PENA	PEV	SP	CHAR	Listen	SYNC	MM	SWRST
	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-1

Figure 12.15: USART Control Register UCTL

Bit 0: The USART state machines and operating flags are initialized to the reset condition if the software reset bit is set. Until the SWRST bit is reset, all affected logic is held in the reset state. This implies that after a system reset the USART must be re-enabled by resetting this bit. The receive and transmit enable flags URXE and UTXE are not altered by SWRST.

Bit 1: Multiprocessor mode (address/idle line wake up).
Two multiprocessor protocols, idle line and address bit, are supported by the USART module. The choice of multiprocessor mode affects the operation of the automatic address decoding functions.
MM = "0" : Idle line multiprocessor protocol
MM = "1" : Address bit multiprocessor protocol
The conventional asynchronous protocol uses MM bit reset

Bit 2: Mode or function of USART module selected.
The SYNC bit selects the function of the USART peripheral interface module. Some of the USART control bits will have different functions in UART and SPI mode.
SYNC = 0 : UART function is selected.
SYNC = 1 : SPI function is selected.

Bit 3: The Listen bit selects if the transmitted data is fed back internally to the receiver.
Listen = 0 : No feed back.
Listen = 1 : Transmit signal is internally fed back to the receiver. Each transmission from the MSP430’s USART is received parallel and no external signal is received anymore.

Bit 4: Character length.
This register bit selects the length of the character to be transmitted as 7 or 8 bits. Characters of 7 bits do not use the eighth bit in URXBUF and UTXBUF and this bit is padded with "0".
CHAR = 0 : 7 bit data.
CHAR = 1 : 8 bit data.

Bit 5: Number of stop bits.
This bit determines the number of stop bits transmitted. The receiver checks for one stop bit only.
SP = 0 : one stop bit.
SP = 1 : two stop bits.

Bit 6: Parity odd/even.
If PENA bit is set (parity bit is enabled), the PEV bit defines odd or even parity according to the number of odd or even "1" bits in both transmitted and received characters, address bit (address bit multiprocessor mode) and parity bit.
PEV = 0 : Odd parity
PEV = 1 : Even parity.

Bit 7: Parity enable.
If parity is disabled no parity bit is generated during transmission or expected during reception. A received parity bit is not transferred to the URXBUF with the received data as it is not considered as one of the data bits. During address bit multiprocessor mode, the address bit is included in the parity calculation.
PEN = 0 : Parity disable
PEN = 1 : Parity enable

> **Note: MARK, SPACE definition**
>
> The MARK condition is identically to the signal level in the idle state. SPACE is the opposite signal level: the start bit is always SPACE.

12.3.2 Transmit Control Register UTCTL

The register UTCTL controls the USART hardware associated with transmit operation.

	7							0
UTCTL 071h	unused	CKPL	SSEL1	SSEL0	URXSE	TXWake	unused	TXEPT
	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-1

Figure 12.16: USART Transmitter Control Register

Bit 0: The transmitter empty TXEPT flag is set when the transmitter shift register and UTXBUF are empty, and reset when data is written to UTXBUF. It is set on SWRST.

Bit 1: unused

Bit 2: The TXWake bit is used to control the transmit features of the multiprocessor communication modes. Each transmission - started by loading the UTXBUF - uses the state of the TXWake bit to initialize the feature of address identification. It should not be cleared - the USART hardware clears this bit once it has been transferred to "Wake Up Temporary", WUT; SWRST also clears TXWake bit.

Bit 3: The receive start edge control bit requests - if set - a receive interrupt service. For a successful interrupt service the corresponding enable bits URXIE and GIE should be set. The advantage of this bit is to start the controller’s clock system including MCLK along with the interrupt service, and keep it running by modifying the mode control bits. The USART is working with selected MCLK properly, even if the system is switched to a low power mode with disabled MCLK.

Bit 4,5: Source Select 0 and 1.
The source select bit defines which clock source is used for the baud rate generation:

SSEL1,SSEL0	0	external clock selected, UCLKI
	1	auxiliary clock selected, ACLK
	2, 3	main system clock selected, MCLK

Bit 6: Clock polarity CKPL.
The CKPL bit controls the polarity of the UCLKI signal.
CKPL = 0: the UCLKI signal has same polarity than UCLK signal.
CKPL = 1: the UCLKI signal has inverted polarity of UCLK signal.

Bit 7: Unused

12.3.3 Receive Control Register URCTL

The register URCTL controls the USART hardware associated with the receiver operation and holds error and wakeup conditions modified by the latest character written to the receive buffer URXBUF. Once any of the bits FE, PE, OE, BRK, RXERR or RXWake is set, they are not reset by receiving another character. They are reset by accessing the receive buffer URXBUF, by a USART SW reset SWRST, a system reset PUC or by instruction.

	7							0
URCTL 072h	FE	PE	OE	BRK	URXEIE	URXWIE	RXWake	RXERR
	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0

Figure 12.17: USART Receiver Control Register

Bit 0: The receive error bit RXERR indicates that one or more error flags (FE, PE, OE or BRK) are set. It is not reset when the error bits are cleared by instruction.

Bit 1: Receiver Wake-up Detect
RXWake bit is set when a received character is an address character and is transferred into the receive buffer.
Address bit multiprocessor mode: RXWake is set when the address bit is set in the character received.
Idle line multiprocessor mode: RXWake is set if an idle URXD line was detected (11 bits of Mark level) in front of the received character.
RXWake is reset by accessing the receive buffer URXBUF, by a USART SW reset SWRST or a system reset PUC.

Bit 2: The receive wake-up interrupt enable bit URXWIE selects the type of character that will set the interrupt flag URXIFG:
URXWIE=0: each character received will set the URXIFG
URXWIE=1: only characters that are marked as address characters will set the interrupt flag URXIFG. It operates identically in both multiprocessor modes.
The wake-up interrupt enable feature depends on the receive erroneous character feature. See also URXEIE bit.

Bit 3: The receive erroneous character interrupt enable bit URXEIE selects if an erroneous character will set the interrupt flag URXIFG.
URXEIE=0: each erroneous character received will not alter the interrupt flag URXIFG
URXEIE=1: all characters can set the interrupt flag URXIFG depending on the conditions set by URXWIE bit.

URXEIE	URXWIE	Char. w/ Error	Char. address	Description Flag URXIFG after a character was received
0	x	1	x	unchanged
0	0	0	x	set
0	1	0	0	unchanged
0	1	0	1	set
1	0	x	x	set (will receive all characters)
1	1	x	0	unchanged
1	1	x	1	set

Bit 4: The break detect bit BRK is set when a break condition occurs and URXEIE bit is set. The break condition is recognized if the RXD line remains continuously low for at least 10 bits, beginning after a missing first stop bit. It is not cleared by receipt of a character after the break is detected - but reset by SWRST, system reset, and by reading the URXBUF.

Bit 5: The overrun error flag bit OE is set when a character is transferred into the URXBUF before the previous character has been read out. The previous character is overwritten and lost. OE is reset by SWRST, system reset, and by reading the URXBUF.

Bit 6: The parity error bit PE is set when a character is received with a mismatch between the number of "1's" and its parity bit and is loaded into the receive buffer. The parity checker includes the address bit - used with the address bit multiprocessor mode - in the calculation. The flag is disabled if parity generation and detection is not enabled. In such a case, it is read as "0". It is reset by SWRST, system reset, and by reading the URXBUF.

Bit 7: The framing error flag bit FE is set when a character is received with a "0" stop bit and is loaded into the receive buffer. Only the first stop bit is checked when more than one is used. The missing stop bit indicates that synchronization with the start bit has been lost and the character is incorrectly framed. FE is reset by SWRST, system reset, and reading URXBUF.

Note: Receive Status Control bits

The receive status control bits FE, PE, OE, BRK and RXWake are set conditionally by the hardware according to the conditions of the characters received. Once bits are set they remain set until the software will reset them directly or by reading the receive buffer. False character interpretation or missing interrupt capability can be the result of non-cleared error bits.

12.3.4 Baud Rate Select and Modulation Control Registers

The baud rate generator uses the content of both baud rate select registers UBR1 and UBR0 together with the modulation control register to generate the bit timing for the serial data stream.

UBR0
074h

7							0
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
rw	rw	rw	rw	rw	rw	rw	rw

UBR1
075h

7							0
2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^9	2^8
rw	rw	rw	rw	rw	rw	rw	rw

Figure 12.18: USART Baud Rate Select Register

$$\text{Baudrate} = \frac{\text{BRCLK}}{\text{UBR} + \frac{1}{n}\sum_{i=0}^{n-1} mi} \qquad \text{with UBR= [UBR1,UBR0]}$$

The baud rate control register range is: $3 \leq \text{UBR} < \text{0FFFFh}$

The modulation control register ensures a proper timing generation together with UBR0/1, even with crystal frequencies that are not integer multiples of the required baud rate.

