



# Universal Serial Communication Interface – UART Mode

**NOTE:** This chapter is an excerpt from the *MSP430x5xx and MSP430x6xx Family User's Guide*. The most recent version of the full user's guide is available from <http://www.ti.com/lit/pdf/slau208>.

The universal serial communication interface (USCI) supports multiple serial communication modes with one hardware module. This chapter discusses the operation of the asynchronous UART mode.

Topic	Page
1.1 Universal Serial Communication Interface (USCI) Overview .....	2
1.2 USCI Introduction – UART Mode .....	3
1.3 USCI Operation – UART Mode .....	5
1.4 USCI_A UART Mode Registers .....	21

## 1.1 Universal Serial Communication Interface (USCI) Overview

The USCI modules support multiple serial communication modes. Different USCI modules support different modes. Each different USCI module is named with a different letter. For example, USCI\_A is different from USCI\_B, etc. If more than one identical USCI module is implemented on one device, those modules are named with incrementing numbers. For example, if one device has two USCI\_A modules, they are named USCI\_A0 and USCI\_A1. See the device-specific data sheet to determine which USCI modules, if any, are implemented on which devices.

USCI\_Ax modules support:

- UART mode
- Pulse shaping for IrDA communications
- Automatic baud-rate detection for LIN communications
- SPI mode

USCI\_Bx modules support:

- I<sup>2</sup>C mode
- SPI mode

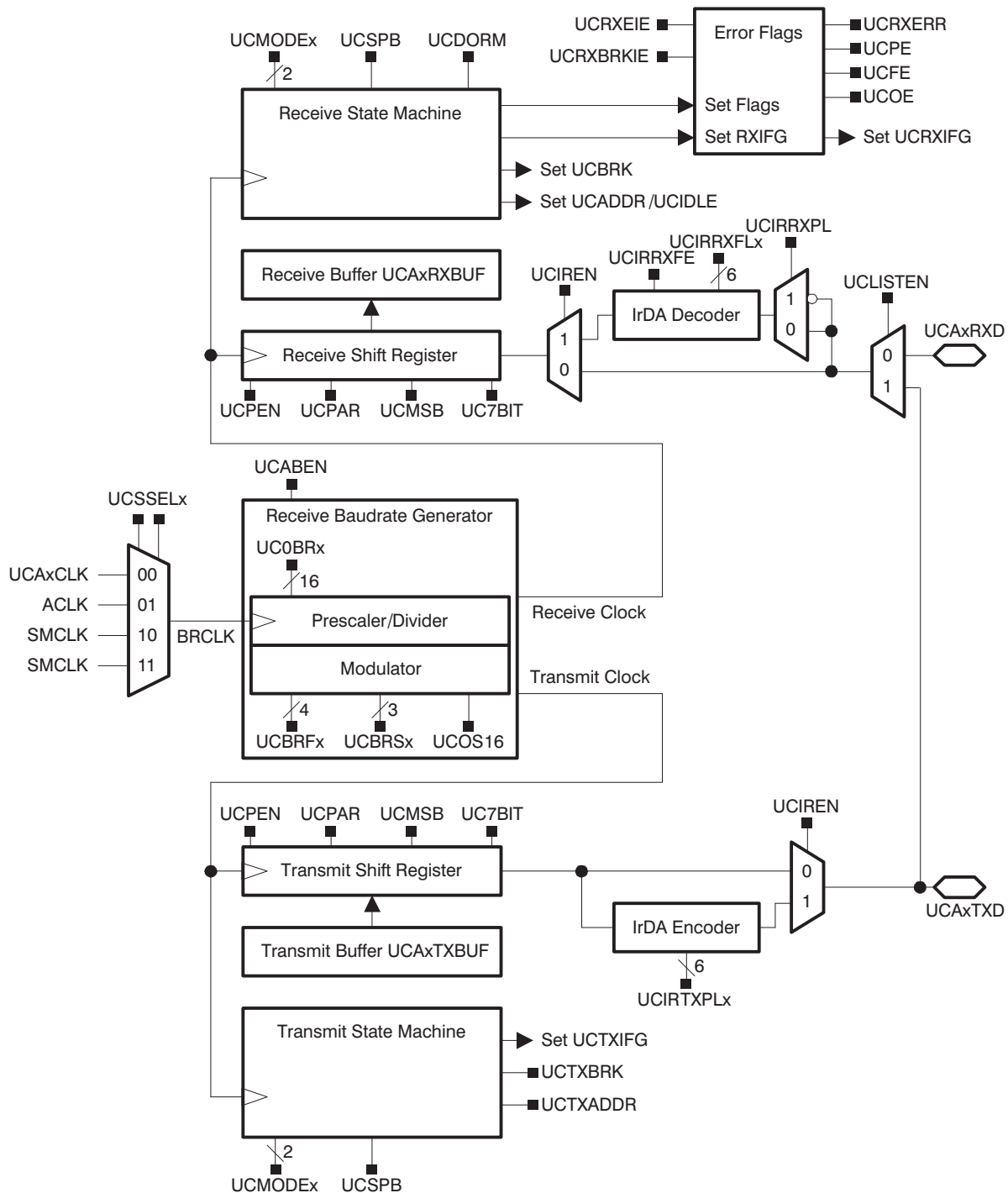
## 1.2 USCI Introduction – UART Mode

In asynchronous mode, the USCI\_Ax modules connect the device to an external system via two external pins, UCxRXD and UCxTXD. UART mode is selected when the UCSYNC bit is cleared.

UART mode features include:

- 7- or 8-bit data with odd, even, or non-parity
- Independent transmit and receive shift registers
- Separate transmit and receive buffer registers
- LSB-first or MSB-first data transmit and receive
- Built-in idle-line and address-bit communication protocols for multiprocessor systems
- Receiver start-edge detection for auto wake up from LPMx modes (wake up from LPMx.5 is not supported)
- Programmable baud rate with modulation for fractional baud-rate support
- Status flags for error detection and suppression
- Status flags for address detection
- Independent interrupt capability for receive and transmit

Figure 1-1 shows the USCI\_Ax when configured for UART mode.



**Figure 1-1. USCI\_Ax Block Diagram – UART Mode (UCSYNC = 0)**

## 1.3 USCI Operation – UART Mode

In UART mode, the USCI transmits and receives characters at a bit rate asynchronous to another device. Timing for each character is based on the selected baud rate of the USCI. The transmit and receive functions use the same baud-rate frequency.

### 1.3.1 USCI Initialization and Reset

The USCI is reset by a PUC or by setting the UCSWRST bit. After a PUC, the UCSWRST bit is automatically set, keeping the USCI in a reset condition. When set, the UCSWRST bit resets the UCRXIE, UCTXIE, UCRXIFG, UCRXERR, UCBRK, UCPE, UCOE, UCFE, UCSTOE, and UCBTOE bits, and sets the UCTXIFG bit. Clearing UCSWRST releases the USCI for operation.

To avoid unpredictable behavior, configure or reconfigure the USCI\_A module only when UCSWRST is set.

---

**NOTE:** Initializing or reconfiguring the USCI module

The recommended USCI initialization/reconfiguration process is:

1. Set UCSWRST (BIS.B  
#UCSWRST, &UCAxCTL1).
  2. Initialize all USCI registers with UCSWRST = 1 (including UCAxCTL1).
  3. Configure ports.
  4. Clear UCSWRST via software (BIC.B  
#UCSWRST, &UCAxCTL1).
  5. Enable interrupts (optional) via UCRXIE and/or UCTXIE.
- 

### 1.3.2 Character Format

The UART character format (see [Figure 1-2](#)) consists of a start bit, seven or eight data bits, an even/odd/no parity bit, an address bit (address-bit mode), and one or two stop bits. The UCMSB bit controls the direction of the transfer and selects LSB or MSB first. LSB first is typically required for UART communication.

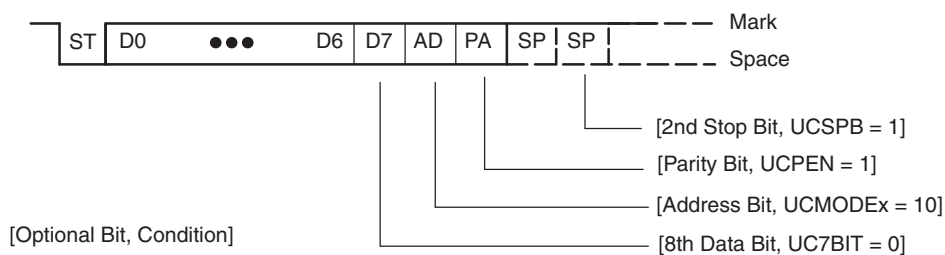


Figure 1-2. Character Format

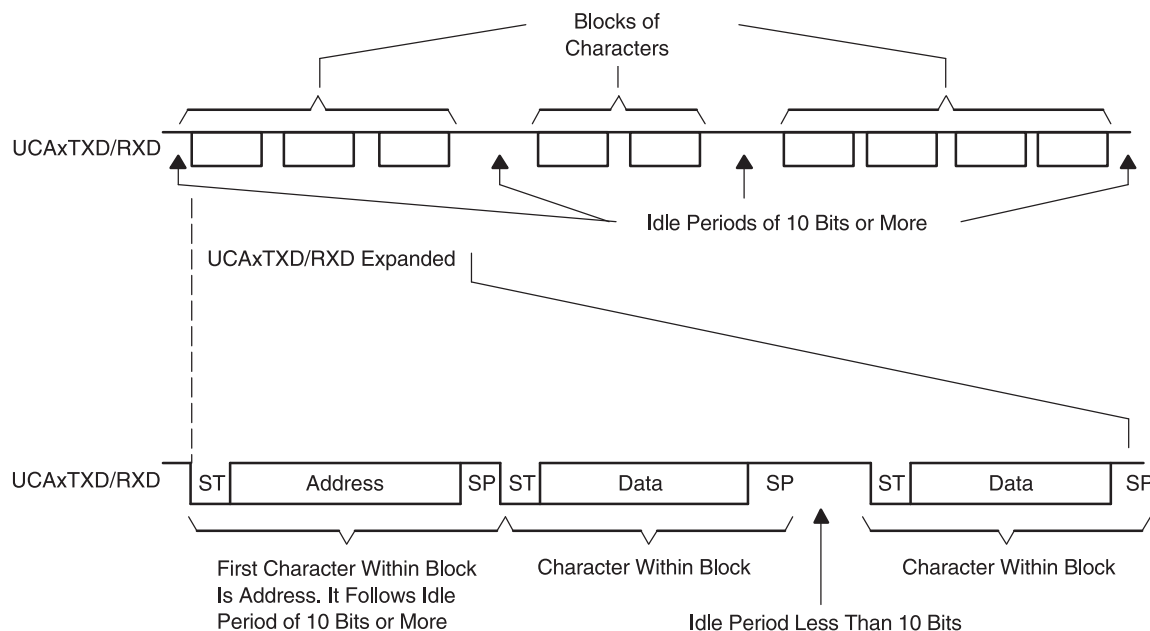
### 1.3.3 Asynchronous Communication Format

When two devices communicate asynchronously, no multiprocessor format is required for the protocol. When three or more devices communicate, the USCI supports the idle-line and address-bit multiprocessor communication formats.

