

# The USART

# Introduction

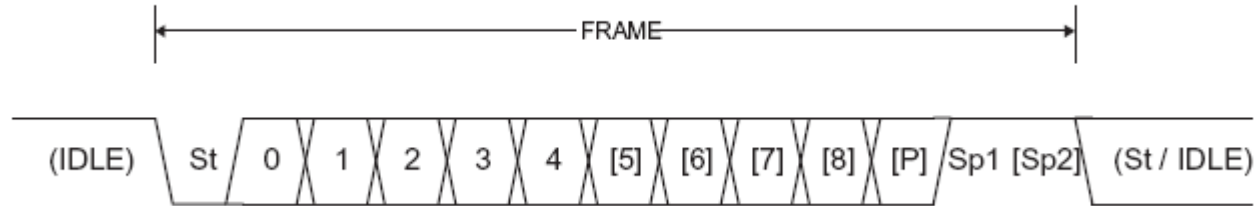
- The Universal Synchronous Asynchronous Receiver Transmitter is a highly flexible communication device that transmits and receives data serially
- Commonly used with microcontrollers
- Data is loaded from the databus onto the transmit buffer and serially shifted out of the shift register after loading the shift register with the data in the transmit buffer
- Data is read serially into the shift register and then onto the receive buffer, from which it is read by the controller through the data bus

# Frame Format

- A serial frame is defined to be one character of data bits with synchronization bits (start and stop bits), and optionally a parity bit for error checking.
- The USART accepts all the combinations of the following as valid frame formats:
  - 1 start bit
  - 5, 6, 7, 8, or 9 data bits
  - no, even or odd parity bit
  - 1 or 2 stop bits

# Frame Format

Figure 72. Frame Formats



A frame starts with the start bit followed by the least significant data bit.

Then the next data bits, up to a total of nine, are succeeding, ending with the most significant bit.

If enabled, the parity bit is inserted after the data bits, before the stop bits.

When a complete frame is transmitted, it can be directly followed by a new frame, or the communication line can be set to an idle (high) state.

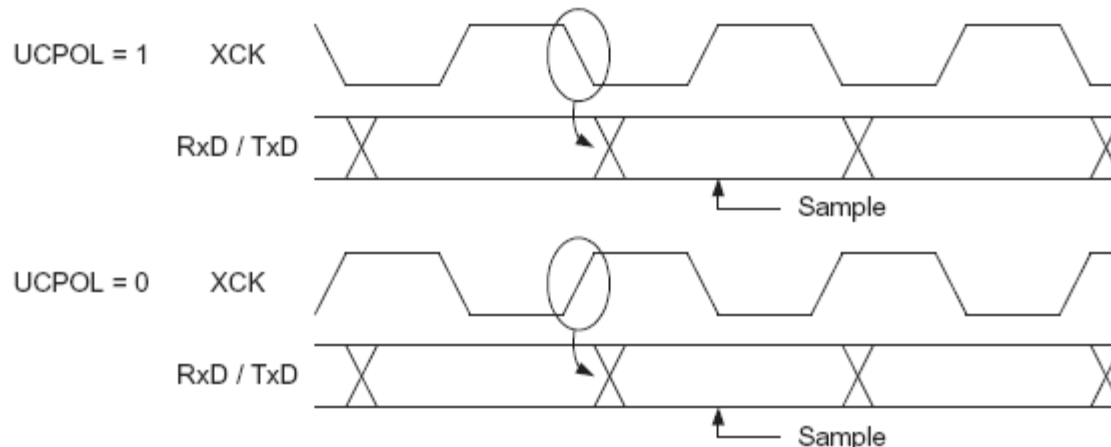
# Asynchronous and Synchronous Modes

- Asynchronous transmission allows data to be transmitted without the sender having to send a clock signal to the receiver. Instead, the sender and receiver must agree on timing parameters in advance and special bits are added to each word which are used to synchronize the sending and receiving units.
- When a word is given to the USART for Asynchronous transmissions, a bit called the "Start Bit" is added to the beginning of each word that is to be transmitted. The Start Bit is used to alert the receiver that a word of data is about to be sent, and to force the clock in the receiver into synchronization with the clock in the transmitter. These two clocks must be accurate enough to not have the frequency drift by more than 10% during the transmission of the remaining bits in the word.

# Asynchronous and Synchronous Modes

- In *synchronous* transmission, the clock data is recovered separately from the data stream and no start/stop bits are used. This improves the efficiency of transmission on suitable channels since more of the bits sent are usable data and not character framing.