UMCTL
073h

7							0
m7	m6	m5	m4	m3	m2	m1	m0
rw	rw	rw	rw	rw	rw	rw	rw

Figure 12.19: USART Modulation Control Register

The timing of the running bit is expanded by one clock cycle of the input clock of the baud rate divider if the actual bit m_i is set.

Each time a bit is received or transmitted the next bit in the modulation control register is used to determine the present bit timing. The first bit time in the protocol - the start bit time - is determined by UBR plus m0; the next bit by UBR plus m1,...
The modulation sequence is:

m0 - m1 - m2 - m3 - m4 - m5 - m6 - m7 - m0 - m1 - m2 - .....

12.3.5 USART Receiver Data Buffer URXBUF

The receiver buffer URXBUF contains previous data from the receiver shift register. Reading URXBUF resets the receive error bits, RXWake bit and interrupt flag URXIFG.

	7							0
URXBUF 076h	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
	r	r	r	r	r	r	r	r

Figure 12.20: USART Receive Buffer

In 7-bit length mode, the MSB of the URXBUF is always reset.

The receive buffer is loaded with the recently received character when receive and control conditions are true:

URXEIE	URXWIE	Load URXBUF by	PE	FE	BRK
0	1	error-free address characters	0	0	0
1	1	all address characters	x	x	x
0	0	error-free characters	0	0	0
1	0	all characters	x	x	x

12.3.6 USART Transmit Data Buffer UTXBUF

The transmit buffer contains current data to be transmitted by the transmitter.

	7							0
TXBUF 077h	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
	rw	rw	rw	rw	rw	rw	rw	rw

Figure 12.21: USART Transmit Buffer

The UTXIFG flag indicates that UTXBUF is ready to accept another character for transmission.
The transmission will be initialized by writing data to UTXBUF. The transmission of this data is started immediately, if the transmitter shift register is empty or is going to be empty.
Writing data to the transmit buffer should be done only if the buffer UTXBUF is empty, otherwise an unpredictable character can be transmitted.

12.4 UART Mode, Utilizing Features of low power Modes

There are several functions or operational features implemented that support the basic ultra-low power system of the MSP430 architecture:

- System start from any processor mode through sensing of UART frame start condition
- Use lowest input clock frequency for required baud rate
- Support of multiprocessor modes for reduced use of MSP430 resources.

12.4.1 Start Receive Operation from UART Frame

The most effective use of the start detection in the receive path is reached when the baudrate requires to run the system main clock MCLK, but the entire system can operate without MCLK. The receive start condition is the negative edge from the signal at URXD pin. Each time when it triggers the interrupt flag URXS, it requests a service when URXIE and GIE enable bits are set. The MSP430 system returns to the active mode and full system performance with MCLK and ACLK active.

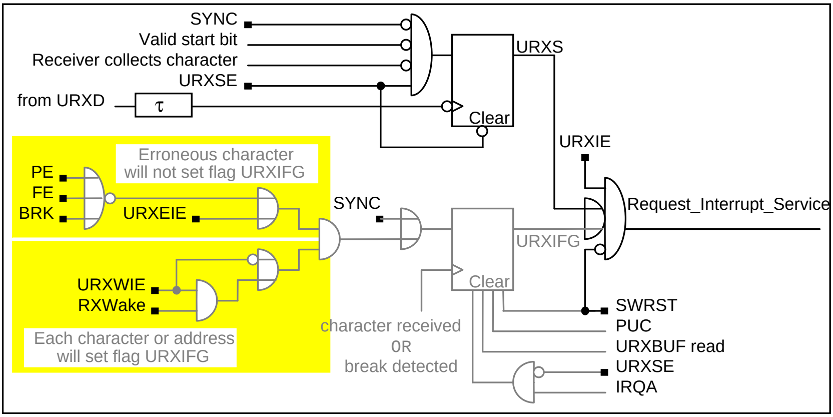

SYNC
Valid start bit
Receiver collects character
URXSE
from URXD
τ
URXS
Clear
URXIE
Erroneous character will not set flag URXIFG
PE
FE
BRK
URXEIE
SYNC
Request_Interrupt_Service
URXIFG
URXWIE
RXWake
Clear
Each character or address will set flag URXIFG
character received 0R break detected
SWRST
PUC
URXBUF read
URXSE
IRQA

Figure 12.22: Receive Start Conditions

Three character streams will not set the interrupt flag URXIFG:

- erroneous characters (URXEIE=0)
- address characters (URXWIE=1)
- and invalid start bit detect.

The interrupt software should handle these conditions. The interrupt handler must configure the correct clock system condition and the clock system will continue operation - and current consumption - until it is modified by the software. Whenever the CPU operates in the active mode the clock system is operating normally and start condition detection should not be used.

Start conditions

The URXD signal feed into the USART module is going first into a deglitch circuit. Glitches can not trigger the receive start condition flag URXS. This prevents the module from being started from small glitches on the URXD line. In noisy environments the current consumption is reduced, since glitches does not start system and USART.

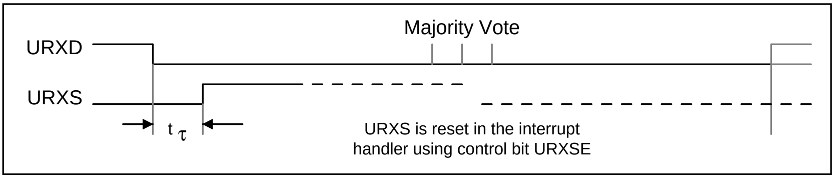

Majority Vote
URXD
URXS
t τ
URXS is reset in the interrupt handler using control bit URXSE

Figure 12.23: Receive Start Timing using URXS flag, start bit accepted

The UART stops receiving a character when the URXD signal exceeds the deglitch time t_τ but then the majority vote of the signal fails to start bit detection. The software should handle this condition and hold the system in the appropriate low power mode. The interrupt flag URXIFG is not set.

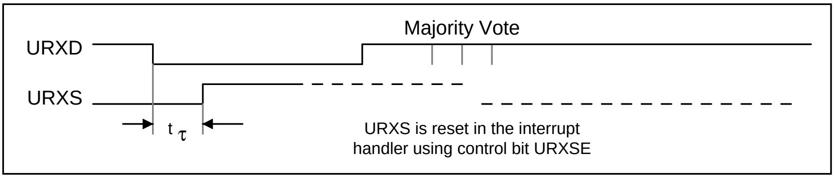

Majority Vote
URXD
URXS
t τ
URXS is reset in the interrupt handler using control bit URXSE

Figure 12.24: Receive Start Timing using URXS flag, start bit not accepted

Glitches at the URXD line are suppressed automatically and no further activity in the MSP430 is started. The data for the deglitch time t_τ is noted in the corresponding device specification.

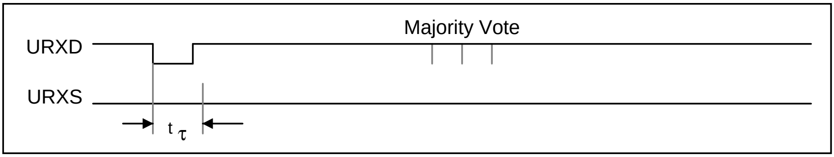


Figure 12.25: Receive Start Timing using URXS flag, glitch suppression

The interrupt handler should reset the URXSE bit in the control register UCTL to prevent further interrupt service requests from URXS signal and to enable the basic function of receive interrupt flag URXIFG.

```
********************************************************************
*     INTERRUPT HANDLER FOR FRAME START CONDITION AND              *
*     CHARACTER RECEIVE                                            *
********************************************************************
IFG2     .EQU     3                ; URXIFG AND UTXIFG IN ADDRESS 3
UTCTL    .EQU     71H              ;
UTXIFG   .EQU     0                ;
URXSE    .EQU     8                ;
                                   ;

URX_INT  BIT.B    #URXIFG,&IFG2    ; TEST URXIFG SIGNAL TO CHAECK
         JNE      ST_COND          ; IF FRAME START CONDITION
         .....
         .....
ST_COND  BIC.B    #URXSE,&UTCTL    ; CLEAR FF/SIGNAL URXS, STOP
                                   ; FURTHER INTERRUPT REQUESTS
         BIS.B    #URXSE,&UTCTL    ; PREPARE FF_URXS FOR NEXT FRAME
         .....                     ;.START CONDITION
         .....                     ; AND SET THE CONDITIONS TO RUN
         .....                     ; THE CLOCK NEEDED FOR UART RX
```

Note: Break detect BRK bit with halted UART clock

If the UART is operating with the feature of wake-up with a start condition, and to switch off the UCLK whenever a character is completely received, the break of the communication line can not be detected automatically by the UART hardware. The break detect needs the clock BRSCLK out of the baud rate generator to detect this conditions, but it is stopped upon the missing UCLK.