#### 1.3.3.1 Idle-Line Multiprocessor Format

When UCMODEx = 01, the idle-line multiprocessor format is selected. Blocks of data are separated by an idle time on the transmit or receive lines (see [Figure 1-3](#)). An idle receive line is detected when ten or more continuous ones (marks) are received after the one or two stop bits of a character. The baud-rate generator is switched off after reception of an idle line until the next start edge is detected. When an idle line is detected, the UCIDLE bit is set.

The first character received after an idle period is an address character. The UCIDLE bit is used as an address tag for each block of characters. In idle-line multiprocessor format, this bit is set when a received character is an address.



**Figure 1-3. Idle-Line Format**

The UCDORM bit is used to control data reception in the idle-line multiprocessor format. When UCDORM = 1, all non-address characters are assembled but not transferred into the UCAxRXBUF, and interrupts are not generated. When an address character is received, the character is transferred into UCAxRXBUF, UCRXIFG is set, and any applicable error flag is set when UCRXEIE = 1. When UCRXEIE = 0 and an address character is received but has a framing error or parity error, the character is not transferred into UCAxRXBUF and UCRXIFG is not set.

If an address is received, user software can validate the address and must reset UCDORM to continue receiving data. If UCDORM remains set, only address characters are received. When UCDORM is cleared during the reception of a character, the receive interrupt flag is set after the reception completed. The UCDORM bit is not modified by the USCI hardware automatically.

For address transmission in idle-line multiprocessor format, a precise idle period can be generated by the USCI to generate address character identifiers on UCAxTXD. The double-buffered UCTXADDR flag indicates if the next character loaded into UCAxTXBUF is preceded by an idle line of 11 bits. UCTXADDR is automatically cleared when the start bit is generated.

#### 1.3.3.1.1 Transmitting an Idle Frame

The following procedure sends out an idle frame to indicate an address character followed by associated data:

1. Set UCTXADDR, then write the address character to UCAxTXBUF. UCAxTXBUF must be ready for new data (UCTXIFG = 1).  
This generates an idle period of exactly 11 bits followed by the address character. UCTXADDR is reset automatically when the address character is transferred from UCAxTXBUF into the shift register.
2. Write desired data characters to UCAxTXBUF. UCAxTXBUF must be ready for new data (UCTXIFG = 1).  
The data written to UCAxTXBUF is transferred to the shift register and transmitted as soon as the shift register is ready for new data.  
The idle-line time must not be exceeded between address and data transmission or between data transmissions. Otherwise, the transmitted data is misinterpreted as an address.

### 1.3.3.2 Address-Bit Multiprocessor Format

When UCMODEx = 10, the address-bit multiprocessor format is selected. Each processed character contains an extra bit used as an address indicator (see Figure 1-4). The first character in a block of characters carries a set address bit that indicates that the character is an address. The USCI UCADDR bit is set when a received character has its address bit set and is transferred to UCAxRXBUF.

The UCDORM bit is used to control data reception in the address-bit multiprocessor format. When UCDORM is set, data characters with address bit = 0 are assembled by the receiver but are not transferred to UCAxRXBUF and no interrupts are generated. When a character containing a set address bit is received, the character is transferred into UCAxRXBUF, UCRXIFG is set, and any applicable error flag is set when UCRXEIE = 1. When UCRXEIE = 0 and a character containing a set address bit is received but has a framing error or parity error, the character is not transferred into UCAxRXBUF and UCRXIFG is not set.

If an address is received, user software can validate the address and must reset UCDORM to continue receiving data. If UCDORM remains set, only address characters with address bit = 1 are received. The UCDORM bit is not modified by the USCI hardware automatically.

When UCDORM = 0, all received characters set the receive interrupt flag UCRXIFG. If UCDORM is cleared during the reception of a character, the receive interrupt flag is set after the reception is completed.

For address transmission in address-bit multiprocessor mode, the address bit of a character is controlled by the UCTXADDR bit. The value of the UCTXADDR bit is loaded into the address bit of the character transferred from UCAxTXBUF to the transmit shift register. UCTXADDR is automatically cleared when the start bit is generated.

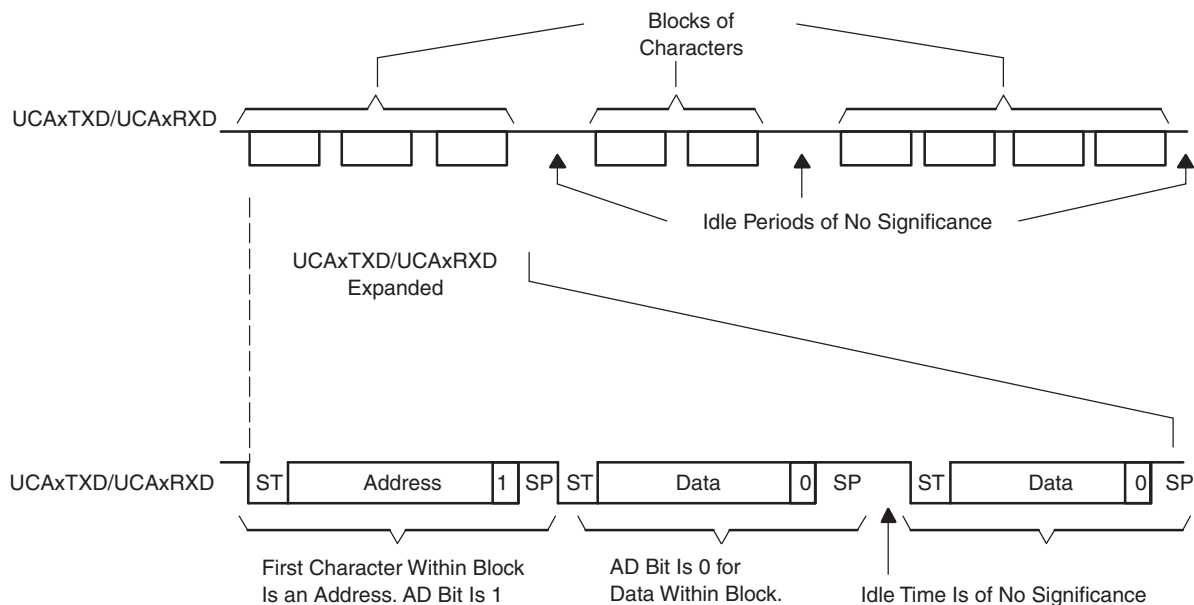


Figure 1-4. Address-Bit Multiprocessor Format

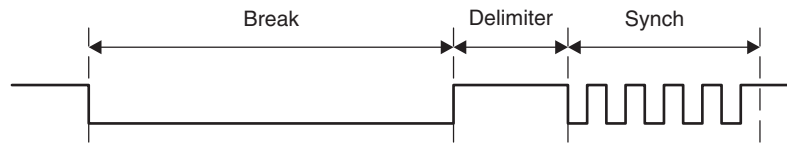
#### 1.3.3.2.1 Break Reception and Generation

When UCMODEx = 00, 01, or 10, the receiver detects a break when all data, parity, and stop bits are low, regardless of the parity, address mode, or other character settings. When a break is detected, the UCBRK bit is set. If the break interrupt enable bit (UCBRKIE) is set, the receive interrupt flag UCRXIFG is also set. In this case, the value in UCAxRXBUF is 0h, because all data bits were zero.

To transmit a break, set the UCTXBRK bit, then write 0h to UCAxTXBUF. UCAxTXBUF must be ready for new data (UCTXIFG = 1). This generates a break with all bits low. UCTXBRK is automatically cleared when the start bit is generated.

### 1.3.4 Automatic Baud-Rate Detection

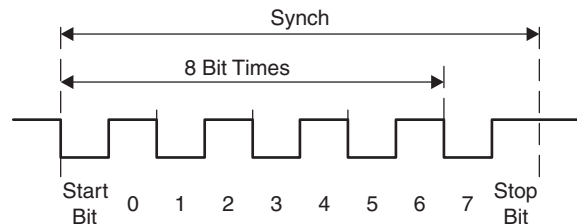
When UCMODEx = 11, UART mode with automatic baud-rate detection is selected. For automatic baud-rate detection, a data frame is preceded by a synchronization sequence that consists of a break and a synch field. A break is detected when 11 or more continuous zeros (spaces) are received. If the length of the break exceeds 21 bit times the break timeout error flag UCBTOE is set. The USCI can not transmit data while receiving the break/synch field. The synch field follows the break as shown in Figure 1-5.



**Figure 1-5. Auto Baud-Rate Detection – Break/Synch Sequence**

For LIN conformance, the character format should be set to eight data bits, LSB first, no parity, and one stop bit. No address bit is available.

The synch field consists of the data 055h inside a byte field (see Figure 1-6). The synchronization is based on the time measurement between the first falling edge and the last falling edge of the pattern. The transmit baud-rate generator is used for the measurement if automatic baud-rate detection is enabled by setting UCABDEN. Otherwise, the pattern is received but not measured. The result of the measurement is transferred into the baud-rate control registers (UCAxBR0, UCAxBR1, and UCAxMCTL). If the length of the synch field exceeds the measurable time, the synch timeout error flag UCSTOE is set.



**Figure 1-6. Auto Baud-Rate Detection – Synch Field**

The UCDORM bit is used to control data reception in this mode. When UCDORM is set, all characters are received but not transferred into the UCAxRXBUF, and interrupts are not generated. When a break/synch field is detected, the UCBRK flag is set. The character following the break/synch field is transferred into UCAxRXBUF and the UCRXIFG interrupt flag is set. Any applicable error flag is also set. If the UCBRKIE bit is set, reception of the break/synch sets the UCRXIFG. The UCBRK bit is reset by user software or by reading the receive buffer UCAxRXBUF.

When a break/synch field is received, user software must reset UCDORM to continue receiving data. If UCDORM remains set, only the character after the next reception of a break/synch field is received. The UCDORM bit is not modified by the USCI hardware automatically.