**Figure 71.** Synchronous Mode XCK Timing.



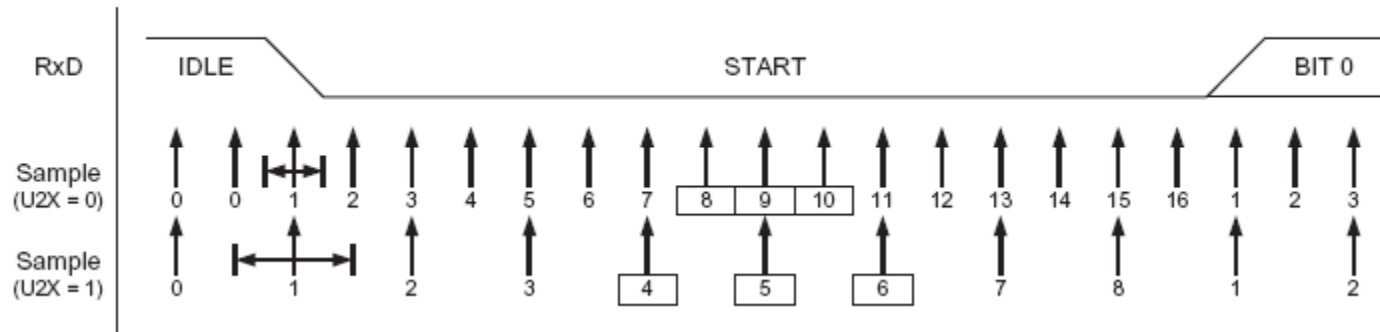
# Asynchronous and Synchronous Modes

- The UCPOL bit UCRSC selects which XCK clock edge is used for data sampling and which is used for data change.
- When UCPOL is zero the data will be changed at rising XCK edge and sampled at falling XCK edge. If UCPOL is set, the data will be changed at falling XCK edge and sampled at rising XCK edge.

# Asynchronous Data Recovery

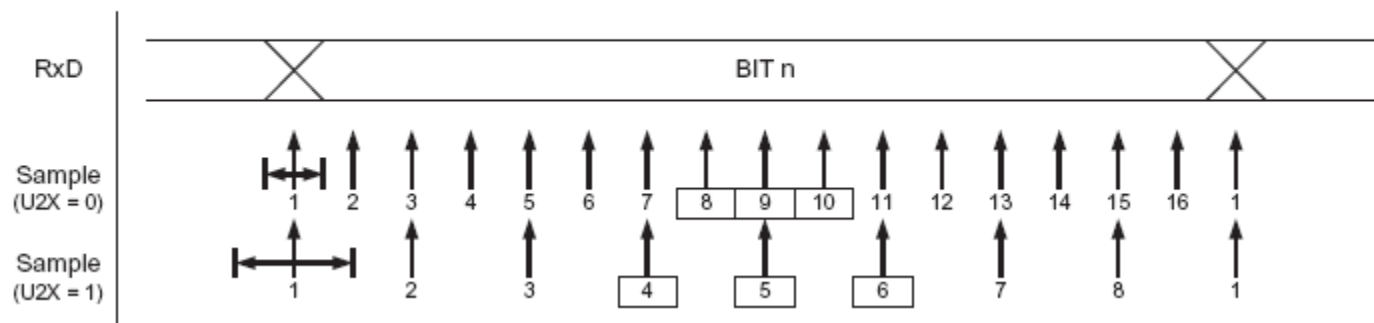
- The data is sampled 8 (double speed mode) or 16 times the baudrate (normal mode) . A majority decision is taken based on the values of the 8,9 and 10 samples (normal mode) or 4,5 and 6 samples (double speed mode)

**Figure 73. Start Bit Sampling**

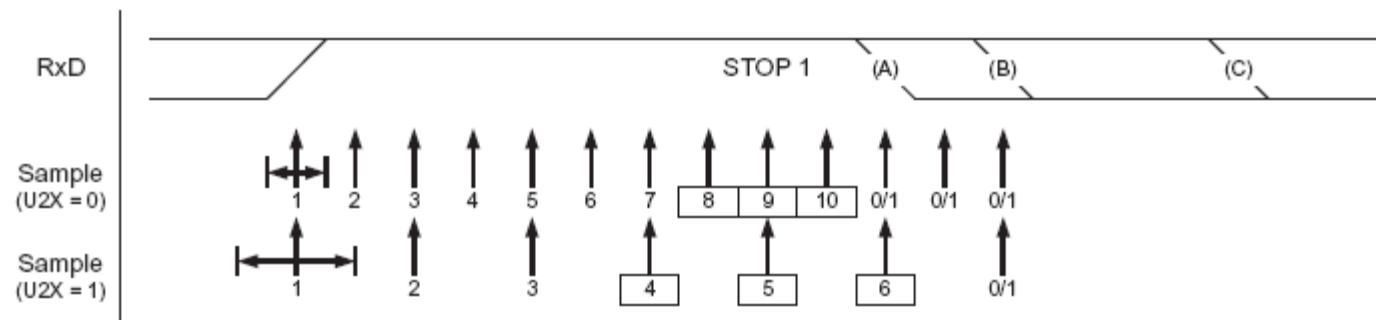




**Figure 74. Sampling of Data and Parity Bit**



**Figure 75. Stop Bit Sampling and Next Start Bit Sampling**



# Computing BAUD Rates

**Table 60.** Equations for Calculating Baud Rate Register Setting

Operating Mode	Equation for Calculating Baud Rate <sup>(1)</sup>	Equation for Calculating UBRR Value
Asynchronous Normal Mode (U2X = 0)	$BAUD = \frac{f_{osc}}{16(UBRR + 1)}$	$UBRR = \frac{f_{osc}}{16BAUD} - 1$
Asynchronous Double Speed Mode (U2X = 1)	$BAUD = \frac{f_{osc}}{8(UBRR + 1)}$	$UBRR = \frac{f_{osc}}{8BAUD} - 1$
Synchronous Master Mode	$BAUD = \frac{f_{osc}}{2(UBRR + 1)}$	$UBRR = \frac{f_{osc}}{2BAUD} - 1$

Note: 1. The baud rate is defined to be the transfer rate in bit per second (bps).