12.4.2 Maximum Utilization of Clock Frequency vs. Baud Rate UART Mode

The current consumption depends linearly on the clock frequency. It should be kept to the minimum required to meet the application conditions. Fast communication speed is needed due to various reason - calibration and test in manufacturing processes, alarm situations in critical applications, response time to human requests for information,......
The baud rate generator in the MSP430 USART is realized to meet baud rates up to $^1/_3$ of the clock frequency. An additional modulation of the baud rate timing gives extra benefit since the timing for the single bit in a frame can be adjusted. The timing is adjusted from bit to bit to meet the requirements even when a non-integer division is needed. Baud rates can be done from a 32,768Hz crystal up to 4800 Baud with errors of max. 11%. Standard UART's can - with the worse maximum error (-14.6%) reach maximum baud rates of 75 Baud.

12.4.3 Support of multiprocessor modes for reduced use of MSP430 resources

Communication systems with multiple character protocols can use the features of multiprocessor modes - whether the idle line or the address bit protocol. The first character can be a target address, a message identifier or can have another definition. This character is interpreted by software, and if there is any significance for the application the succeeding characters are collected and further activities defined. No significance of the first character would stop any activity for the processing device. The application of this feature is supported by the wake-up interrupt feature in receive situation, and to send wake-up conditions along with transmission. Avoiding activity on characters without any significance reduces the use of MSP430 resources and the system can remain in the most efficient power conserving mode.
Additional to the multiprocessor modes, rejection of erroneous characters avoids interrupt handling of these characters. This is useful whenever erroneous characters will not be processed anyway. The processor waits in the most efficient power conserving mode until a character can be processed.

12.5 Baud Rate Considerations

The baud rate generator of the MSP430 uses one divider and a modulator. A given crystal's frequency and a required baud rate will determine the needed division factor N:

N = **Error!**

The necessary division factor N usually has an integer part and a fraction. The divider in the baudrate generator realizes the integer portion of the division factor N and the modulator is responsible for meeting the fractional part as close as possible. The factor N is defined:

$$N = UBR + \frac{1}{n} \sum_{i=0}^{n-1} m_i$$

where N is the target division factor
UBR is the 16-bit representative of register UBR1 and UBR0
i is the actual bit in the frame
n is the number bits in the frame
m_i is the data of the actual modulation bit.

$$\text{Baudrate} = \frac{BRCLK}{N} = \frac{BRCLK}{UBR + \frac{1}{n} \sum_{i=0}^{n-1} mi}$$

Bit Timing in Transmit Operation

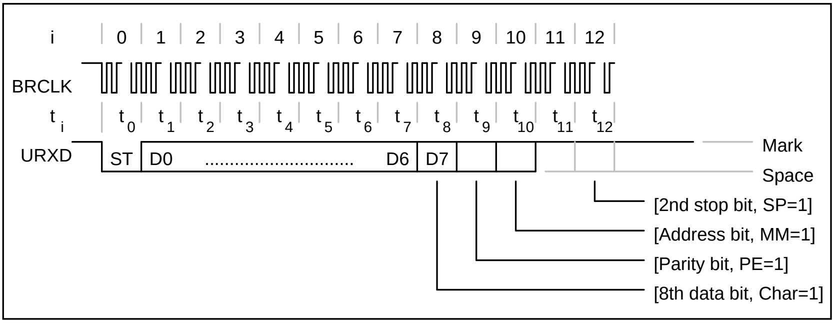


Figure 12.26: MSP430 Transmit Bit Timing

The timing for each individual bit in one frame or character is the sum of the actual bit timings. The error of the baud rate generation in respect to the required ideal timing is calculated for each individual bit. The relevant information is the error relative to the actual bit, not the overall relative error.

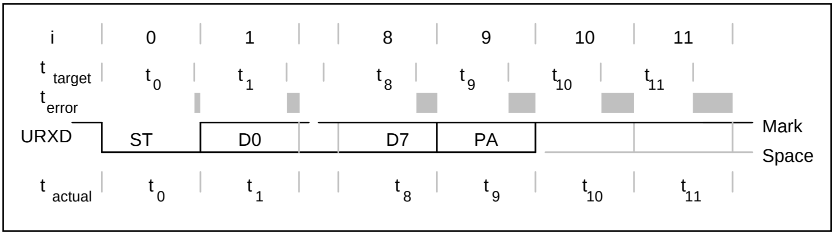

i
0 1 8 9 10 11
t_{target}
t_0 t_1 t_8 t_9 t_{10} t_{11}
t_{error}
URXD
ST
D0
D7
PA
Mark
Space
t_{actual}
t_0 t_1 t_8 t_9 t_{10} t_{11}

Figure 12.27: MSP430 Transmit Bit Timing Errors

Even small errors per bit (relative errors) end up in larger errors - they should be considered to be accumulative, not relative. The error of an individual bit can be calculated by:

$$\boxed{\text{Error [\%]} = \frac{\sum_{i=0}^{n-1} t_{\text{actual}_i} - \sum_{i=0}^{n-1} t_{\text{target}_i}}{t_{\text{baud rate}}} \times 100\%}$$

OR

$$\boxed{\text{Error [\%]} = \left(\left(\frac{\text{baud rate}}{\text{BRCLK}} \text{ x } ((i+1) \text{ x UBR } + \sum_{i=0}^{n-1} m_i\right) - (i+1)\right) \text{ x } 100\%}$$

with baud rate is the required baud rate
BRCLK is the input frequency - selected for UCLK, ACLK or MClK
i=0 for the start bit, 1 for data bit D0, ...
UBR is division factor in registers UBR1 and UBR0

Example 1

The following data are assumed:

baud rate =	2400 Baud
BRCLK =	32,768Hz (ACLK)
UBR =	13, since the ideal division factor should be 13.67
m = 6Bh:	m7=0, m6=1, m5=1, m4=0, m3=1, m2=0, m1=1 and m0=1 The LSB (m0) of the modulation register is used first.

Start bit $\text{Error [\%]} = (\frac{\text{baud rate}}{\text{BRCLK}} \text{ x } ((0+1) \text{ x UBR } + 1) - 1) \text{ x } 100\% = 2.54\ \%$

Data bit D0 $\text{Error [\%]} = (\frac{\text{baud rate}}{\text{BRCLK}} \text{ x } ((1+1) \text{ x UBR } + 2) - 2) \text{ x } 100\% = 5.08\ \%$

Data bit D1 $\text{Error [\%]} = (\frac{\text{baud rate}}{\text{BRCLK}} \text{ x } ((2+1) \text{ x UBR } + 2) - 3) \text{ x } 100\% = 0.29\ \%$

Data bit D2 $\text{Error [\%]} = (\frac{\text{baud rate}}{\text{BRCLK}} \text{ x } ((3+1) \text{ x UBR } + 3) - 4) \text{ x } 100\% = 2.83\ \%$

Data bit D3 $\text{Error [\%]} = (\frac{\text{baud rate}}{\text{BRCLK}} \text{ x } ((4+1) \text{ x UBR } + 3) - 5) \text{ x } 100\% = -1.95\ \%$

Data bit D4 $\text{Error [\%]} = (\frac{\text{baud rate}}{\text{BRCLK}} \text{ x } ((5+1) \text{ x UBR } + 4) - 6) \text{ x } 100\% = 0.59\ \%$

Data bit D5 $\text{Error [\%]} = (\frac{\text{baud rate}}{\text{BRCLK}} \text{ x } ((6+1) \text{ x UBR } + 5) - 7) \text{ x } 100\% = 3.13\ \%$

Data bit D6 $\text{Error [\%]} = (\frac{\text{baud rate}}{\text{BRCLK}} \text{ x } ((7+1) \text{ x UBR } + 5) - 8) \text{ x } 100\% = -1.66\ \%$

Data bit D7 $\text{Error [\%]} = (\frac{\text{baud rate}}{\text{BRCLK}} \text{ x } ((8+1) \text{ x UBR } + 6) - 9) \text{ x } 100\% = 0.88\ \%$

Parity bit $\text{Error [\%]} = (\frac{\text{baud rate}}{\text{BRCLK}} \text{ x } ((9+1) \text{ x UBR } + 7) - 10) \text{ x } 100\% = 3.42\ \%$

Stop bit 1 $\text{Error [\%]} = (\frac{\text{baud rate}}{\text{BRCLK}} \text{ x } ((10+1) \text{ x UBR } + 7) - 11) \text{ x } 100\% = -1.37\ \%$