When UCDORM = 0, all received characters set the receive interrupt flag UCRXIFG. If UCDORM is cleared during the reception of a character, the receive interrupt flag is set after the reception is complete.

The counter used to detect the baud rate is limited to 07FFFh (32767) counts. This means the minimum baud rate detectable is 488 baud in oversampling mode and 30 baud in low-frequency mode.

The automatic baud-rate detection mode can be used in a full-duplex communication system with some restrictions. The USCI can not transmit data while receiving the break/synch field and, if a 0h byte with framing error is received, any data transmitted during this time gets corrupted. The latter case can be discovered by checking the received data and the UCFE bit.



### 1.3.4.1 Transmitting a Break/Synch Field

The following procedure transmits a break/synch field:

1. Set UCTXBRK with UMODEx = 11.
2. Write 055h to UCAxTXBUF. UCAxTXBUF must be ready for new data (UCTXIFG = 1).

This generates a break field of 13 bits followed by a break delimiter and the synch character. The length of the break delimiter is controlled with the UCDELIMx bits. UCTXBRK is reset automatically when the synch character is transferred from UCAxTXBUF into the shift register.

3. Write desired data characters to UCAxTXBUF. UCAxTXBUF must be ready for new data (UCTXIFG = 1).

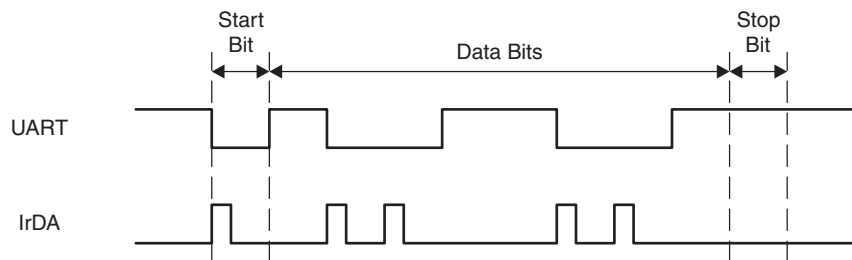
The data written to UCAxTXBUF is transferred to the shift register and transmitted as soon as the shift register is ready for new data.

### 1.3.5 IrDA Encoding and Decoding

When UCIREN is set, the IrDA encoder and decoder are enabled and provide hardware bit shaping for IrDA communication.

#### 1.3.5.1 IrDA Encoding

The encoder sends a pulse for every zero bit in the transmit bitstream coming from the UART (see Figure 1-7). The pulse duration is defined by UCIRTXPLx bits specifying the number of one-half clock periods of the clock selected by UCIRTXCLK.



**Figure 1-7. UART vs IrDA Data Format**

To set the pulse time of 3/16 bit period required by the IrDA standard, the BITCLK16 clock is selected with UCIRTXCLK = 1, and the pulse length is set to six one-half clock cycles with UCIRTXPLx = 6 – 1 = 5.

When UCIRTXCLK = 0, the pulse length  $t_{PULSE}$  is based on BRCLK and is calculated as:

$$UCIRTXPLx = t_{PULSE} \times 2 \times f_{BRCLK} - 1$$

When UCIRTXCLK = 0, the prescaler UCBRx must to be set to a value greater or equal to 5.

#### 1.3.5.2 IrDA Decoding

The decoder detects high pulses when UCIRRXPL = 0. Otherwise, it detects low pulses. In addition to the analog deglitch filter, an additional programmable digital filter stage can be enabled by setting UCIRRXFE. When UCIRRXFE is set, only pulses longer than the programmed filter length are passed. Shorter pulses are discarded. The equation to program the filter length UCIRRXFLx is:

$$UCIRRXFLx = (t_{PULSE} - t_{WAKE}) \times 2 \times f_{BRCLK} - 4$$

Where:

$t_{PULSE}$  = Minimum receive pulse width

$t_{WAKE}$  = Wake time from any low-power mode. Zero when the device is in active mode.

### 1.3.6 Automatic Error Detection

Glitch suppression prevents the USCI from being accidentally started. Any pulse on UCxRXD shorter than the deglitch time  $t_d$  (approximately 150 ns) is ignored (see the device-specific data sheet for parameters).

When a low period on UCxRXD exceeds  $t_d$ , a majority vote is taken for the start bit. If the majority vote fails to detect a valid start bit, the USCI halts character reception and waits for the next low period on UCxRXD. The majority vote is also used for each bit in a character to prevent bit errors.

The USCI module automatically detects framing errors, parity errors, overrun errors, and break conditions when receiving characters. The bits UCFE, UCPE, UCOE, and UCBRK are set when their respective condition is detected. When the error flags UCFE, UCPE, or UCOE are set, UCRXERR is also set. The error conditions are described in [Table 1-1](#).

**Table 1-1. Receive Error Conditions**

Error Condition	Error Flag	Description
Framing error	UCFE	A framing error occurs when a low stop bit is detected. When two stop bits are used, both stop bits are checked for framing error. When a framing error is detected, the UCFE bit is set.
Parity error	UCPE	A parity error is a mismatch between the number of 1s in a character and the value of the parity bit. When an address bit is included in the character, it is included in the parity calculation. When a parity error is detected, the UCPE bit is set.
Receive overrun	UCOE	An overrun error occurs when a character is loaded into UCxRXBUF before the prior character has been read. When an overrun occurs, the UCOE bit is set.
Break condition	UCBRK	When not using automatic baud-rate detection, a break is detected when all data, parity, and stop bits are low. When a break condition is detected, the UCBRK bit is set. A break condition can also set the interrupt flag UCRXIFG if the break interrupt enable UCBRKIE bit is set.

When UCRXEIE = 0 and a framing error or parity error is detected, no character is received into UCxRXBUF. When UCRXEIE = 1, characters are received into UCxRXBUF and any applicable error bit is set.

When any of the UCFE, UCPE, UCOE, UCBRK, or UCRXERR bit is set, the bit remains set until user software resets it or UCxRXBUF is read. UCOE must be reset by reading UCxRXBUF. Otherwise, it does not function properly. To detect overflows reliably the following flow is recommended. After a character was received and UCRXIFG is set, first read UCxSTAT to check the error flags including the overflow flag UCOE. Read UCxRXBUF next. This clears all error flags except UCOE, if UCxRXBUF was overwritten between the read access to UCxSTAT and to UCxRXBUF. Therefore, the UCOE flag should be checked after reading UCxRXBUF to detect this condition. Note that, in this case, the UCRXERR flag is not set.

### 1.3.7 USCI Receive Enable

The USCI module is enabled by clearing the UCSWRST bit and the receiver is ready and in an idle state. The receive baud rate generator is in a ready state but is not clocked nor producing any clocks.

The falling edge of the start bit enables the baud rate generator and the UART state machine checks for a valid start bit. If no valid start bit is detected the UART state machine returns to its idle state and the baud rate generator is turned off again. If a valid start bit is detected, a character is received.

When the idle-line multiprocessor mode is selected with UCMODEx = 01 the UART state machine checks for an idle line after receiving a character. If a start bit is detected another character is received. Otherwise the UCIDLE flag is set after 10 ones are received and the UART state machine returns to its idle state and the baud rate generator is turned off.

#### 1.3.7.1 Receive Data Glitch Suppression

Glitch suppression prevents the USCI from being accidentally started. Any glitch on UCAXRXD shorter than the deglitch time  $t_d$  (approximately 150 ns) is ignored by the USCI, and further action is initiated as shown in Figure 1-8 (see the device-specific data sheet for parameters).

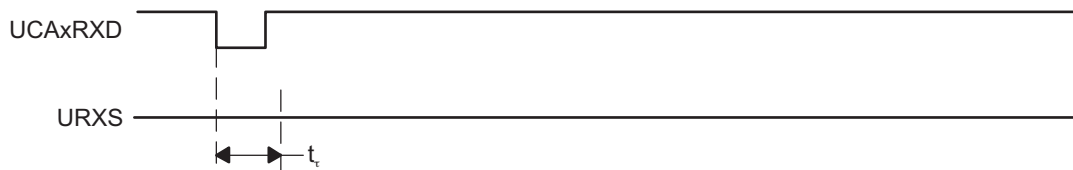


Figure 1-8. Glitch Suppression, USCI Receive Not Started

When a glitch is longer than  $t_d$  or a valid start bit occurs on UCAXRXD, the USCI receive operation is started and a majority vote is taken (see Figure 1-9). If the majority vote fails to detect a start bit, the USCI halts character reception.

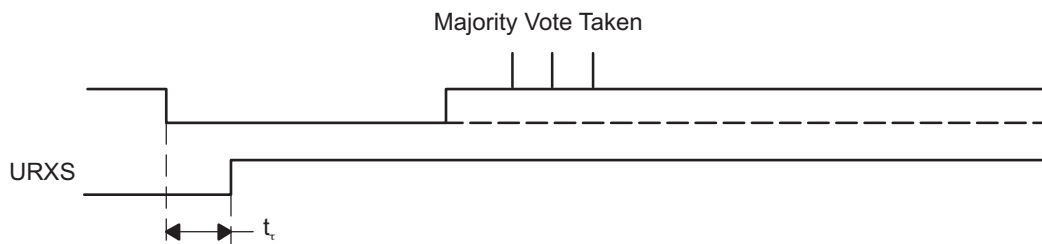


Figure 1-9. Glitch Suppression, USCI Activated

### 1.3.8 USCI Transmit Enable

The USCI module is enabled by clearing the UCSWRST bit and the transmitter is ready and in an idle state. The transmit baud-rate generator is ready but is not clocked nor producing any clocks.

A transmission is initiated by writing data to UCAXTXBUF. When this occurs, the baud-rate generator is enabled, and the data in UCAXTXBUF is moved to the transmit shift register on the next BITCLK after the transmit shift register is empty. UCTXIFG is set when new data can be written into UCAXTXBUF.

Transmission continues as long as new data is available in UCAXTXBUF at the end of the previous byte transmission. If new data is not in UCAXTXBUF when the previous byte has transmitted, the transmitter returns to its idle state and the baud-rate generator is turned off.