# USART Initialization

- The USART has to be initialized before any communication can take place. The initialization process normally consists of setting the baud rate, setting frame format and enabling the Transmitter or the Receiver depending on the usage. For interrupt driven USART operation, the Global Interrupt Flag should be cleared (and interrupts globally disabled) when doing the initialization.

# USART Initialization

```
#define FOSC 1843200// Clock Speed
#define BAUD 9600
#define MYUBRR FOSC/16/BAUD-1
void main( void )
{
    ...
    USART_Init ( MYUBRR );
    ...
}
void USART_Init( unsigned int ubrr)
{
    /* Set baud rate */
    UBRRH = (unsigned char) (ubrr>>8);
    UBRRL = (unsigned char)ubrr;
    /* Enable receiver and transmitter */
    UCSRB = (1<<RXEN) | (1<<TXEN);
    /* Set frame format: 8data, 2stop bit */
    UCSRC = (1<<URSEL) | (1<<USBS) | (3<<UCSZ0);
}
```

# USART Transmission

- A data transmission is initiated by loading the transmit buffer with the data to be transmitted. The CPU can load the transmit buffer by writing to the UDR I/O location. The buffered data in the transmit buffer will be moved to the Shift Register when the Shift Register is ready to send a new frame.
- The Shift Register is loaded with new data if it is in idle state (no ongoing transmission) or immediately after the last stop bit of the previous frame is transmitted.
- When the Shift Register is loaded with new data, it will transfer one complete frame at the rate given by the Baud Register, U2X bit or by XCK depending on mode of operation.

# USART Transmission

```
void USART_Transmit( unsigned char data )
{
    /* Wait for empty transmit buffer */
    while ( !( UCSRA & (1<<UDRE)) )
        ;
    /* Put data into buffer, sends the data */
    UDR = data;
}
```

Transmission of data less than 8 bits

# Transmitting 9 bits of data

## C Code Example<sup>(1)</sup>

```
void USART_Transmit( unsigned int data )
{
    /* Wait for empty transmit buffer */
    while ( !( UCSRA & (1<<UDRE)) )
        ;
    /* Copy 9th bit to TXB8 */
    UCSRB &= ~(1<<TXB8);
    if ( data & 0x0100 )
        UCSRB |= (1<<TXB8);
    /* Put data into buffer, sends the data */
    UDR = data;
}
```

- The 9<sup>th</sup> bit or MSB should be written to TXB8 bit ins UCSRB register before writing the lower byte into UDR.
- Note that the TXB8 is first cleared of earlier values before writing to it

# USART Receiver

- The USART Receiver is enabled by writing the Receive Enable (RXEN) bit in the UCSRB Register to one. When the receiver is enabled, the normal pin operation of the RxD pin is overridden by the USART and given the function as the receiver's serial input
- The receiver starts data reception when it detects a valid start bit. Each bit that follows the start bit will be sampled at the baud rate or XCK clock, and shifted into the receive Shift Register until the first stop bit of a frame is received. A second stop bit will be ignored by the receiver.
- When the first stop bit is received, i.e., a complete serial frame is present in the receive Shift Register, the contents of the Shift Register will be moved into the receive buffer.
- The receive buffer can then be read by reading the UDR I/O location.



# USART Receiver

C Code Example

```
unsigned char USART_Receive( void )
{
    /* Wait for data to be received */
    while ( !(UCSRA & (1<<RXC)) )
        ;
    /* Get and return received data from buffer */
    return UDR;
}
```

The function simply waits for data to be present in the receive buffer by checking the RXC Flag, before reading the buffer and returning the value.