Stop bit 2 $\text{Error [\%]} = (\frac{\text{baud rate}}{\text{BRCLK}} \text{ x } ((11+1) \text{ x UBR } + 8) - 12) \text{ x } 100\% = 1.17\ \%$

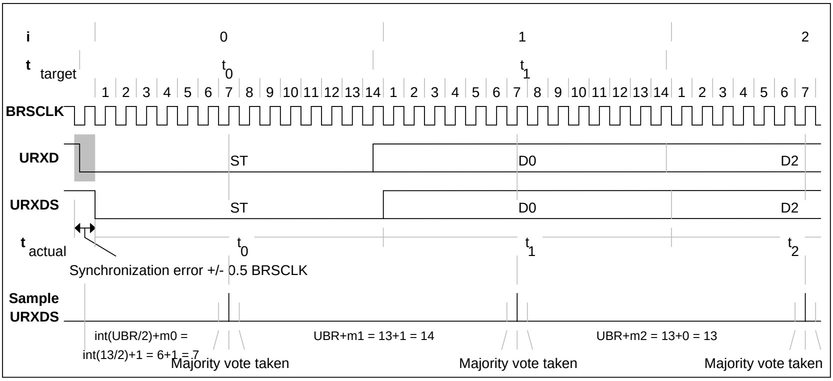
The standard baud rate data needed for the baud rate registers and the modulation register are listed for the watch crystal 32,768Hz (ACLK) and MCLK assumed to be 32-times the ACLK frequency. The error listed is calculated for the transmit and receive path. Additionally to this error for the receive situation, the synchronization error should also be considered.

Baud rate	Divide by		ACLK (32 768Hz)			max. TX error %	max. RX error %	Synchr. RX error %	MCLK (1 048 576Hz)			max. TX error %	max. RX error %
	ACLK	MCLK	UBR1	UBR0	UMOD				UBR1	UBR0	UMOD		
75	436.91	13981	1	B4	FF	-.1/.3	-.1/.3	+/-2	36	9D	FF	0/.1	+/-2
110	297.89	9532.51	1	29	FF	0/.5	0/.5	+/-3	25	3C	FF	0/.1	+/-3
150	218.45	6990.5	0	DA	55	0/.4	0/.4	+/-2	1B	4E	FF	0/.1	+/-2
300	109.23	3495.25	0	6D	22	-.3/.7	-.3/.7	+/-2	0D	A7	00	-.1/0	+/-2
600	54.61	1747.63	0	36	D5	-1/1	-1/1	+/-2	06	D3	FF	0/.3	+/-2
1200	27.31	873.81	0	1B	03	-4/3	-4/3	+/-2	03	69	FF	0/.3	+/-2
2400	13.65	436.91	0	0D	6B	6/3	-6/3	+/-4	01	B4	FF	0/.3	+/-2
4800	6.83	218.45	0	06	6F	-9/11	-9/11	+/-7	0	DA	55	0/.4	+/-2
9600	3.41	109.23	0	03	4A	-21/12	-21/12	+/-15	0	6D	03	-.4/1	+/-2
19 200		54.61							0	36	6B	-.2/2	+/-2
38 400		27.31							0	1B	03	-4/3	+/-2
76 800		13.65							0	0D	6B	-6/3	+/-4
115 200		9.10							0	09	08	-5/7	+/-7

Table 12.2: Mostly used Baud Rates, Baudrate data and errors

The synchronization error results from the asynchronous timing between the data signal at the URXD pin and the internal clock system. The receive signal is synchronized with the BRSCLK clock. The BRSCLK clock is sixteen to thirty-one times faster than the bit timing:

BRSCLK = BRCLK	for		N	≤ 1F
BRSCLK = BRCLK/2	for	20h	≤ N	≤ 3Fh
BRSCLK = BRCLK/4	for	40h	≤ N	≤ 7Fh
BRSCLK = BRCLK/8	for	80h	≤ N	≤ FFh
BRSCLK = BRCLK/16	for	100	≤ N	≤ 1FF
BRSCLK = BRCLK/32	for	200	≤ N	≤ 3FFh
BRSCLK = BRCLK/64	for	400	≤ N	≤ 7FFh
BRSCLK = BRCLK/128	for	800h	≤ N	≤ FFFh
BRSCLK = BRCLK/256	for	1000h	≤ N	≤ 1FFFh
BRSCLK = BRCLK/512	for	2000h	≤ N	≤ 3FFFh
BRSCLK = BRCLK/1024	for	4000h	≤ N	≤ 7FFFh
BRSCLK = BRCLK/2048	for	8000h	≤ N	≤ FFFFh



The target baud rate timing $t_{target0}$ for the start bit detection is half the baud rate timing $t_{baud\ rate}$ since the bit is tested in the middle of its period. The target baud rate timing $t_{target i}$ for the all other succeeding bits is the baud rate timing $t_{baud\ rate}$.

$$\text{Error [\%]} = \frac{t_{actual0} + t_{target0}}{0.5 \text{ x } t_{target0}} + \frac{\sum_{i=1}^{n-1} t_{actual i} - \sum_{i=1}^{n-1} t_{target i}}{t_{target i}} \text{ x } 100\%$$

OR

$$\text{Error [\%]} = \left(\frac{\text{baud rate}}{\text{BRCLK}} \text{ x } \{2 \text{ x } [m0 + \text{int}(UBR / 2)] + (i\text{x}UBR + \sum_{i=1}^{n-1} m_i)\} - 1 - i\right) \text{x } 100\%$$

where baud rate is the required baud rate
BRCLK is the input frequency - selected for UCLK, ACLK or MClK
i=0 for the start bit, 1 for data bit D0, ...
UBR is division factor in registers UBR1 and UBR0

Example 2

The following data are assumed:

baud rate = 2400 Baud
BRCLK = 32,768Hz (ACLK)
UBR = 13, since the ideal division factor should be 13.67
m = 6Bh: m7=0, m6=1, m5=1, m4=0, m3=1, m2=0, m1=1 and m0=1
The LSB (m0) of the modulation register is used first.

Start bit Error [%] $= \{\frac{\text{baud rate}}{\text{BRCLK}} \text{ x } [2\text{x}(1+6)+(0\text{xUBR}+0-0)]-1\} \text{ x } 100\% = 2.54\ \%$

Data bit D0 Error [%] $= \{\frac{\text{baud rate}}{\text{BRCLK}} \text{ x } [2\text{x}(1+6)+(1\text{xUBR}+1)]-1-1\} \text{ x } 100\% = 5.08\ \%$

Data bit D1 Error [%] $= \{\frac{\text{baud rate}}{\text{BRCLK}} \text{ x } [2\text{x}(1+6)+(2\text{xUBR}+1)]-1-2\} \text{ x } 100\% = 0.29\ \%$

Data bit D2 Error [%] $= \{\frac{\text{baud rate}}{\text{BRCLK}} \text{ x } [2\text{x}(1+6)+(3\text{xUBR}+2)]-1-3\} \text{ x } 100\% = 2.83\ \%$

Data bit D3 Error [%] $= \{\frac{\text{baud rate}}{\text{BRCLK}} \text{ x } [2\text{x}(1+6)+(4\text{xUBR}+2)]-1-4\} \text{ x } 100\% = -1.95\ \%$

Data bit D4 Error [%] $= \{\frac{\text{baud rate}}{\text{BRCLK}} \text{ x } [2\text{x}(1+6)+(5\text{xUBR}+3)]-1-5\} \text{ x } 100\% = 0.59\ \%$

Data bit D5 Error [%] $= \{\frac{\text{baud rate}}{\text{BRCLK}} \text{ x } [2\text{x}(1+6)+(6\text{xUBR}+4)]-1-6\} \text{ x } 100\% = 3.13\ \%$

Data bit D6 Error [%] $= \{\frac{\text{baud rate}}{\text{BRCLK}} \text{ x } [2\text{x}(1+6)+(7\text{xUBR}+4)]-1-7\} \text{ x } 100\% = -1.66\ \%$

Data bit D7 Error [%] $= \{\frac{\text{baud rate}}{\text{BRCLK}} \text{ x } [2\text{x}(1+6)+(8\text{xUBR}+5)]-1-8\} \text{ x } 100\% = 0.88\ \%$

Parity bit Error [%] $= \{\frac{\text{baud rate}}{\text{BRCLK}} \text{ x } [2\text{x}(1+6)+(9\text{xUBR}+6)]-1-9\} \text{ x } 100\% = 3.42\ \%$

Stop bit 1 Error [%] $= \{\frac{\text{baud rate}}{\text{BRCLK}} \text{ x } [2\text{x}(1+6)+(10\text{xUBR}+6)]-1-10\} \text{ x } 100\% = -1.37\ \%$

Stop bit 2 Error [%] $= \{\frac{\text{baud rate}}{\text{BRCLK}} \text{ x } [2\text{x}(1+6)+(11\text{xUBR}+7)]-1-11\} \text{ x } 100\% = 1.17\ \%$

Baud Rate Considerations - Conclusion

The system chosen to generate a proper serial communication bit stream allows baud rates up to nearly the clock rate fed into the USART. It enables low accumulating errors through modulation of the individual bit timing. In practice an error margin of 20% to 30% should make proper serial communication possible.