### 1.3.9 UART Baud-Rate Generation

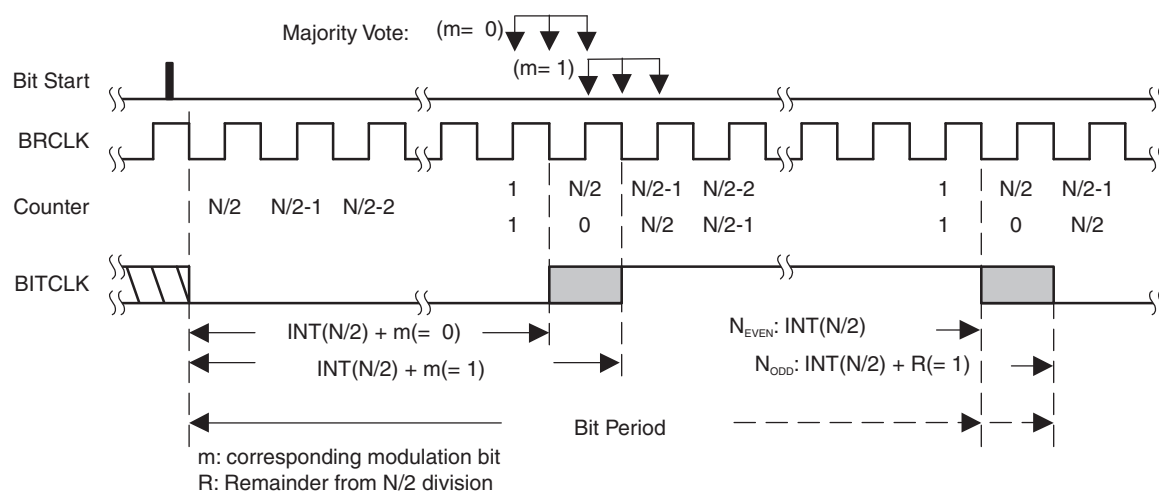
The USCI baud-rate generator is capable of producing standard baud rates from nonstandard source frequencies. It provides two modes of operation selected by the UCOS16 bit. The baud-rate is generated using the BRCLK that can be sourced by the external clock UCAxCLK, or the internal clocks ACLK or SMCLK depending on the UCSSELx settings.

#### 1.3.9.1 Low-Frequency Baud-Rate Generation

The low-frequency mode is selected when UCOS16 = 0. This mode allows generation of baud rates from low frequency clock sources (for example, 9600 baud from a 32768-Hz crystal). By using a lower input frequency, the power consumption of the module is reduced. Using this mode with higher frequencies and higher prescaler settings causes the majority votes to be taken in an increasingly smaller window and, thus, decrease the benefit of the majority vote.

In low-frequency mode, the baud-rate generator uses one prescaler and one modulator to generate bit clock timing. This combination supports fractional divisors for baud-rate generation. In this mode, the maximum USCI baud rate is one-third the UART source clock frequency BRCLK.

Timing for each bit is shown in Figure 1-10. For each bit received, a majority vote is taken to determine the bit value. These samples occur at the  $N/2 - 1/2$ ,  $N/2$ , and  $N/2 + 1/2$  BRCLK periods, where N is the number of BRCLKs per BITCLK.



**Figure 1-10. BITCLK Baud-Rate Timing With UCOS16 = 0**

Modulation is based on the UCBSRx setting (see Table 1-2). A 1 in the table indicates that  $m = 1$  and the corresponding BITCLK period is one BRCLK period longer than a BITCLK period with  $m = 0$ . The modulation wraps around after eight bits but restarts with each new start bit.

**Table 1-2. BITCLK Modulation Pattern**

UCBSRx	Bit 0 (Start Bit)	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0
2	0	1	0	0	0	1	0	0
3	0	1	0	1	0	1	0	0
4	0	1	0	1	0	1	0	1
5	0	1	1	1	0	1	0	1
6	0	1	1	1	0	1	1	1
7	0	1	1	1	1	1	1	1

### 1.3.9.2 Oversampling Baud-Rate Generation

The oversampling mode is selected when UCOS16 = 1. This mode supports sampling a UART bitstream with higher input clock frequencies. This results in majority votes that are always 1/16 of a bit clock period apart. This mode also easily supports IrDA pulses with a 3/16 bit time when the IrDA encoder and decoder are enabled.

This mode uses one prescaler and one modulator to generate the BITCLK16 clock that is 16 times faster than the BITCLK. An additional divider and modulator stage generates BITCLK from BITCLK16. This combination supports fractional divisions of both BITCLK16 and BITCLK for baud-rate generation. In this mode, the maximum USCI baud rate is 1/16 the UART source clock frequency BRCLK. When UCBRx is set to 0 or 1, the first prescaler and modulator stage is bypassed and BRCLK is equal to BITCLK16 – in this case, no modulation for the BITCLK16 is possible and, thus, the UCBRFx bits are ignored.

Modulation for BITCLK16 is based on the UCBRFx setting (see [Table 1-3](#)). A 1 in the table indicates that the corresponding BITCLK16 period is one BRCLK period longer than the periods m = 0. The modulation restarts with each new bit timing.

Modulation for BITCLK is based on the UCBRSx setting (see [Table 1-2](#)) as previously described.

**Table 1-3. BITCLK16 Modulation Pattern**

UCBRFx	Number of BITCLK16 Clocks After Last Falling BITCLK Edge															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
00h	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
01h	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
02h	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
03h	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1
04h	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	1
05h	0	1	1	1	0	0	0	0	0	0	0	0	0	0	1	1
06h	0	1	1	1	0	0	0	0	0	0	0	0	0	1	1	1
07h	0	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1
08h	0	1	1	1	1	0	0	0	0	0	0	0	1	1	1	1
09h	0	1	1	1	1	1	0	0	0	0	0	0	1	1	1	1
0Ah	0	1	1	1	1	1	0	0	0	0	0	1	1	1	1	1
0Bh	0	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1
0Ch	0	1	1	1	1	1	1	0	0	0	1	1	1	1	1	1
0Dh	0	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1
0Eh	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
0Fh	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

### 1.3.10 Setting a Baud Rate

For a given BRCLK clock source, the baud rate used determines the required division factor N:

$$N = f_{BRCLK} / \text{Baudrate}$$

The division factor N is often a noninteger value, thus, at least one divider and one modulator stage is used to meet the factor as closely as possible.

If N is equal or greater than 16, the oversampling baud-rate generation mode can be chosen by setting UCOS16.

#### 1.3.10.1 Low-Frequency Baud-Rate Mode Setting

In low-frequency mode, the integer portion of the divisor is realized by the prescaler:

$$UCBRx = \text{INT}(N)$$

and the fractional portion is realized by the modulator with the following nominal formula:

$$UCBRsX = \text{round}[(N - \text{INT}(N)) \times 8]$$

Incrementing or decrementing the UCBRSx setting by one count may give a lower maximum bit error for any given bit. To determine if this is the case, a detailed error calculation must be performed for each bit for each UCBRSx setting.

#### 1.3.10.2 Oversampling Baud-Rate Mode Setting

In the oversampling mode, the prescaler is set to:

$$UCBRx = \text{INT}(N/16)$$

and the first stage modulator is set to:

$$UCBRFx = \text{round}[(N/16 - \text{INT}(N/16)) \times 16]$$

When greater accuracy is required, the UCBRSx modulator can also be implemented with values from 0 to 7. To find the setting that gives the lowest maximum bit error rate for any given bit, a detailed error calculation must be performed for all settings of UCBRSx from 0 to 7 with the initial UCBRFx setting, and with the UCBRFx setting incremented and decremented by one.

### 1.3.11 Transmit Bit Timing

The timing for each character is the sum of the individual bit timings. Using the modulation features of the baud-rate generator reduces the cumulative bit error. The individual bit error can be calculated using the following steps.

#### 1.3.11.1 Low-Frequency Baud-Rate Mode Bit Timing

In low-frequency mode, calculate the length of bit i  $T_{\text{bit,TX}}[i]$  based on the UCBRx and UCBRSx settings:

$$T_{\text{bit,TX}}[i] = (1/f_{BRCLK})(UCBRx + m_{UCBRSx}[i])$$

Where:

$$m_{UCBRSx}[i] = \text{Modulation of bit } i \text{ from Table 1-2}$$

#### 1.3.11.2 Oversampling Baud-Rate Mode Bit Timing

In oversampling baud-rate mode, calculate the length of bit i  $T_{\text{bit,TX}}[i]$  based on the baud-rate generator UCBRx, UCBRFx and UCBRSx settings:

$$T_{\text{bit,TX}}[i] = \frac{1}{f_{BRCLK}} \left( (16 + m_{UCBRSx}[i]) \times UCBRx + \sum_{j=0}^{15} m_{UCBRFx}[j] \right)$$

Where:

$$\sum_{j=0}^{15} m_{UCBRFx}[j] = \text{Sum of ones from the corresponding row in Table 1-3}$$

$$m_{UCBRSx}[i] = \text{Modulation of bit } i \text{ from Table 1-2}$$

This results in an end-of-bit time  $t_{\text{bit,TX}[i]}$  equal to the sum of all previous and the current bit times:

$$T_{\text{bit,TX}[i]} = \sum_{j=0}^i T_{\text{bit,TX}[j]}$$

To calculate bit error, this time is compared to the ideal bit time  $t_{\text{bit,ideal,TX}[i]}$ :

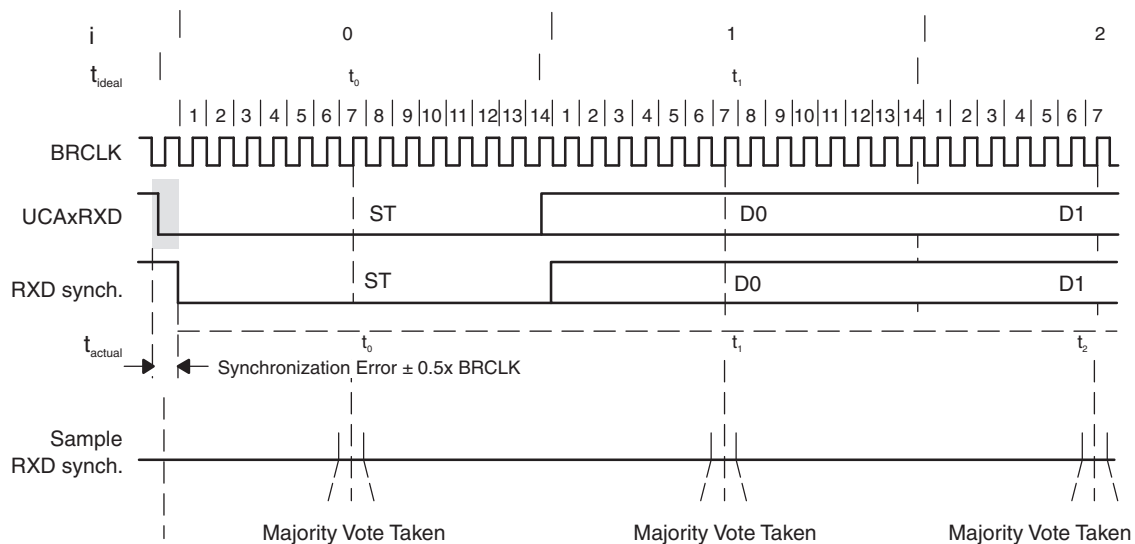
$$t_{\text{bit,ideal,TX}[i]} = (1/\text{Baudrate})(i + 1)$$

This results in an error normalized to one ideal bit time (1/baudrate):

$$\text{Error}_{\text{TX}[i]} = (t_{\text{bit,TX}[i]} - t_{\text{bit,ideal,TX}[i]}) \times \text{Baudrate} \times 100\%$$

### 1.3.12 Receive Bit Timing

Receive timing error consists of two error sources. The first is the bit-to-bit timing error similar to the transmit bit timing error. The second is the error between a start edge occurring and the start edge being accepted by the USCI module. Figure 1-11 shows the asynchronous timing errors between data on the UCAxRXD pin and the internal baud-rate clock. This results in an additional synchronization error. The synchronization error  $t_{\text{SYNC}}$  is between  $-0.5$  BRCLKs and  $+0.5$  RCLKs, independent of the selected baud-rate generation mode.