# Receiving 9 bits of data

## C Code Example<sup>(1)</sup>

```
unsigned int USART_Receive( void )
{
    unsigned char status, resh, resl;
    /* Wait for data to be received */
    while ( !(UCSRA & (1<<RXC)) )
        ;
    /* Get status and 9th bit, then data */
    /* from buffer */
    status = UCSRA;
    resh = UCSRB;
    resl = UDR;
    /* If error, return -1 */
    if ( status & (1<<FE) | (1<<DOR) | (1<<PE) )
        return -1;
    /* Filter the 9th bit, then return */
    resh = (resh >> 1) & 0x01;
    return ((resh << 8) | resl);
}
```

Note: 1 See "About Code Examples" on page 7

- The key is to read the status and 9<sup>th</sup> bit first and then the lower byte from UDR. The 9<sup>th</sup> bit is bit 1 of the UCSRB register, the bit RXB8

# USART Interrupts

- The USART transmitter has two flags that indicate its state: USART Data Register Empty (UDRE) and Transmit Complete (TXC). Both flags can be used for generating interrupts.
- The Data Register Empty (UDRE) Flag indicates whether the transmit buffer is ready to receive new data. This bit is set when the transmit buffer is empty, and cleared when the transmit buffer contains data to be transmitted that has not yet been moved into the Shift Register.
- When the Data Register empty Interrupt Enable (UDRIE) bit in UCSRB is written to one, the USART Data Register Empty Interrupt will be executed as long as UDRE is set (provided that global interrupts are enabled). UDRE is cleared by writing UDR.
- When interrupt-driven data transmission is used, the Data Register Empty Interrupt routine must either write new data to UDR in order to clear UDRE or disable the Data Register empty Interrupt, otherwise a new interrupt will occur once the interrupt routine terminates.

# USART Interrupts

- The Transmit Complete (TXC) Flag bit is set one when the entire frame in the transmit Shift Register has been shifted out and there are no new data currently present in the transmit buffer.
- The TXC Flag bit is automatically cleared when a transmit complete interrupt is executed, or it can be cleared by writing a one to its bit location.
- When the Transmit Complete Interrupt Enable (TXCIE) bit in UCSRB is set, the USART Transmit Complete Interrupt will be executed when the TXC Flag becomes set (provided that global interrupts are enabled).
- When the transmit complete interrupt is used, the interrupt handling routine does not have to clear the TXC Flag, this is done automatically when the interrupt is executed.

# USART Interrupts

- The USART Receiver has one flag that indicates the receiver state.
- The Receive Complete (RXC) Flag indicates if there are unread data present in the receive buffer. This flag is one when unread data exist in the receive buffer, and zero when the receive buffer is empty (i.e., does not contain any unread data).
- When the Receive Complete Interrupt Enable (RXCIE) in UCSRB is set, the USART Receive Complete Interrupt will be executed as long as the RXC Flag is set (provided that global interrupts are enabled).
- When interrupt-driven data reception is used, the receive complete routine must read the received data from UDR in order to clear the RXC Flag, otherwise a new interrupt will occur once the interrupt routine terminates.

# USART Registers

USART I/O Data  
Register – UDR

Bit	7	6	5	4	3	2	1	0	
	RXB[7:0]								UDR (Read)
	TXB[7:0]								UDR (Write)
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The USART Transmit Data Buffer Register and USART Receive Data Buffer Registers share the same I/O address referred to as USART Data Register or UDR. The Transmit Data Buffer Register (TXB) will be the destination for data written to the UDR Register location. Reading the UDR Register location will return the contents of the Receive Data Buffer Register (RXB).

The transmit buffer can only be written when the UDRE Flag in the UCSRA Register is set. Data written to UDR when the UDRE Flag is not set, will be ignored by the USART Transmitter. When data is written to the transmit buffer, and the Transmitter is enabled, the Transmitter will load the data into the transmit Shift Register when the Shift Register is empty. Then the data will be serially transmitted on the TxD pin.

# • USART Control and Status Reg A

Bit	7	6	5	4	3	2	1	0	
	<b>RXC</b>	<b>TXC</b>	<b>UDRE</b>	<b>FE</b>	<b>DOR</b>	<b>PE</b>	<b>U2X</b>	<b>MPCM</b>	UCSRA
Read/Write	R	R/W	R	R	R	R	R/W	R/W	
Initial Value	0	0	1	0	0	0	0	0	

## • Bit 7 – RXC: USART Receive Complete

This flag bit is set when there are unread data in the receive buffer and cleared when the receive buffer is empty (i.e., does not contain any unread data). If the receiver is disabled, the receive buffer will be flushed and consequently the RXC bit will become zero. The RXC Flag can be used to generate a Receive Complete interrupt (see description of the RXCIE bit).

## • Bit 6 – TXC: USART Transmit Complete

This flag bit is set when the entire frame in the transmit Shift Register has been shifted out and there are no new data currently present in the transmit buffer (UDR). The TXC Flag bit is automatically cleared when a transmit complete interrupt is executed, or it can be cleared by writing a one to its bit location. The TXC Flag can generate a Transmit Complete interrupt (see description of the TXCIE bit).

## • Bit 5 – UDRE: USART Data Register Empty

The UDRE Flag indicates if the transmit buffer (UDR) is ready to receive new data. If UDRE is one, the buffer is empty, and therefore ready to be written. The UDRE Flag can generate a Data Register empty Interrupt (see description of the UDRIE bit).

UDRE is set after a reset to indicate that the transmitter is ready.