13 USART Peripheral Interface, SPI Mode

The synchronous interface is a serial channel which allows a serial bit stream of 7 or 8 bits to be shifted into and out of the MSP430, at an externally determined rate or at an internally programmed rate. The USART module is connected to the CPU as a byte peripheral. It connects the controller to the external system environment by three or four external pins.

USART's serial synchronous communication features:

- Control bit SYNC in control register UCTL is set to select synchronous mode
- Supports 3 pin and 4 pin SPI operation via SOMI, SIMO, UCLK and STE
- Select master or slave mode
- Separate shift registers for receive (URXBUF) and transmit (UTXBUF)
- Double buffering for receive and transmit
- Clock polarity and clock phase control
- Clock frequency control in master mode
- Character length 7 or 8 bits/character

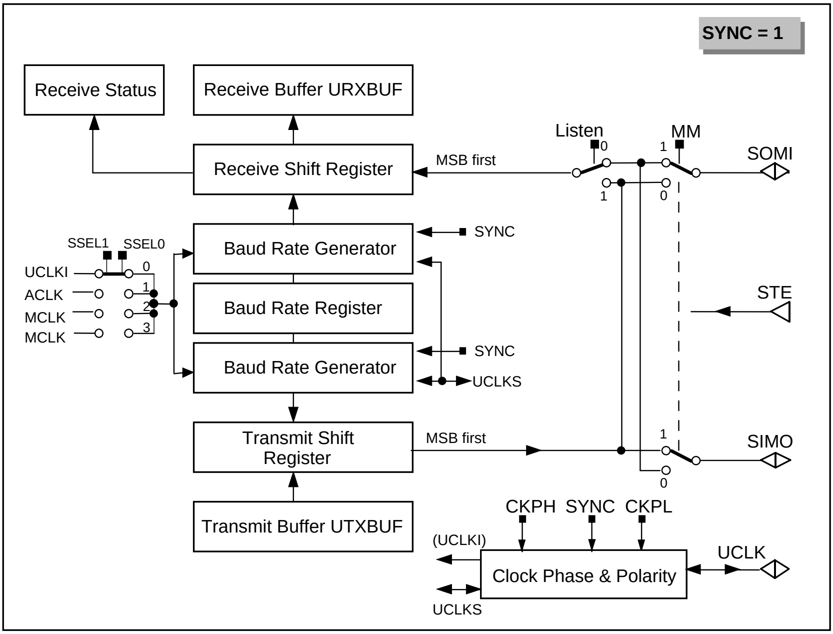


Figure 13.1: Block diagram of USART - SPI mode

13.1 USART’s Synchronous Operation

In the synchronous mode, data and clock signals are used to transmit and receive serial data. The master supplies the clock and data. The slave(s) use this clock to shift the serial information in and out. The 4 pin SPI mode uses a control line additionally, to enable a slave to receive and transmit data. It is controlled by the master.

Three or four signals are used for the data exchange:

- SIMO Slave in, master out
- SOMI Slave out, master in
- UCLK USART clock, the master drives this signal and the slave uses it to receive and transmit data
- STE Slave transmit enable, used in 4-pin mode to control more than one slave in a multiple master and slave system.

The interconnection of the USART in synchronous mode to another device’s serial port with one common transmit receive shift register is shown when MSP430 is master or slave. The operation will remain identical. The master initiates the transfer by sending the UCLK signal. For the master, data is shifted out of the transmit shift register on one clock edge and shifted into the receive shift register on the opposite edge. For the slave, the data shifting operation is the same, using one common register shift for transmitting and receiving data. Master and slave send and receive data at the same time.

Whether or not the data is meaningful or dummy data depends upon the application software:

1. Master sends data and Slave sends dummy data
2. Master sends data and Slave sends data
3. Master sends dummy data and Slave sends data.

The master can initiate data transfer at any time, and controls the UCLK. The software protocol determines the way in which the master knows when the slave wishes to broadcast data.

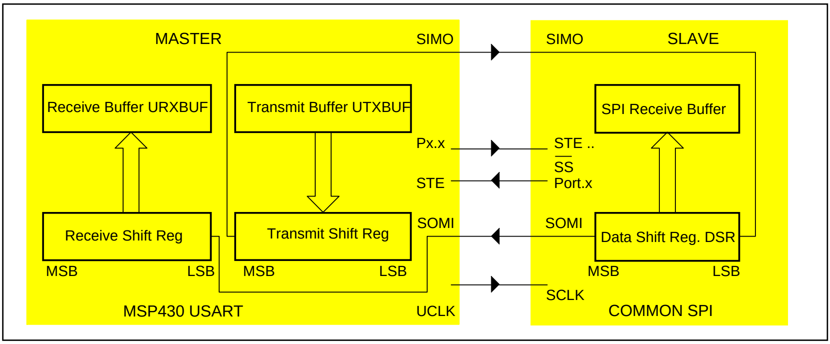
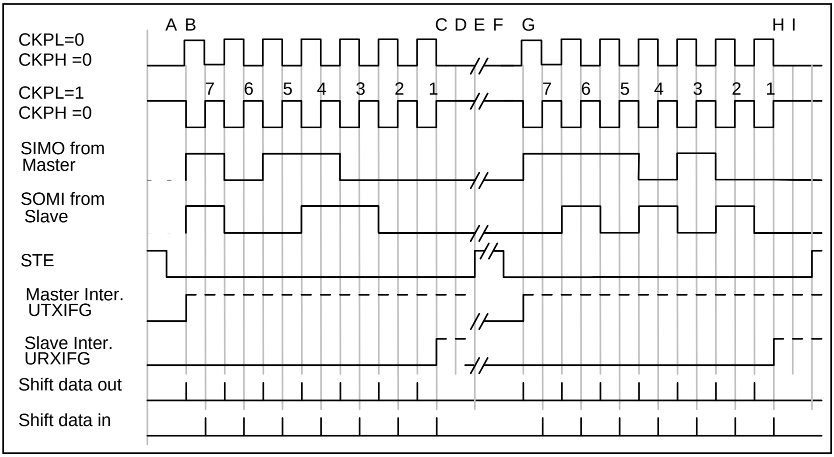
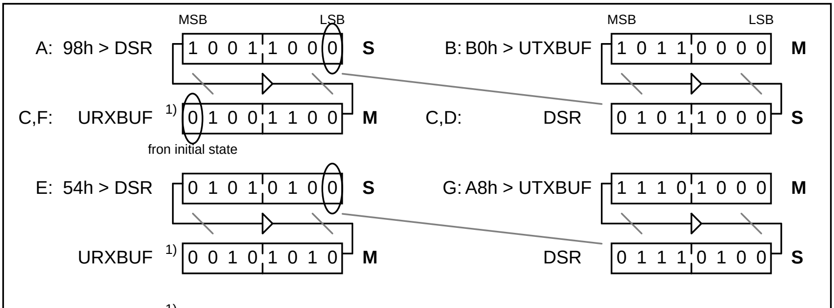


Figure 13.2: MSP430 USART as Master, external device with SPI as slave

There follows an example of serial synchronous data transfer for a character length of seven bits. The initial content of receive shift register is 00.



A: Slave writes 98h to the DSR and waits for the master to shift out data.
B: Master writes B0h to UTXBUF which is immediately transferred to the Transmit Shift Register and starts the transmission.
C: First character is finished and sets the interrupt flags.
D: Slave reads 58h from its receive buffer (right justified).
E: Slave writes 54h to its DSR and waits for the master to shift out data.
F: Master reads 4Ch from receive buffer URXBUF (right justified).
G: Master writes E8h to the transmit buffer UTXBUF and starts the transmission.
H: Second character is finished and sets the interrupt flag.
I: Master receives 2Ah and slave receives 74h (right justified).