**Figure 1-11. Receive Error**

The ideal sampling time  $t_{\text{bit,ideal,RX}[i]}$  is in the middle of a bit period:

$$t_{\text{bit,ideal,RX}[i]} = (1/\text{Baudrate})(i + 0.5)$$

The real sampling time,  $t_{\text{bit,RX}[i]}$ , is equal to the sum of all previous bits according to the formulas shown in the transmit timing section, plus one-half BITCLK for the current bit  $i$ , plus the synchronization error  $t_{\text{SYNC}}$ .

This results in the following  $t_{\text{bit,RX}[i]}$  for the low-frequency baud-rate mode:

$$t_{\text{bit,RX}[i]} = t_{\text{SYNC}} + \sum_{j=0}^{i-1} T_{\text{bit,RX}[j]} + \frac{1}{f_{\text{BRCLK}}} \left( \text{INT}(\frac{1}{2} \text{UCBRx}) + m_{\text{UCBRsx}[i]} \right)$$

Where:

$$T_{\text{bit,RX}[i]} = (1/f_{\text{BRCLK}})(\text{UCBRx} + m_{\text{UCBRsx}[i]})$$

$$m_{\text{UCBRsx}[i]} = \text{Modulation of bit } i \text{ from Table 1-2}$$

For the oversampling baud-rate mode, the sampling time  $t_{\text{bit,RX}}[i]$  of bit  $i$  is calculated by:

$$t_{\text{bit,RX}}[i] = t_{\text{SYNC}} + \sum_{j=0}^{i-1} T_{\text{bit,RX}}[j] + \frac{1}{f_{\text{BRCLK}}} \left( (8 + m_{\text{UCBRsx}}[i]) \times \text{UCBRx} + \sum_{j=0}^{7 + m_{\text{UCBRsx}}[i]} m_{\text{UCBRfx}}[j] \right)$$

Where:

$$T_{\text{bit,RX}}[i] = \frac{1}{f_{\text{BRCLK}}} \left( (16 + m_{\text{UCBRsx}}[i]) \times \text{UCBRx} + \sum_{j=0}^{15} m_{\text{UCBRfx}}[j] \right)$$

$\sum_{j=0}^{7 + m_{\text{UCBRsx}}[i]} m_{\text{UCBRfx}}[j]$  = Sum of ones from columns 0 to  $(7 + m_{\text{UCBRsx}}[i])$  from the corresponding row in [Table 1-3](#).

$m_{\text{UCBRsx}}[i]$  = Modulation of bit  $i$  from [Table 1-2](#)

This results in an error normalized to one ideal bit time (1/baudrate) according to the following formula:

$$\text{Error}_{\text{RX}}[i] = (t_{\text{bit,RX}}[i] - t_{\text{bit,ideal,RX}}[i]) \times \text{Baudrate} \times 100\%$$

### 1.3.13 Typical Baud Rates and Errors

Standard baud-rate data for UCBRx, UCBRSx, and UCBRFx are listed in [Table 1-4](#) and [Table 1-5](#) for a 32,768-Hz crystal sourcing ACLK and typical SMCLK frequencies. Make sure that the selected BRCLK frequency does not exceed the device-specific maximum USCI input frequency (see the device-specific data sheet).

The receive error is the accumulated time versus the ideal scanning time in the middle of each bit. The worst-case error is given for the reception of an 8-bit character with parity and one stop bit including synchronization error.

The transmit error is the accumulated timing error versus the ideal time of the bit period. The worst-case error is given for the transmission of an 8-bit character with parity and stop bit.

**Table 1-4. Commonly Used Baud Rates, Settings, and Errors, UCOS16 = 0**

BRCLK Frequency (Hz)	Baud Rate (baud)	UCBRx	UCBRSx	UCBRFx	Maximum TX Error (%)		Maximum RX Error (%)	
32 768	1200	27	2	0	-2.8	1.4	-5.9	2.0
32 768	2400	13	6	0	-4.8	6.0	-9.7	8.3
32 768	4800	6	7	0	-12.1	5.7	-13.4	19.0
32 768	9600	3	3	0	-21.1	15.2	-44.3	21.3
1 000 000	9600	104	1	0	-0.5	0.6	-0.9	1.2
1 000 000	19200	52	0	0	-1.8	0	-2.6	0.9
1 000 000	38400	26	0	0	-1.8	0	-3.6	1.8
1 000 000	57600	17	3	0	-2.1	4.8	-6.8	5.8
1 000 000	115200	8	6	0	-7.8	6.4	-9.7	16.1
1 048 576	9600	109	2	0	-0.2	0.7	-1.0	0.8
1 048 576	19200	54	5	0	-1.1	1.0	-1.5	2.5
1 048 576	38400	27	2	0	-2.8	1.4	-5.9	2.0
1 048 576	57600	18	1	0	-4.6	3.3	-6.8	6.6
1 048 576	115200	9	1	0	-1.1	10.7	-11.5	11.3
4 000 000	9600	416	6	0	-0.2	0.2	-0.2	0.4
4 000 000	19200	208	3	0	-0.2	0.5	-0.3	0.8
4 000 000	38400	104	1	0	-0.5	0.6	-0.9	1.2
4 000 000	57600	69	4	0	-0.6	0.8	-1.8	1.1
4 000 000	115200	34	6	0	-2.1	0.6	-2.5	3.1
4 000 000	230400	17	3	0	-2.1	4.8	-6.8	5.8
4 194 304	9600	436	7	0	-0.3	0	-0.3	0.2



**Table 1-4. Commonly Used Baud Rates, Settings, and Errors, UCOS16 = 0 (continued)**

BRCLK Frequency (Hz)	Baud Rate (baud)	UCBRx	UCBR5x	UCBRFx	Maximum TX Error (%)		Maximum RX Error (%)	
4 194 304	19200	218	4	0	-0.2	0.2	-0.3	0.6
4 194 304	57600	72	7	0	-1.1	0.6	-1.3	1.9
4 194 304	115200	36	3	0	-1.9	1.5	-2.7	3.4
8 000 000	9600	833	2	0	-0.1	0	-0.2	0.1
8 000 000	19200	416	6	0	-0.2	0.2	-0.2	0.4
8 000 000	38400	208	3	0	-0.2	0.5	-0.3	0.8
8 000 000	57600	138	7	0	-0.7	0	-0.8	0.6
8 000 000	115200	69	4	0	-0.6	0.8	-1.8	1.1
8 000 000	230400	34	6	0	-2.1	0.6	-2.5	3.1
8 000 000	460800	17	3	0	-2.1	4.8	-6.8	5.8
8 388 608	9600	873	7	0	-0.1	0.06	-0.2	0.1
8 388 608	19200	436	7	0	-0.3	0	-0.3	0.2
8 388 608	57600	145	5	0	-0.5	0.3	-1.0	0.5
8 388 608	115200	72	7	0	-1.1	0.6	-1.3	1.9
12 000 000	9600	1250	0	0	0	0	-0.05	0.05
12 000 000	19200	625	0	0	0	0	-0.2	0
12 000 000	38400	312	4	0	-0.2	0	-0.2	0.2
12 000 000	57600	208	2	0	-0.5	0.2	-0.6	0.5
12 000 000	115200	104	1	0	-0.5	0.6	-0.9	1.2
12 000 000	230400	52	0	0	-1.8	0	-2.6	0.9
12 000 000	460800	26	0	0	-1.8	0	-3.6	1.8
16 000 000	9600	1666	6	0	-0.05	0.05	-0.05	0.1
16 000 000	19200	833	2	0	-0.1	0.05	-0.2	0.1
16 000 000	38400	416	6	0	-0.2	0.2	-0.2	0.4
16 000 000	57600	277	7	0	-0.3	0.3	-0.5	0.4
16 000 000	115200	138	7	0	-0.7	0	-0.8	0.6
16 000 000	230400	69	4	0	-0.6	0.8	-1.8	1.1
16 000 000	460800	34	6	0	-2.1	0.6	-2.5	3.1
16 777 216	9600	1747	5	0	-0.04	0.03	-0.08	0.05
16 777 216	19200	873	7	0	-0.09	0.06	-0.2	0.1
16 777 216	57600	291	2	0	-0.2	0.2	-0.5	0.2
16 777 216	115200	145	5	0	-0.5	0.3	-1.0	0.5
20 000 000	9600	2083	2	0	-0.05	0.02	-0.09	0.02
20 000 000	19200	1041	6	0	-0.06	0.06	-0.1	0.1
20 000 000	38400	520	7	0	-0.2	0.06	-0.2	0.2
20 000 000	57600	347	2	0	-0.06	0.2	-0.3	0.3
20 000 000	115200	173	5	0	-0.4	0.3	-0.8	0.5
20 000 000	230400	86	7	0	-1.0	0.6	-1.0	1.7
20 000 000	460800	43	3	0	-1.4	1.3	-3.3	1.8