Bit	7	6	5	4	3	2	1	0	
	<b>RXCIE</b>	<b>TXCIE</b>	<b>UDRIE</b>	<b>RXEN</b>	<b>TXEN</b>	<b>UCSZ2</b>	<b>RXB8</b>	<b>TXB8</b>	<b>UCSRB</b>
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – RXCIE: RX Complete Interrupt Enable**

Writing this bit to one enables interrupt on the RXC Flag. A USART Receive Complete Interrupt will be generated only if the RXCIE bit is written to one, the Global Interrupt Flag in SREG is written to one and the RXC bit in UCSRA is set.

- **Bit 6 – TXCIE: TX Complete Interrupt Enable**

Writing this bit to one enables interrupt on the TXC Flag. A USART Transmit Complete Interrupt will be generated only if the TXCIE bit is written to one, the Global Interrupt Flag in SREG is written to one and the TXC bit in UCSRA is set.

- **Bit 5 – UDRIE: USART Data Register Empty Interrupt Enable**

Writing this bit to one enables interrupt on the UDRE Flag. A Data Register Empty Interrupt will be generated only if the UDRIE bit is written to one, the Global Interrupt Flag in SREG is written to one and the UDRE bit in UCSRA is set.

- **Bit 4 – RXEN: Receiver Enable**

Writing this bit to one enables the USART Receiver. The Receiver will override normal port operation for the RxD pin when enabled. Disabling the Receiver will flush the receive buffer invalidating the FE, DOR, and PE Flags.

- **Bit 3 – TXEN: Transmitter Enable**

Writing this bit to one enables the USART Transmitter. The Transmitter will override normal port operation for the TxD pin when enabled. The disabling of the Transmitter (writing TXEN to zero) will not become effective until ongoing and pending transmissions are completed, i.e., when the transmit Shift Register and transmit Buffer Register do not contain data to be transmitted. When disabled, the transmitter will no longer override the TxD port.

- **Bit 2 – UCSZ2: Character Size**

The UCSZ2 bits combined with the UCSZ1:0 bit in UCSRC sets the number of data bits (Char-



# USCRC

Bit	7	6	5	4	3	2	1	0	
	<b>URSEL</b>	<b>UMSEL</b>	<b>UPM1</b>	<b>UPM0</b>	<b>USBS</b>	<b>UCSZ1</b>	<b>UCSZ0</b>	<b>UCPOL</b>	<b>USRC</b>
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	1	0	0	0	0	1	1	0	

- **Bit 7 - URSEL: Register Select:** This bit selects between accessing the USRC or the UBRRH Register. It is read as one when reading USRC. The URSEL must be one when writing the USRC

**Table 65. USBS Bit Settings**

USBS	Stop Bit(s)
0	1-bit
1	2-bit

# USCRC

**Table 66. UCSZ Bits Settings**

<b>UCSZ2</b>	<b>UCSZ1</b>	<b>UCSZ0</b>	<b>Character Size</b>
0	0	0	5-bit
0	0	1	6-bit
0	1	0	7-bit
0	1	1	8-bit
1	0	0	Reserved
1	0	1	Reserved
1	1	0	Reserved
1	1	1	9-bit

# Accessing USCRC and UBRRH

- The USCRC and UBRRH share the same I/O location
- the USART Register Select (URSEL) bit, controls which one of the two registers that will be written. If URSEL is zero during a write operation, the UBRRH value will be updated. If URSEL is one, the UCSRC setting will be updated.

```
...  
/* Set UBRRH to 2 */  
UBRRH = 0x02;  
  
...  
/* Set the USBS and the UCSZ1 bit to one, and */  
/* the remaining bits to zero. */  
UCSRC = (1<<URSEL) | (1<<USBS) | (1<<UCSZ1);  
  
...
```

- In the above code, initial URSEL bit is low and hence the UBRRH value gets updated. Next while writing to UCSRC

# The USART

## Introduction

- The Universal Synchronous Asynchronous Receiver Transmitter is a highly flexible communication device that transmits and receives data serially
- Commonly used with microcontrollers
- Data is loaded from the databus onto the transmit buffer and serially shifted out of the shift register after loading the shift register with the data in the transmit buffer
- Data is read serially into the shift register and then onto the receive buffer, from which it is read by the controller through the data bus

2

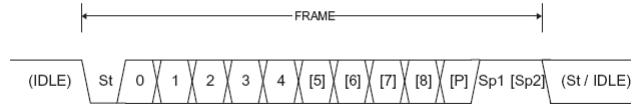
af

# Frame Format

- A serial frame is defined to be one character of data bits with synchronization bits (start and stop bits), and optionally a parity bit for error checking.
- The USART accepts all the combinations of the following as valid frame formats:
  - 1 start bit
  - 5, 6, 7, 8, or 9 data bits
  - no, even or odd parity bit
  - 1 or 2 stop bits

# Frame Format

Figure 72. Frame Formats



A frame starts with the start bit followed by the least significant data bit.

Then the next data bits, up to a total of nine, are succeeding, ending with the most significant bit.

If enabled, the parity bit is inserted after the data bits, before the stop bits.

When a complete frame is transmitted, it can be directly followed by a new frame, or the communication line can be set to an idle (high) state.