[1] In 7bit Mode, the MSB of RXBUF is always read 0.
S: Slave **M:** Master

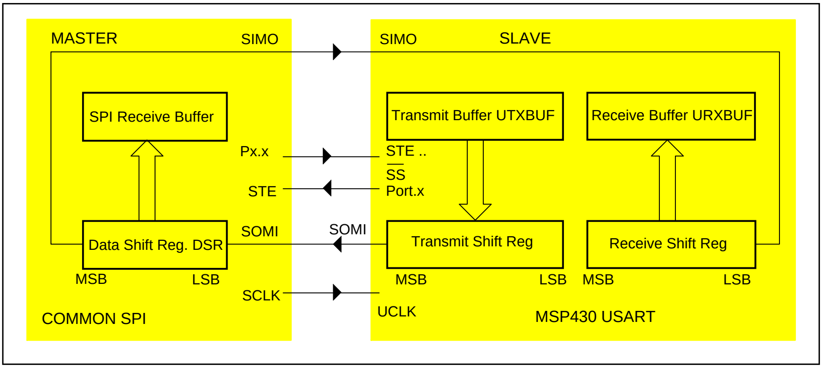


Figure 13.3: MSP430 USART as Slave in 3 pin or 4 pin configuration

13.1.1 Master Mode in Synchronous USART Mode, MM=1, SYNC=1

The master mode is selected when master mode bit MM in the control register UCTL is set. The USART controls the serial communication network by providing UCLK at the UCLK pin. Data is output on the SIMO pin on the first UCLK period and latched from the SOMI pin in the middle of the corresponding UCLK period.
The data written to the transmit buffer UTXBUF is moved to the transmit shift register as soon as it is empty and this initiates the data transfer on the SIMO pin, with the most significant bit first. At the same time, received data is shifted into the receive shift register, and upon completing of receiving the selected number of bits, the received data is transferred to the receive buffer URXBUF and the receive interrupt flag URXIFG is set. Data is shifted into the receive shift register, with the most significant bit first. It is stored right justified in receive buffer URXBUF. When previous data was not read from the receive buffer URXBUF the overrun error bit OE is set.

Note: USART Synchronous Master Mode, Receive initiation

The master should write data to the transmit buffer UTXBUF to receive a character. The receive starts when the transmit shift register is empty and the data is transferred into it. Receive and transmit always take place together, at opposite clock edges.

The control of the protocol can be done by using the transmit interrupt flag UTXIFG or the receive interrupt flag URXIFG. Using the UTXIFG immediately after sending the data from the shift register to the slave the data from the buffer is transferred to the shift register and the transmission starts. The slave receive timing needs to ensure pick-up of the data in time. The URXIFG flag indicates when the data is shifted out and in completely. The master can use URXIFG to ensure that the slave should be ready to receive the next data properly.

Any standard digital output including STE in standard digital port function can be used to select a slave. The slave use the STE signal to enable its access to the SOMI data line and to enable to receive the clocks on UCLK.

4-pin SPI master mode, SYNC=1, STC=0, MM=1

The signal on STE is used by the active master to prevent bus conflicts with another master. The STE pin is input when the corresponding PnSEL bit selects the module function. The master operates normally while the STE signal is high. Whenever the STE signal is set to low - e.g. another device requests to become master - the actual master reacts with:

- the pins that drive the SPI bus lines SIMO and UCLK, are set to inputs
- the error bit FE and the interrupt flag URXIFG in the URCTL register are set.

The bus conflict is then removed - SIMO and UCLK do not drive the bus lines - and the error flag indicates to the software the violation of the system integrity. The pins SIMO and UCLK are forced to inputs while STE is low, and return to the conditions defined by the corresponding control bits when STE returns to high.

In the 3-pin mode the STE input signal is not relevant for the master.

13.1.2 Slave Mode in SPI Mode, MM=0, SYNC=1

The slave mode is selected when the master mode bit MM in the control register is reset and synchronous mode is selected.

The UCLK pin is used as the input for the serial shift clock supplied by an external master. The transfer rate is determined by this clock and not by the internal bit rate generator. The data, loaded into transmit shift register via transmit buffer UTXBUF before start of UCLK, is transmitted on SOMI pin using the UCLK applied by the master. Simultaneously the serial data applied to SIMO pin are shifted into the receive shift register on the opposite edge of the clock.

The receive interrupt flag URXIFG indicates when data is received and transferred into the receive buffer. The overrun error bit is set when previous received data is not read before the new data is written to the receive buffer.

4 pin SPI slave mode, SYNC=1, MM=0, STC=0

In the 4 pin SPI mode the signal STE is used by the slave to enable transmit and receive operation. The STE signal is used to enable the receive and transmit function of the slave. It is applied from the SPI master. The receive and transmit operation is disabled when the STE signal is high, and enabled when it is low. Whenever the STE signal becomes high any started receive operation is halted, and continues when the STE signal is low again. The STE signal is used to enable one slave to access the data lines. The SOMI is input if STE is high.

13.2 Interrupt and Control Function

The USART peripheral serves two main interrupt sources, the transmission and receive. Two individual interrupt vectors are available, one for receive and one for transmit interrupt events.

The control bits of the USART are located in the SFR address range:

• Receive Interrupt Flag	URXIFG	initial state reset (by PUC/SWRST)
• Receive Interrupt Enable	URXIE	initial state reset (by PUC/SWRST)
• Receive Enable	URXE	initial state reset (by PUC)
• Transmit Interrupt Flag	UTXIFG	initial state set (by PUC/SWRST)
• Transmit Interrupt Enable	UTXIE	initial state reset (by PUC/SWRST)
• Transmit Enable	UTXE	initial state reset (by PUC)

The receiver and transmitter of the USART operate in parallel and use the same baud rate generator in synchronous master mode. In synchronous slave mode the external clock - applied to UCLK - is used for receiver and transmitter.

13.2.1 USART Receive Enable

The Receiver Enable bit URXE enables or disables the receiver from collecting the bit stream on the URXD/SOMI data line. Disabling the USART receiver (URXE=0) will stop the receive operation after completing a started receive operation, or stop immediately if no receive operation is active. In synchronous mode the clock UCLK does not shift any data into the receiver shift register.

Receive when MSP430 is master
The receive operation is identical for 3-pin and 4-pin mode, when MSP430 USART is selected to be SPI master.

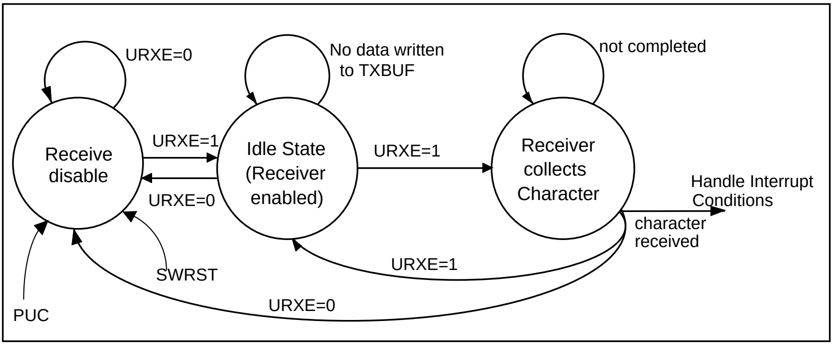


Figure 13.4: State diagram on Receiver enable URXE. MSP430 is master

Receive when MSP430 is slave, 3-pin mode

The receive operation is different for 3-pin and 4-pin mode when MSP430 USART is selected to be SPI slave. In the 3-pin mode no external SPI receive control signal stops a receive operation which has started. Power-up clear PUC, software reset SWRST or receive enable URXE can stop a receive operation and reset the USART.

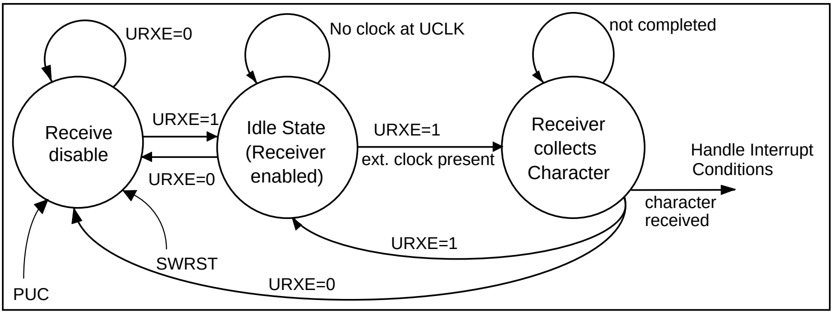


Figure 13.5: State diagram on Receiver enable URXE. MSP430 is slave/3-pin mode

> **Note: URXE re-enable, SPI Mode**
>
> Since the receiver is completely disabled a re-enable of the receiver is asynchronous to any data stream on the communication line. Synchronization to the data stream should be handled by the software protocol as usual in 3-pin SPI mode.

Receive when MSP430 is slave, 4-pin mode
In the 4-pin mode the external SPI receive control signal applied to pin STE stops a started receive operation. Power-up clear PUC, software reset SWRST or receive enable URXE can stop a receive operation and reset the operation control state machine. Whenever the STE signal is set to high, the receive operation is halted.