**Table 1-5. Commonly Used Baud Rates, Settings, and Errors, UCOS16 = 1**

BRCLK Frequency (Hz)	Baud Rate (baud)	UCBRx	UCBRsX	UCBRFx	Maximum TX Error (%)		Maximum RX Error (%)	
1 000 000	9600	6	0	8	-1.8	0	-2.2	0.4
1 000 000	19200	3	0	4	-1.8	0	-2.6	0.9
1 048 576	9600	6	0	13	-2.3	0	-2.2	0.8
1 048 576	19200	3	1	6	-4.6	3.2	-5.0	4.7
4 000 000	9600	26	0	1	0	0.9	0	1.1
4 000 000	19200	13	0	0	-1.8	0	-1.9	0.2
4 000 000	38400	6	0	8	-1.8	0	-2.2	0.4
4 000 000	57600	4	5	3	-3.5	3.2	-1.8	6.4
4 000 000	115200	2	3	2	-2.1	4.8	-2.5	7.3
4 194 304	9600	27	0	5	0	0.2	0	0.5
4 194 304	19200	13	0	10	-2.3	0	-2.4	0.1
4 194 304	57600	4	4	7	-2.5	2.5	-1.3	5.1
4 194 304	115200	2	6	3	-3.9	2.0	-1.9	6.7
8 000 000	9600	52	0	1	-0.4	0	-0.4	0.1
8 000 000	19200	26	0	1	0	0.9	0	1.1
8 000 000	38400	13	0	0	-1.8	0	-1.9	0.2
8 000 000	57600	8	0	11	0	0.88	0	1.6
8 000 000	115200	4	5	3	-3.5	3.2	-1.8	6.4
8 000 000	230400	2	3	2	-2.1	4.8	-2.5	7.3
8 388 608	9600	54	0	10	0	0.2	-0.05	0.3
8 388 608	19200	27	0	5	0	0.2	0	0.5
8 388 608	57600	9	0	2	0	2.8	-0.2	3.0
8 388 608	115200	4	4	7	-2.5	2.5	-1.3	5.1
12 000 000	9600	78	0	2	0	0	-0.05	0.05
12 000 000	19200	39	0	1	0	0	0	0.2
12 000 000	38400	19	0	8	-1.8	0	-1.8	0.1
12 000 000	57600	13	0	0	-1.8	0	-1.9	0.2
12 000 000	115200	6	0	8	-1.8	0	-2.2	0.4
12 000 000	230400	3	0	4	-1.8	0	-2.6	0.9
16 000 000	9600	104	0	3	0	0.2	0	0.3
16 000 000	19200	52	0	1	-0.4	0	-0.4	0.1
16 000 000	38400	26	0	1	0	0.9	0	1.1
16 000 000	57600	17	0	6	0	0.9	-0.1	1.0
16 000 000	115200	8	0	11	0	0.9	0	1.6
16 000 000	230400	4	5	3	-3.5	3.2	-1.8	6.4
16 000 000	460800	2	3	2	-2.1	4.8	-2.5	7.3
16 777 216	9600	109	0	4	0	0.2	-0.02	0.3
16 777 216	19200	54	0	10	0	0.2	-0.05	0.3
16 777 216	57600	18	0	3	-1.0	0	-1.0	0.3
16 777 216	115200	9	0	2	0	2.8	-0.2	3.0
20 000 000	9600	130	0	3	-0.2	0	-0.2	0.04
20 000 000	19200	65	0	2	0	0.4	-0.03	0.4
20 000 000	38400	32	0	9	0	0.4	0	0.5
20 000 000	57600	21	0	11	-0.7	0	-0.7	0.3
20 000 000	115200	10	0	14	0	2.5	-0.2	2.6
20 000 000	230400	5	0	7	0	2.5	0	3.5

**Table 1-5. Commonly Used Baud Rates, Settings, and Errors, UCOS16 = 1 (continued)**

BRCLK Frequency (Hz)	Baud Rate (baud)	UCBRx	UCBRSx	UCBRFx	Maximum TX Error (%)		Maximum RX Error (%)	
20 000 000	460800	2	6	10	-3.2	1.8	-2.8	4.6

### 1.3.14 Using the USCI Module in UART Mode With Low-Power Modes

The USCI module provides automatic clock activation for use with low-power modes. When the USCI clock source is inactive because the device is in a low-power mode, the USCI module automatically activates it when needed, regardless of the control-bit settings for the clock source. The clock remains active until the USCI module returns to its idle condition. After the USCI module returns to the idle condition, control of the clock source reverts to the settings of its control bits.

### 1.3.15 USCI Interrupts in UART Mode

The USCI has only one interrupt vector that is shared for transmission and for reception. USCI\_Ax and USC\_Bx do not share the same interrupt vector.

#### 1.3.15.1 UART Transmit Interrupt Operation

The UCTXIFG interrupt flag is set by the transmitter to indicate that UCAxTXBUF is ready to accept another character. An interrupt request is generated if UCTXIE and GIE are also set. UCTXIFG is automatically reset if a character is written to UCAxTXBUF.

UCTXIFG is set after a PUC or when UCSWRST = 1. UCTXIE is reset after a PUC or when UCSWRST = 1.

#### 1.3.15.2 UART Receive Interrupt Operation

The UCRXIFG interrupt flag is set each time a character is received and loaded into UCAxRXBUF. An interrupt request is generated if UCRXIE and GIE are also set. UCRXIFG and UCRXIE are reset by a system reset PUC signal or when UCSWRST = 1. UCRXIFG is automatically reset when UCAxRXBUF is read.

Additional interrupt control features include:

- When UCAxRXEIE = 0, erroneous characters do not set UCRXIFG.
- When UCDORM = 1, nonaddress characters do not set UCRXIFG in multiprocessor modes. In plain UART mode, no characters are set UCRXIFG.
- When UCBRKIE = 1, a break condition sets the UCBRK bit and the UCRXIFG flag.

#### 1.3.15.3 UCAxIV, Interrupt Vector Generator

The USCI interrupt flags are prioritized and combined to source a single interrupt vector. The interrupt vector register UCAxIV is used to determine which flag requested an interrupt. The highest-priority enabled interrupt generates a number in the UCAxIV register that can be evaluated or added to the program counter to automatically enter the appropriate software routine. Disabled interrupts do not affect the UCAxIV value.

Any access, read or write, of the UCAxIV register automatically resets the highest-pending interrupt flag. If another interrupt flag is set, another interrupt is immediately generated after servicing the initial interrupt.

### 1.3.15.3.1 UCAXIV Software Example

The following software example shows the recommended use of UCAXIV. The UCAXIV value is added to the PC to automatically jump to the appropriate routine. The following example is given for USCI\_A0.

```
USCI_UART_ISR
    ADD    &UCA0IV, PC    ; Add offset to jump table
    RETI                                ; Vector 0: No interrupt
    JMP    RXIFG_ISR      ; Vector 2: RXIFG
TXIFG_ISR
    ...                                ; Task starts here
    RETI                                ; Return
RXIFG_ISR
    ...                                ; Vector 2
    ...                                ; Task starts here
    RETI                                ; Return
```

### 1.3.16 DMA Operation

In devices with a DMA controller, the eUSCI module can trigger DMA transfers when the transmit buffer UCAXTXBUF is empty or when data was received in the UCAXRXBUF buffer. The DMA trigger signals correspond to the UCTXIFG transmit interrupt flag and the UCRXIFG receive interrupt flag, respectively. The interrupt functionality must be disabled for the selected DMA triggers with UCTXIE = 0 and UCRXIE = 0.

A DMA read access to UCAXRXBUF has the same effects as a CPU (software) read: all error flags (UCRXERR, UCFE, UCPE, UCOE, and UCBRK) are cleared after the read. Thus these errors might go unnoticed.

## 1.4 USCI\_A UART Mode Registers

The USCI registers applicable in UART mode listed in [Table 1-6](#). The base address can be found in the device-specific data sheet. The address offsets are listed in [Table 1-6](#).

**Table 1-6. USCI\_A UART Mode Registers**

Offset	Acronym	Register Name	Type	Access	Reset	Section
00h	UCAxCTLW0	USCI_Ax Control Word 0	Read/write	Word	0001h	
00h	UCAxCTL1	USCI_Ax Control 1	Read/write	Byte	01h	<a href="#">Section 1.4.2</a>
01h	UCAxCTL0	USCI_Ax Control 0	Read/write	Byte	00h	<a href="#">Section 1.4.1</a>
06h	UCAxBRW	USCI_Ax Baud Rate Control Word	Read/write	Word	0000h	
06h	UCAxBR0	USCI_Ax Baud Rate Control 0	Read/write	Byte	00h	<a href="#">Section 1.4.3</a>
07h	UCAxBR1	USCI_Ax Baud Rate Control 1	Read/write	Byte	00h	<a href="#">Section 1.4.4</a>
08h	UCAxMCTL	USCI_Ax Modulation Control	Read/write	Byte	00h	<a href="#">Section 1.4.5</a>
09h		Reserved - reads zero	Read	Byte	00h	
0Ah	UCAxSTAT	USCI_Ax Status	Read/write	Byte	00h	<a href="#">Section 1.4.6</a>
0Bh		Reserved - reads zero	Read	Byte	00h	
0Ch	UCAxRXBUF	USCI_Ax Receive Buffer	Read/write	Byte	00h	<a href="#">Section 1.4.7</a>
0Dh		Reserved - reads zero	Read	Byte	00h	
0Eh	UCAxTXBUF	USCI_Ax Transmit Buffer	Read/write	Byte	00h	<a href="#">Section 1.4.8</a>
0Fh		Reserved - reads zero	Read	Byte	00h	
10h	UCAxABCTL	USCI_Ax Auto Baud Rate Control	Read/write	Byte	00h	<a href="#">Section 1.4.11</a>
11h		Reserved - reads zero	Read	Byte	00h	
12h	UCAxIRCTL	USCI_Ax IrDA Control	Read/write	Word	0000h	
12h	UCAxIRTCTL	USCI_Ax IrDA Transmit Control	Read/write	Byte	00h	<a href="#">Section 1.4.9</a>
13h	UCAxIRRCTL	USCI_Ax IrDA Receive Control	Read/write	Byte	00h	<a href="#">Section 1.4.10</a>
1Ch	UCAxICTL	USCI_Ax Interrupt Control	Read/write	Word	0000h	
1Ch	UCAxIE	USCI_Ax Interrupt Enable	Read/write	Byte	00h	<a href="#">Section 1.4.12</a>
1Dh	UCAxIFG	USCI_Ax Interrupt Flag	Read/write	Byte	00h	<a href="#">Section 1.4.13</a>
1Eh	UCAxIV	USCI_Ax Interrupt Vector	Read	Word	0000h	<a href="#">Section 1.4.14</a>

### 1.4.1 UCxCTL0 Register

USCI\_Ax Control Register 0

**Figure 1-12. UCxCTL0 Register**

7	6	5	4	3	2	1	0
UCPEN	UCPAR	UCMSB	UC7BIT	UCSPB	UCMODEx		UCSYNC
rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0

Modify only when UCSWRST = 1.