# Asynchronous and Synchronous Modes

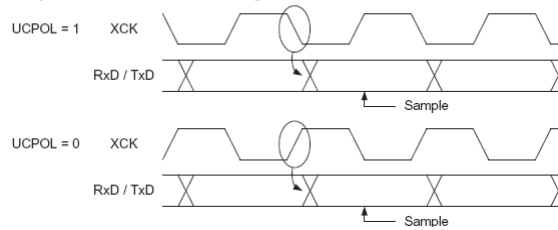
- Asynchronous transmission allows data to be transmitted without the sender having to send a clock signal to the receiver. Instead, the sender and receiver must agree on timing parameters in advance and special bits are added to each word which are used to synchronize the sending and receiving units.
- When a word is given to the USART for Asynchronous transmissions, a bit called the "Start Bit" is added to the beginning of each word that is to be transmitted. The Start Bit is used to alert the receiver that a word of data is about to be sent, and to force the clock in the receiver into synchronization with the clock in the transmitter. These two clocks must be accurate enough to not have the frequency drift by more than 10% during the transmission of the remaining bits in the word.



# Asynchronous and Synchronous Modes

- In *synchronous* transmission, the clock data is recovered separately from the data stream and no start/stop bits are used. This improves the efficiency of transmission on suitable channels since more of the bits sent are usable data and not character framing.

Figure 71. Synchronous Mode XCK Timing.



# Asynchronous and Synchronous Modes

- The UCPOL bit UCRSC selects which XCK clock edge is used for data sampling and which is used for data change.
- When UCPOL is zero the data will be changed at rising XCK edge and sampled at falling XCK edge. If UCPOL is set, the data will be changed at falling XCK edge and sampled at rising XCK edge.

# Asynchronous Data Recovery

- The data is sampled 8 (double speed mode) or 16 times the baudrate (normal mode) . A majority decision is taken based on the values of the 8,9 and 10 samples (normal mode) or 4,5 and 6 samples (double speed mode)

**Figure 73.** Start Bit Sampling

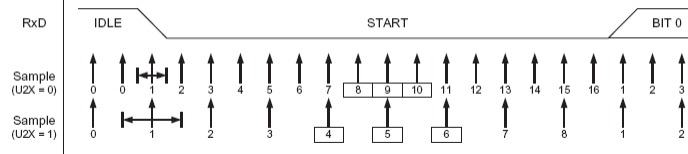


Figure 74. Sampling of Data and Parity Bit

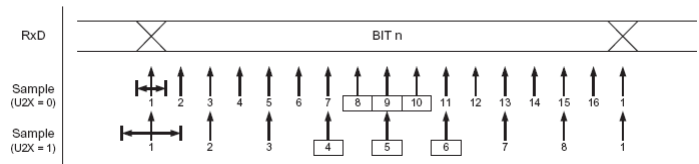
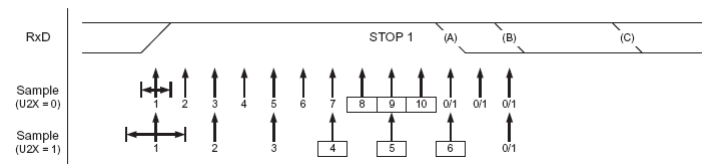


Figure 75. Stop Bit Sampling and Next Start Bit Sampling



# Computing BAUD Rates

**Table 60.** Equations for Calculating Baud Rate Register Setting

Operating Mode	Equation for Calculating Baud Rate <sup>(1)</sup>	Equation for Calculating UBRR Value
Asynchronous Normal Mode (U2X = 0)	$BAUD = \frac{f_{OSC}}{16(UBRR + 1)}$	$UBRR = \frac{f_{OSC}}{16BAUD} - 1$
Asynchronous Double Speed Mode (U2X = 1)	$BAUD = \frac{f_{OSC}}{8(UBRR + 1)}$	$UBRR = \frac{f_{OSC}}{8BAUD} - 1$
Synchronous Master Mode	$BAUD = \frac{f_{OSC}}{2(UBRR + 1)}$	$UBRR = \frac{f_{OSC}}{2BAUD} - 1$

Note: 1. The baud rate is defined to be the transfer rate in bit per second (bps).

# USART Initialization

- The USART has to be initialized before any communication can take place. The initialization process normally consists of setting the baud rate, setting frame format and enabling the Transmitter or the Receiver depending on the usage. For interrupt driven USART operation, the Global Interrupt Flag should be cleared (and interrupts globally disabled) when doing the initialization.