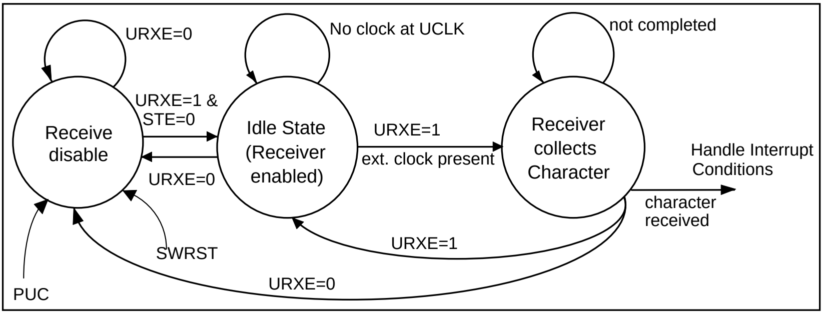


Figure 13.6: State diagram on Receiver enable URXE. MSP430 is slave/4-pin mode

13.2.2 USART Transmit Enable

The transmit enable bit UTXE enables or disables a character from being shifted onto the serial data line. If this bit is reset, the transmitter is disabled but any active transmission is not halted until all data previously written into the transmit buffer has been sent. If the transmission is completed any further write to the transmitter buffer will not result in a data transmission. When the UTXBUF was ready, a pending request for transmission will remain, and this results in an immediate start of transmission when UTXE is set and the transmitter is empty. A low signal on the STE signal removes the active master (4-pin mode) from the bus. Low at STE indicates that another master requests the active master function.

USART Transmit Enable, MSP430 is master

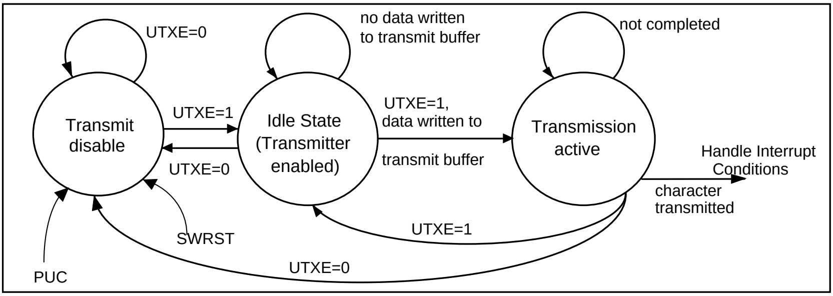


Figure 13.7: State diagram on Transmitter enable, MSP430 is master

USART Transmit Enable, MSP430 is slave

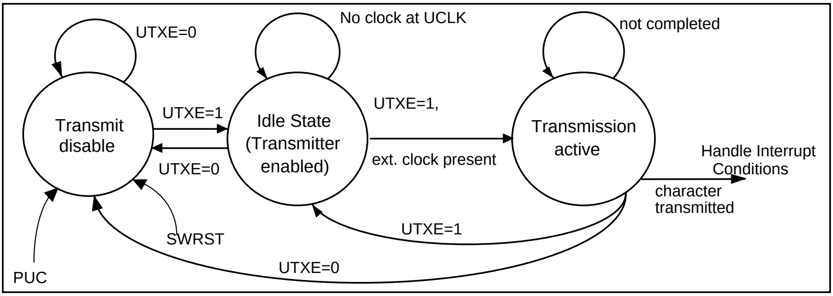


Figure 13.8: State diagram on Transmitter enable, MSP430 is slave

When UTXE is reset any data can be written regularly into the transmit buffer, but no transmission is started. Once the UTXE bit is set, the data in the transmit buffer are immediately loaded into the transmit shift register and the transmission of the character is started.

Note: Write to UTXBUF, SPI Mode

Data should never be written into the transmit buffer UTXBUF when it is not ready (UTXIFG=0) but the transmitter is enabled (UTXE=1). The character shifted out can be random.

13.2.3 USART Receive Interrupt Operation

The receive interrupt flag URXIFG is set each time a character is received and loaded into the receive buffer. Asynchronous conditions are not used.

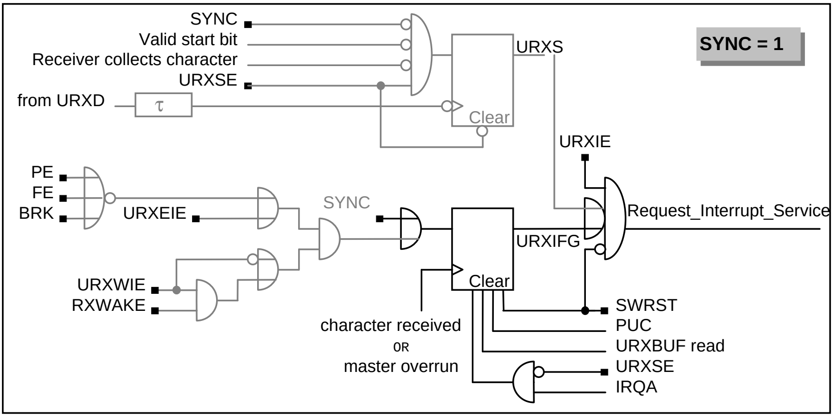


Figure 13.9: Receive Interrupt Conditions

URXIFG is reset at system reset PUC and at a software reset SWRST. URXIFG is reset automatically if the interrupt is served or the receive buffer URXBUF is read. The Receive Interrupt Enable bit URXIE enables, if set, serving of a pending interrupt request. Both, the receive interrupt flag URXIFG and the receive interrupt enable bit URXIE are reset with PUC and SWRST.

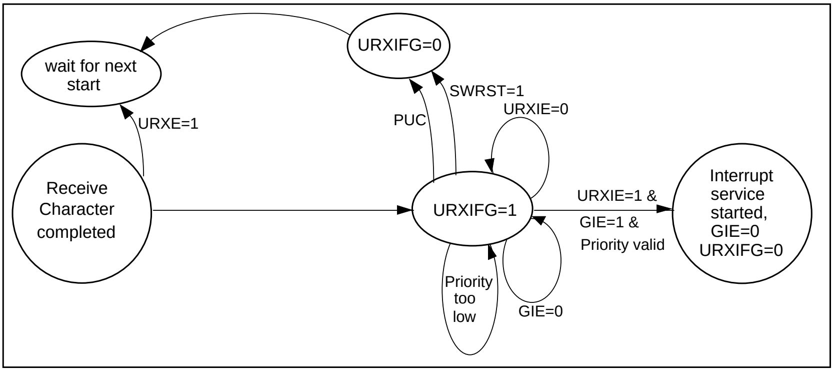


Figure 13.10: State diagrams on receive interrupt

13.2.4 USART Transmit Interrupt Operation

The transmit interrupt flag UTXIFG is set by the transmitter to indicate that the transmitter buffer UTXBUF is ready to accept another character. This bit is automatically reset if the interrupt request service is started or a character is written into the UTXBUF. This flag will assert a transmitter interrupt if the local (UTXIE) and general (GIE) interrupt enable bits are set. The UTXIFG is set after system reset PUC or SWRST are removed.

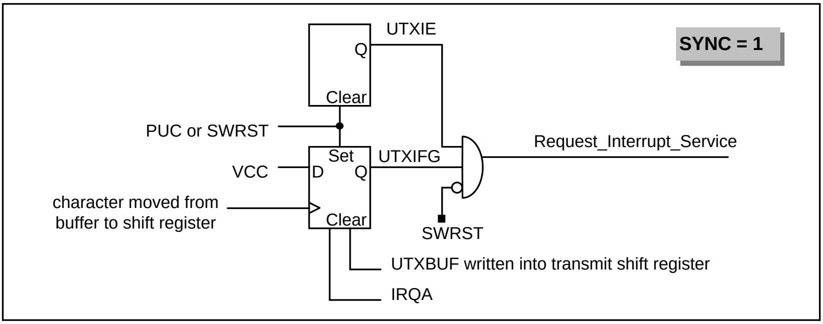


Figure 13.11: Transmit Interrupt Condition

The transmit interrupt enable UTXIE bit controls the ability of the UTXIFG to request an interrupt, but does not prevent the flag UTXIFG from being set. The UTXIE is reset with PUC or software reset bit SWRST. The UTXIFG bit is set after system reset PUC or software reset but the UTXIE bit is reset to ensure full interrupt control capability.

13.3 Control and Status Register

The USART module hardware is byte structured and should be accessed by byte processing instructions (suffix 'B').