**Table 1-7. UCxCTL0 Register Description**

Bit	Field	Type	Reset	Description
7	UCPEN	RW	0h	Parity enable 0b = Parity disabled 1b = Parity enabled. Parity bit is generated (UCxTXD) and expected (UCxRXD). In address-bit multiprocessor mode, the address bit is included in the parity calculation.
6	UCPAR	RW	0h	Parity select. UCPAR is not used when parity is disabled. 0b = Odd parity 1b = Even parity
5	UCMSB	RW	0h	MSB first select. Controls the direction of the receive and transmit shift register. 0b = LSB first 1b = MSB first
4	UC7BIT	RW	0h	Character length. Selects 7-bit or 8-bit character length. 0b = 8-bit data 1b = 7-bit data
3	UCSPB	RW	0h	Stop bit select. Number of stop bits. 0b = One stop bit 1b = Two stop bits
2-1	UCMODEx	RW	0h	USCI mode. The UCMODEx bits select the asynchronous mode when UCSYNC = 0. 00b = UART mode 01b = Idle-line multiprocessor mode 10b = Address-bit multiprocessor mode 11b = UART mode with automatic baud-rate detection
0	UCSYNC	RW	0h	Synchronous mode enable 0b = Asynchronous mode 1b = Synchronous mode

## 1.4.2 UCAXCTL1 Register

USCI\_Ax Control Register 1

**Figure 1-13. UCAXCTL1 Register**

7	6	5	4	3	2	1	0
UCSSELx	UCRXEIE	UCBRKIE	UCDORM	UCTXADDR	UCTXBRK	UCSWRST	
rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-1

Modify only when UCSWRST = 1.

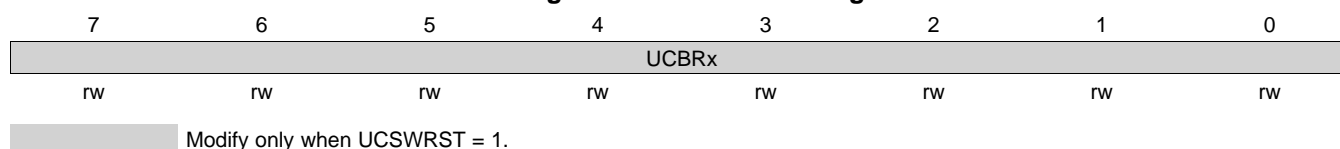
**Table 1-8. UCAXCTL1 Register Description**

Bit	Field	Type	Reset	Description
7-6	UCSSELx	RW	0h	USCI clock source select. These bits select the BRCLK source clock. 00b = UCAXCLK (external USCI clock) 01b = ACLK 10b = SMCLK 11b = SMCLK
5	UCRXEIE	RW	0h	Receive erroneous-character interrupt enable 0b = Erroneous characters rejected and UCRXIFG is not set. 1b = Erroneous characters received set UCRXIFG.
4	UCBRKIE	RW	0h	Receive break character interrupt enable 0b = Received break characters do not set UCRXIFG. 1b = Received break characters set UCRXIFG.
3	UCDORM	RW	0h	Dormant. Puts USCI into sleep mode. 0b = Not dormant. All received characters set UCRXIFG. 1b = Dormant. Only characters that are preceded by an idle-line or with address bit set UCRXIFG. In UART mode with automatic baud-rate detection, only the combination of a break and synch field sets UCRXIFG.
2	UCTXADDR	RW	0h	Transmit address. Next frame to be transmitted is marked as address, depending on the selected multiprocessor mode. 0b = Next frame transmitted is data. 1b = Next frame transmitted is an address.
1	UCTXBRK	RW	0h	Transmit break. Transmits a break with the next write to the transmit buffer. In UART mode with automatic baud-rate detection, 055h must be written into UCAXTXBUF to generate the required break/synch fields. Otherwise, 0h must be written into the transmit buffer. 0b = Next frame transmitted is not a break. 1b = Next frame transmitted is a break or a break/synch.
0	UCSWRST	RW	1h	Software reset enable 0b = Disabled. USCI reset released for operation. 1b = Enabled. USCI logic held in reset state.

### 1.4.3 UCxBR0 Register

USCI\_Ax Baud Rate Control Register 0

**Figure 1-14. UCxBR0 Register**



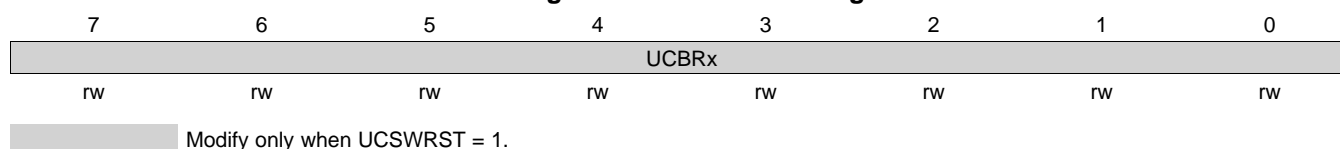
**Table 1-9. UCxBR0 Register Description**

Bit	Field	Type	Reset	Description
7-0	UCBRx	RW	undefined	Low byte of clock prescaler setting of the baud-rate generator. The 16-bit value of (UCxBR0 + UCxBR1 × 256) forms the prescaler value UCBRx.

### 1.4.4 UCxBR1 Register

USCI\_Ax Baud Rate Control Register 1

**Figure 1-15. UCxBR1 Register**



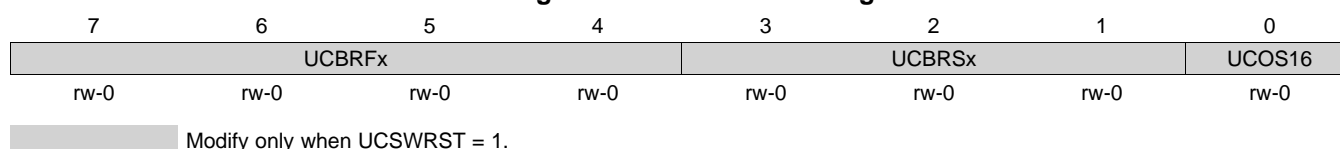
**Table 1-10. UCxBR1 Register Description**

Bit	Field	Type	Reset	Description
7-0	UCBRx	RW	undefined	High byte of clock prescaler setting of the baud-rate generator. The 16-bit value of (UCxBR0 + UCxBR1 × 256) forms the prescaler value UCBRx.

### 1.4.5 UCxMCTL Register

USCI\_Ax Modulation Control Register

**Figure 1-16. UCxMCTL Register**



**Table 1-11. UCxMCTL Register Description**

Bit	Field	Type	Reset	Description
7-4	UCBRFx	RW	0h	First modulation stage select. These bits determine the modulation pattern for BITCLK16 when UCOS16 = 1. Ignored with UCOS16 = 0. <a href="#">Table 1-2</a> shows the modulation pattern.
3-1	UCBRSx	RW	0h	Second modulation stage select. These bits determine the modulation pattern for BITCLK. <a href="#">Table 1-2</a> shows the modulation pattern.
0	UCOS16	RW	0h	Oversampling mode enabled 0b = Disabled 1b = Enabled



### 1.4.6 UCxSTAT Register

USCI\_Ax Status Register

**Figure 1-17. UCxSTAT Register**

7	6	5	4	3	2	1	0
UCLISTEN	UCFE	UCOE	UCPE	UCBRK	UCRXERR	UCADDR/ UCIDLE	UCBUSY
rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	r-0

Modify only when UCSWRST = 1.

**Table 1-12. UCxSTAT Register Description**

Bit	Field	Type	Reset	Description
7	UCLISTEN	RW	0h	Listen enable. The UCLISTEN bit selects loopback mode. 0b = Disabled 1b = Enabled. UCxTXD is internally fed back to the receiver.
6	UCFE	RW	0h	Framing error flag. UCFE is cleared when UCxRXBUF is read. 0b = No error 1b = Character received with low stop bit
5	UCOE	RW	0h	Overrun error flag. This bit is set when a character is transferred into UCxRXBUF before the previous character was read. UCOE is cleared automatically when UCxRXBUF is read, and must not be cleared by software. Otherwise, it does not function correctly. 0b = No error 1b = Overrun error occurred
4	UCPE	RW	0h	Parity error flag. When UCPEN = 0, UCPE is read as 0. UCPE is cleared when UCxRXBUF is read. 0b = No error 1b = Character received with parity error
3	UCBRK	RW	0h	Break detect flag. UCBRK is cleared when UCxRXBUF is read. 0b = No break condition 1b = Break condition occurred
2	UCRXERR	RW	0h	Receive error flag. This bit indicates a character was received with error(s). When UCRXERR = 1, on or more error flags, UCFE, UCPE, or UCOE is also set. UCRXERR is cleared when UCxRXBUF is read. 0b = No receive errors detected 1b = Receive error detected
1	UCADDR/UCIDLE	RW	0h	UCADDR: Address received in address-bit multiprocessor mode. UCADDR is cleared when UCxRXBUF is read. 0b = Received character is data. 1b = Received character is an address. UCIDLE: Idle line detected in idle-line multiprocessor mode. UCIDLE is cleared when UCxRXBUF is read. 0b = No idle line detected 1b = Idle line detected
0	UCBUSY	R	0h	USCI busy. This bit indicates if a transmit or receive operation is in progress. 0b = USCI inactive 1b = USCI transmitting or receiving

### 1.4.7 UCAXRXBUF Register

USCI\_Ax Receive Buffer Register

**Figure 1-18. UCAXRXBUF Register**



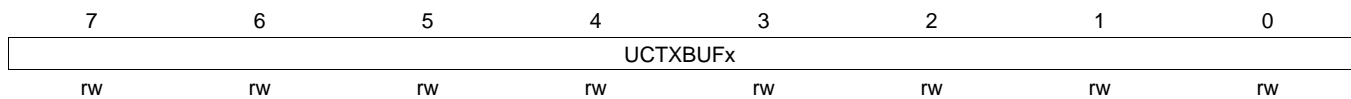
**Table 1-13. UCAXRXBUF Register Description**

Bit	Field	Type	Reset	Description
7-0	UCRXBUFx	R	undefined	The receive-data buffer is user accessible and contains the last received character from the receive shift register. Reading UCAXRXBUF resets the receive-error bits, the UCADDR or UCIDLE bit, and UCRXIFG. In 7-bit data mode, UCAXRXBUF is LSB justified and the MSB is always reset.