# USART Initialization

```
#define FOSC 1843200// Clock Speed
#define BAUD 9600
#define MYUBRR FOSC/16/BAUD-1
void main( void )
{
    ...
    USART_Init ( MYUBRR );
    ...
}
void USART_Init( unsigned int ubrr)
{
    /* Set baud rate */
    UBRRH = (unsigned char)(ubrr>>8);
    UBRRL = (unsigned char)ubrr;
    /* Enable receiver and transmitter */
    UCSRB = (1<<RXEN) | (1<<TXEN);
    /* Set frame format: 8data, 2stop bit */
    UCSRC = (1<<URSEL) | (1<<USBS) | (3<<UCSZ0);
}
```

# USART Transmission

- A data transmission is initiated by loading the transmit buffer with the data to be transmitted. The CPU can load the transmit buffer by writing to the UDR I/O location. The buffered data in the transmit buffer will be moved to the Shift Register when the Shift Register is ready to send a new frame.
- The Shift Register is loaded with new data if it is in idle state (no ongoing transmission) or immediately after the last stop bit of the previous frame is transmitted.
- When the Shift Register is loaded with new data, it will transfer one complete frame at the rate given by the Baud Register, U2X bit or by XCK depending on mode of operation.



# USART Transmission

```
void USART_Transmit( unsigned char data )
{
    /* Wait for empty transmit buffer */
    while ( !( UCSRA & (1<<UDRE)) )
        ;
    /* Put data into buffer, sends the data */
    UDR = data;
}
```

Transmission of data less than 8 bits

# Transmitting 9 bits of data

## C Code Example<sup>(1)</sup>

```
void USART_Transmit( unsigned int data )
{
    /* Wait for empty transmit buffer */
    while ( !( UCSRA & (1<<UDRE)) )
    ;
    /* Copy 9th bit to TXB8 */
    UCSRB &= ~(1<<TXB8);
    if ( data & 0x0100 )
        UCSRB |= (1<<TXB8);
    /* Put data into buffer, sends the data */
    UDR = data;
}
```

- The 9<sup>th</sup> bit or MSB should be written to TXB8 bit ins UCSRB register before writing the lower byte into UDR.
- Note that the TXB8 is first cleared of earlier values before writing to it

# USART Receiver

- The USART Receiver is enabled by writing the Receive Enable (RXEN) bit in the UCSRB Register to one. When the receiver is enabled, the normal pin operation of the RxD pin is overridden by the USART and given the function as the receiver's serial input
- The receiver starts data reception when it detects a valid start bit. Each bit that follows the start bit will be sampled at the baud rate or XCK clock, and shifted into the receive Shift Register until the first stop bit of a frame is received. A second stop bit will be ignored by the receiver.
- When the first stop bit is received, i.e., a complete serial frame is present in the receive Shift Register, the contents of the Shift Register will be moved into the receive buffer.
- The receive buffer can then be read by reading the UDR I/O location.

# USART Receiver

C Code Example

```
unsigned char USART_Receive( void )
{
    /* Wait for data to be received */
    while ( !(UCSRA & (1<<RXC)) )
        ;
    /* Get and return received data from buffer */
    return UDR;
}
```

The function simply waits for data to be present in the receive buffer by checking the RXC Flag, before reading the buffer and returning the value.

# Receiving 9 bits of data

## C Code Example<sup>(1)</sup>

```
unsigned int USART_Receive( void )
{
    unsigned char status, resh, resl;
    /* Wait for data to be received */
    while ( !(UCSRA & (1<<RXC)) )
        ;
    /* Get status and 9th bit, then data */
    /* from buffer */
    status = UCSRA;
    resh = UCSRB;
    resl = UDR;
    /* If error, return -1 */
    if ( status & (1<<PE) | (1<<DOR) | (1<<PE) )
        return -1;
    /* Filter the 9th bit, then return */
    resh = (resh >> 1) & 0x01;
    return ((resh << 8) | resl);
}
```

- Note: <sup>1</sup> See "About Code Examples" on page 7
- The key is to read the status and 9<sup>th</sup> bit first and then the lower byte from UDR. The 9<sup>th</sup> bit is bit 1 of the UCSRB register, the bit RXB8

# USART Interrupts

- The USART transmitter has two flags that indicate its state: USART Data Register Empty (UDRE) and Transmit Complete (TXC). Both flags can be used for generating interrupts.
- The Data Register Empty (UDRE) Flag indicates whether the transmit buffer is ready to receive new data. This bit is set when the transmit buffer is empty, and cleared when the transmit buffer contains data to be transmitted that has not yet been moved into the Shift Register.
- When the Data Register empty Interrupt Enable (UDRIE) bit in UCSRB is written to one, the USART Data Register Empty Interrupt will be executed as long as UDRE is set (provided that global interrupts are enabled). UDRE is cleared by writing UDR.
- When interrupt-driven data transmission is used, the Data Register Empty Interrupt routine must either write new data to UDR in order to clear UDRE or disable the Data Register empty Interrupt, otherwise a new interrupt will occur once the interrupt routine terminates.

# USART Interrupts

- The Transmit Complete (TXC) Flag bit is set one when the entire frame in the transmit Shift Register has been shifted out and there are no new data currently present in the transmit buffer.
- The TXC Flag bit is automatically cleared when a transmit complete interrupt is executed, or it can be cleared by writing a one to its bit location.
- When the Transmit Complete Interrupt Enable (TXCIE) bit in UCSRB is set, the USART Transmit Complete Interrupt will be executed when the TXC Flag becomes set (provided that global interrupts are enabled).
- When the transmit complete interrupt is used, the interrupt handling routine does not have to clear the TXC Flag, this is done automatically when the interrupt is executed.