Register	short form	Register type	Address	Initial state
● USART Control register	UCTL	Type of read/write	070h	See ....
● Transmit Control register	UTCTL	Type of read/write	071h	individual ...
● Receive Control register	URCTL	Type of read/write	072h	bit description
● Modulation Control reg.	UMCTL	Type of read/write	073h	unchanged
● Baud Rate register 0	UBR0	Type of read/write	074h	unchanged
● Baud Rate register 1	UBR1	Type of read/write	075h	unchanged
● Receive Buffer	URXBUF	Type of read/write	076h	unchanged
● Transmit Buffer	UTXBUF	Type of read	077h	unchanged

All bits are random after PUC unless otherwise noted by the detailed functional description.
Reset of the USART is performed by PUC or SWRST bit. After power-up (PUC) the SWRST bit remains set and the USART remains in this condition until the reset is disabled by resetting the SWRST bit. The SPI mode is disabled after PUC.
The USART module operates in asynchronous or in synchronous mode defined by the SYNC bit. The bits in the control registers may have different functions in the two modes. All bits are described with their function in the synchronous mode - SYNC=1. Their function in the asynchronous mode is described in the USART’s serial interface UART mode section.

13.3.1 USART Control register

The information stored in the control register determines the basic operation of the USART module. The register bits select the communication mode and number of bits per character. All bits should be programmed according to the selected mode before reset is disabled by resetting bit SWRST.

	7							0
UCTL 070h	unused	unused	unused	CHAR	Listen	SYNC	MM	SWRST
	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-1

Figure 13.12: USART Control Register

Bit 0: The USART state machines and operating flags are initialized to the reset condition, if the software reset bit is set. Until the SWRST bit is reset, all affected logic is held in the reset state. This implies that after a system reset the USART must be re-enabled by resetting this bit.

Bit 1: Master mode is selected when the MM bit is set. The USART module slave mode is selected when the MM bit is reset.

Bit 2: Peripheral module mode select.
The SYNC bit selects the function of the USART peripheral interface module. Some of the USART control bits will have different functions in UART and SPI mode.
SYNC = 0 : UART function is selected.
SYNC = 1 : SPI function is selected.

Bit 3: The Listen bit selects if internally the transmitted data is fed back into the receiver

Bit 4: Character length.
This register bit selects the length of the character to be transmitted as 7 or 8 bits.
CHAR = 0 : 7 bit data.
CHAR = 1 : 8 bit data.

Bit 5: unused

Bit 6: unused

Bit 7: unused

13.3.2 Transmit Control Register UTCTL

The register controls the USART hardware associated with transmitter operation.

	7							0
UTCTL 071h	CKPH	CKPL	SSEL1	SSEL0	unused	unused	STC	TXEPT
	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-1

Figure 13.13: USART Transmitter Control Register

Bit 0: The transmitter empty TXEPT flag is set when the transmitter shift register and UTXBUF are empty, and reset, when data is written to UTXBUF. It is set on SWRST.

Bit 1: The slave transmit control bit STC selects if the signal at STE pin is used in the master and slave.
STC = 0: Four pin mode of SPI is selected. The STE signal is used by the master to avoid bus conflicts or it is used in slave mode to control transmit and receive enable.
STC = 1: Three pin SPI mode. STE is not used in master mode nor in slave mode.

Bit 2: unused

Bit 3: unused

Bit 4,5: Source Select 0 and 1.

The source select bits define - only when master mode is selected - which clock source is used for the baud rate generation:

SSEL1,SSEL0	0	external clock selected, UCLK
	1	auxiliary clock selected, ACLK
	2, 3	main system clock selected, MCLK

In master mode (MM=1) an external clock at UCLK can not be selected since the master applies the UCLK signal for any slave.

In the slave mode the bits SSEL1 and SSEL0 are not relevant. The external clock UCLK is always used.

Bit 6,7: Clock polarity CKPL and Clock Phase CKPH.

The CKPL bit controls the polarity of the SPICLK signal.

CKPL = 0: the inactive level is low; data is output with the rising edge of UCLK; input data is latched with the falling edge of UCLK.

CKPL = 1: the inactive level is high; data is output with the falling edge of UCLK; input data is latched with the rising edge of SPICLK.

The CKPH bit controls the polarity of the SPICLK signal.

CKPH = 0: normal UCLK clocking scheme.

CKPH = 1: UCLK is delayed by one half cycle.

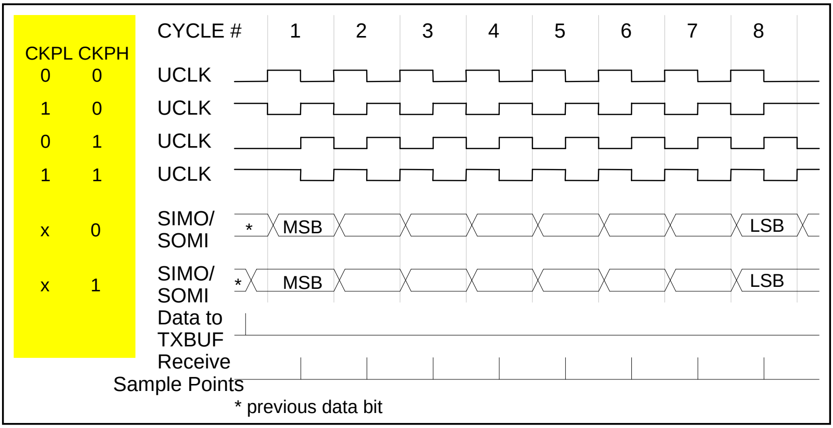


Figure 13.14: USART Clock Phase and Polarity

When operating with the CKPH bit set, the USART (synchronous mode) makes the first bit of data available after the transmit shift register is loaded and before the first edge of UCLK. Data is latched on the first edge of the UCLK and transmitted on the second edge in this mode.

13.3.3 Receive Control Register URCTL

The register URCTL controls the USART hardware associated with receiver operation and holds error conditions.

	7							0
URCTL 072h	FE	undef.	OE	undef.	unused	unused	undef.	undef.
	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0

Figure 13.15: USART Transmitter Control Register

Bit 0: undefined, driven by USART hardware
Bit 1: undefined, driven by USART hardware
Bit 2: unused
Bit 3: unused
Bit 4: undefined, driven by USART hardware
Bit 5: The overrun error flag bit OE is set when a character is transferred into URXBUF before the previous character has been read. The previous character is overwritten and lost. OE is reset by SWRST, system reset, by reading the URXBUF and by instruction.
Bit 6: undefined, driven by USART hardware
Bit 7: Frame error. The FE bit is set when a bus conflict stopped an active master with a negative transition of the signal applied to pin STE - only when 4-pin mode is selected. FE is reset by SWRST, system reset, by reading the URXBUF and by instruction.

13.3.4 Baud Rate Select and Modulation Control Registers

The baud rate generator uses the content of both baud rate select registers UBR1 and UBR0 to generate the bit timing for the serial data stream. The smallest division factor is two.

	7							0
UBR0 074h	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
	rw	rw	rw	rw	rw	rw	rw	rw

	7							0
UBR1 075h	2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^9	2^8
	rw	rw	rw	rw	rw	rw	rw	rw

Figure 13.16: USART Baud Rate Select Register

$$\text{Baudrate} = \frac{\text{BRCLK}}{\text{UBR} + \frac{1}{n}\sum_{i}^{n} mi} \qquad \text{with UBR= [UBR1,UBR0]}$$

The maximum baud rate that can be selected for transmission in master mode is half of the clock input frequency of the baud rate generator. In slave mode, it is determined by the external clock applied to UCLK.

The modulation control is not used for serial synchronous communication. It is recommended to keep it reset (bits m0 to m7 are 0).

	7							0
UMCTL 073h	m7	m6	m5	m4	m3	m2	m1	m0
	rw	rw	rw	rw	rw	rw	rw	rw

Figure 13.17: USART Modulation Control Register

13.3.5 USART Receive Data Buffer URXBUF

The receiver buffer URXBUF contains previous data from the receiver shift register. URXBUF is cleared by SWRST or PUC. Reading URXBUF resets the receive error bits and receive interrupt flag URXIFG.

	7							0
URXBUF 076h	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
	r	r	r	r	r	r	r	r

Figure 13.18: USART Receive Buffer

In 7-bit length mode the MSB of the URXBUF is always reset.

13.3.6 USART Transmit Data Buffer UTXBUF

The transmit buffer contains current data to be transmitted by the transmitter.

	7							0
UTXBUF 077h	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0

Figure 13.19: USART Transmit Buffer

The UTXIFG bit indicates that UTXBUF is ready to accept another character for transmission.
In master mode, the transmission will be initialized by writing data to UTXBUF. The transmission of this data is started immediately, if the transmitter shift register is empty or is going to be empty.
When seven bits/character is selected the data moved into the transmit buffer should be left adjusted since the MSB is shifted out first.