### 1.4.8 UCAXTXBUF Register

USCI\_Ax Transmit Buffer Register

**Figure 1-19. UCAXTXBUF Register**

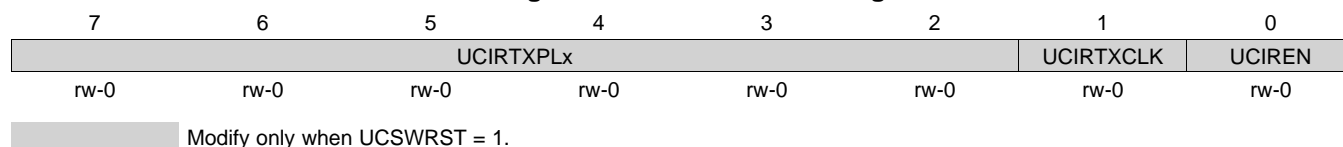


**Table 1-14. UCAXTXBUF Register Description**

Bit	Field	Type	Reset	Description
7-0	UCTXBUFx	RW	undefined	The transmit data buffer is user accessible and holds the data waiting to be moved into the transmit shift register and transmitted on UCAXTXD. Writing to the transmit data buffer clears UCTXIFG. The MSB of UCAXTXBUF is not used for 7-bit data and is reset.

### 1.4.9 UCxIRTCTL Register

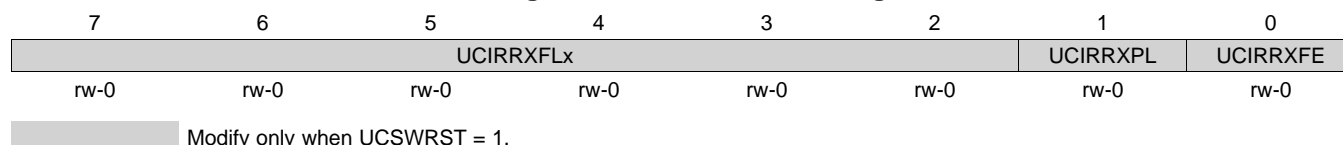
USCI\_Ax IrDA Transmit Control Register

**Figure 1-20. UCxIRTCTL Register**

**Table 1-15. UCxIRTCTL Register Description**

Bit	Field	Type	Reset	Description
7-2	UCIRTXPLx	RW	0h	Transmit pulse length. Pulse length $t(\text{PULSE}) = (\text{UCIRTXPLx} + 1) / [2 \times f(\text{IRTXCLK})]$
1	UCIRTXCLK	RW	0h	IrDA transmit pulse clock select 0b = BRCLK 1b = BITCLK16 when UCOS16 = 1. Otherwise, BRCLK.
0	UCIREN	RW	0h	IrDA encoder and decoder enable 0b = IrDA encoder and decoder disabled 1b = IrDA encoder and decoder enabled

### 1.4.10 UCxIRRCTL Register

USCI\_Ax IrDA Receive Control Register

**Figure 1-21. UCxIRRCTL Register**

**Table 1-16. UCxIRRCTL Register Description**

Bit	Field	Type	Reset	Description
7-2	UCIRRFLx	RW	0h	Receive filter length. The minimum pulse length for receive is given by: $t(\text{MIN}) = (\text{UCIRRFLx} + 4) / (2 \times f(\text{BRCLK}))$
1	UCIRRPL	RW	0h	IrDA receive input UCxRXD polarity 0b = IrDA transceiver delivers a high pulse when a light pulse is seen. 1b = IrDA transceiver delivers a low pulse when a light pulse is seen.
0	UCIRRXFE	RW	0h	IrDA receive filter enabled 0b = Receive filter disabled 1b = Receive filter enabled

### 1.4.11 UCxABCTL Register

USCI\_Ax Auto Baud Rate Control Register

**Figure 1-22. UCxABCTL Register**

7	6	5	4	3	2	1	0
Reserved		UCDELIMx		UCSTOE	UCBTOE	Reserved	UCABDEN
r-0	r-0	rw-0	rw-0	rw-0	rw-0	r-0	rw-0

Modify only when UCSWRST = 1.

**Table 1-17. UCxABCTL Register Description**

Bit	Field	Type	Reset	Description
7-6	Reserved	R	0h	Reserved. Always reads as 0.
5-4	UCDELIMx	RW	0h	Break and synch delimiter length 00b = 1 bit time 01b = 2 bit times 10b = 3 bit times 11b = 4 bit times
3	UCSTOE	RW	0h	Synch field time out error 0b = No error 1b = Length of synch field exceeded measurable time.
2	UCBTOE	RW	0h	Break time out error 0b = No error 1b = Length of break field exceeded 22 bit times.
1	Reserved	R	0h	Reserved. Always reads as 0.
0	UCABDEN	RW	0h	Automatic baud-rate detect enable 0b = Baud-rate detection disabled. Length of break and synch field is not measured. 1b = Baud-rate detection enabled. Length of break and synch field is measured and baud-rate settings are changed accordingly.

### 1.4.12 UCAXIE Register

USCI\_Ax Interrupt Enable Register

**Figure 1-23. UCAXIE Register**

7	6	5	4	3	2	1	0
Reserved						UCTXIE	UCRXIE
r-0	r-0	r-0	r-0	r-0	r-0	rw-0	rw-0

**Table 1-18. UCAXIE Register Description**

Bit	Field	Type	Reset	Description
7-2	Reserved	R	0h	Reserved. Always reads as 0.
1	UCTXIE	RW	0h	Transmit interrupt enable 0b = Interrupt disabled 1b = Interrupt enabled
0	UCRXIE	RW	0h	Receive interrupt enable 0b = Interrupt disabled 1b = Interrupt enabled

### 1.4.13 UCAXIFG Register

USCI\_Ax Interrupt Flag Register

**Figure 1-24. UCAXIFG Register**

7	6	5	4	3	2	1	0
Reserved						UCTXIFG	UCRXIFG
r-0	r-0	r-0	r-0	r-0	r-0	rw-1	rw-0

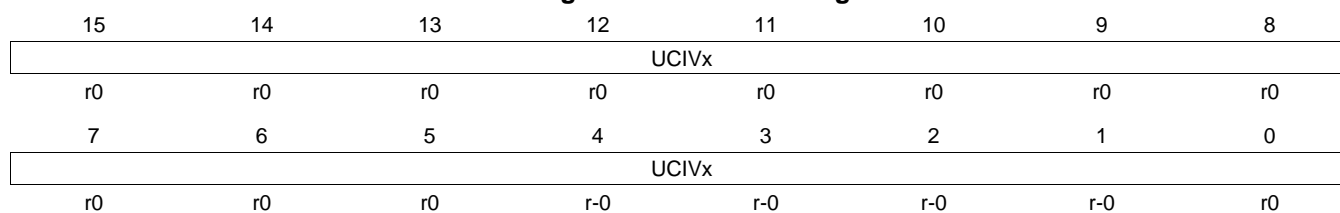
**Table 1-19. UCAXIFG Register Description**

Bit	Field	Type	Reset	Description
7-2	Reserved	R	0h	Reserved. Always reads as 0.
1	UCTXIFG	RW	1h	Transmit interrupt flag. UCTXIFG is set when UCAXTXBUF empty. 0b = No interrupt pending 1b = Interrupt pending
0	UCRXIFG	RW	0h	Receive interrupt flag. UCRXIFG is set when UCAXRXBUF has received a complete character. 0b = No interrupt pending 1b = Interrupt pending

### 1.4.14 UCAXIV Register

USCI\_Ax Interrupt Vector Register

**Figure 1-25. UCAXIV Register**



**Table 1-20. UCAXIV Register Description**

Bit	Field	Type	Reset	Description
15-0	UCIVx	R	0h	USCI interrupt vector value 00h = No interrupt pending 02h = Interrupt Source: Data received; Interrupt Flag: UCRXIFG; Interrupt Priority: Highest 04h = Interrupt Source: Transmit buffer empty; Interrupt Flag: UCTXIFG; Interrupt Priority: Lowest

## IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as "components") are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI's terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

TI assumes no liability for applications assistance or the design of Buyers' products. Buyers are responsible for their products and applications using TI components. To minimize the risks associated with Buyers' products and applications, Buyers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of significant portions of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI components or services with statements different from or beyond the parameters stated by TI for that component or service voids all express and any implied warranties for the associated TI component or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI's goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or "enhanced plastic" are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have **not** been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

### Products

Audio	<a href="http://www.ti.com/audio">www.ti.com/audio</a>
Amplifiers	<a href="http://amplifier.ti.com">amplifier.ti.com</a>
Data Converters	<a href="http://dataconverter.ti.com">dataconverter.ti.com</a>
DLP® Products	<a href="http://www.dlp.com">www.dlp.com</a>
DSP	<a href="http://dsp.ti.com">dsp.ti.com</a>
Clocks and Timers	<a href="http://www.ti.com/clocks">www.ti.com/clocks</a>
Interface	<a href="http://interface.ti.com">interface.ti.com</a>
Logic	<a href="http://logic.ti.com">logic.ti.com</a>
Power Mgmt	<a href="http://power.ti.com">power.ti.com</a>
Microcontrollers	<a href="http://microcontroller.ti.com">microcontroller.ti.com</a>
RFID	<a href="http://www.ti-rfid.com">www.ti-rfid.com</a>
OMAP Applications Processors	<a href="http://www.ti.com/omap">www.ti.com/omap</a>
Wireless Connectivity	<a href="http://www.ti.com/wirelessconnectivity">www.ti.com/wirelessconnectivity</a>

### Applications

Automotive and Transportation	<a href="http://www.ti.com/automotive">www.ti.com/automotive</a>
Communications and Telecom	<a href="http://www.ti.com/communications">www.ti.com/communications</a>
Computers and Peripherals	<a href="http://www.ti.com/computers">www.ti.com/computers</a>
Consumer Electronics	<a href="http://www.ti.com/consumer-apps">www.ti.com/consumer-apps</a>
Energy and Lighting	<a href="http://www.ti.com/energy">www.ti.com/energy</a>
Industrial	<a href="http://www.ti.com/industrial">www.ti.com/industrial</a>
Medical	<a href="http://www.ti.com/medical">www.ti.com/medical</a>
Security	<a href="http://www.ti.com/security">www.ti.com/security</a>
Space, Avionics and Defense	<a href="http://www.ti.com/space-avionics-defense">www.ti.com/space-avionics-defense</a>
Video and Imaging	<a href="http://www.ti.com/video">www.ti.com/video</a>

### TI E2E Community

[e2e.ti.com](http://e2e.ti.com)