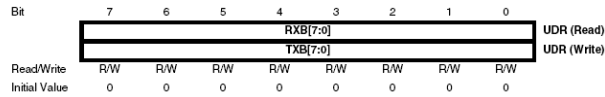
# USART Interrupts

- The USART Receiver has one flag that indicates the receiver state.
- The Receive Complete (RXC) Flag indicates if there are unread data present in the receive buffer. This flag is one when unread data exist in the receive buffer, and zero when the receive buffer is empty (i.e., does not contain any unread data).
- When the Receive Complete Interrupt Enable (RXCIE) in UCSRB is set, the USART Receive Complete Interrupt will be executed as long as the RXC Flag is set (provided that global interrupts are enabled).
- When interrupt-driven data reception is used, the receive complete routine must read the received data from UDR in order to clear the RXC Flag, otherwise a new interrupt will occur once the interrupt routine terminates.



# USART Registers

USART I/O Data  
Register – UDR



The USART Transmit Data Buffer Register and USART Receive Data Buffer Registers share the same I/O address referred to as USART Data Register or UDR. The Transmit Data Buffer Register (TXB) will be the destination for data written to the UDR Register location. Reading the UDR Register location will return the contents of the Receive Data Buffer Register (RXB).

The transmit buffer can only be written when the UDRE Flag in the UCSRA Register is set. Data written to UDR when the UDRE Flag is not set, will be ignored by the USART Transmitter. When data is written to the transmit buffer, and the Transmitter is enabled, the Transmitter will load the data into the transmit Shift Register when the Shift Register is empty. Then the data will be serially transmitted on the TxD pin.

## • USART Control and Status Reg A

Bit	7	6	5	4	3	2	1	0	
	<b>RXC</b>	<b>TXC</b>	<b>UDRE</b>	<b>FE</b>	<b>DOR</b>	<b>PE</b>	<b>U2X</b>	<b>MPCM</b>	UCSRA
Read/Write	R	R/W	R	R	R	R	R/W	R/W	
Initial Value	0	0	1	0	0	0	0	0	

### • Bit 7 – RXC: USART Receive Complete

This flag bit is set when there are unread data in the receive buffer and cleared when the receive buffer is empty (i.e., does not contain any unread data). If the receiver is disabled, the receive buffer will be flushed and consequently the RXC bit will become zero. The RXC Flag can be used to generate a Receive Complete interrupt (see description of the RXCIE bit).

### • Bit 6 – TXC: USART Transmit Complete

This flag bit is set when the entire frame in the transmit Shift Register has been shifted out and there are no new data currently present in the transmit buffer (UDR). The TXC Flag bit is automatically cleared when a transmit complete interrupt is executed, or it can be cleared by writing a one to its bit location. The TXC Flag can generate a Transmit Complete interrupt (see description of the TXCIE bit).

### • Bit 5 – UDRE: USART Data Register Empty

The UDRE Flag indicates if the transmit buffer (UDR) is ready to receive new data. If UDRE is one, the buffer is empty, and therefore ready to be written. The UDRE Flag can generate a Data Register empty Interrupt (see description of the UDRIE bit).

UDRE is set after a reset to indicate that the transmitter is ready.

Bit	7	6	5	4	3	2	1	0	
	<b>RXCIE</b>	<b>TXCIE</b>	<b>UDRIE</b>	<b>RXEN</b>	<b>TXEN</b>	<b>UCSZ2</b>	<b>RXB8</b>	<b>TXB8</b>	UCSRB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – RXCIE: RX Complete Interrupt Enable**

Writing this bit to one enables interrupt on the RXC Flag. A USART Receive Complete Interrupt will be generated only if the RXCIE bit is written to one, the Global Interrupt Flag in SREG is written to one and the RXC bit in UCSRA is set.

- **Bit 6 – TXCIE: TX Complete Interrupt Enable**

Writing this bit to one enables interrupt on the TXC Flag. A USART Transmit Complete Interrupt will be generated only if the TXCIE bit is written to one, the Global Interrupt Flag in SREG is written to one and the TXC bit in UCSRA is set.

- **Bit 5 – UDRIE: USART Data Register Empty Interrupt Enable**

Writing this bit to one enables interrupt on the UDRE Flag. A Data Register Empty Interrupt will be generated only if the UDRIE bit is written to one, the Global Interrupt Flag in SREG is written to one and the UDRE bit in UCSRA is set.

- **Bit 4 – RXEN: Receiver Enable**

Writing this bit to one enables the USART Receiver. The Receiver will override normal port operation for the RxD pin when enabled. Disabling the Receiver will flush the receive buffer invalidating the FE, DOR, and PE Flags.

- **Bit 3 – TXEN: Transmitter Enable**

Writing this bit to one enables the USART Transmitter. The Transmitter will override normal port operation for the TxD pin when enabled. The disabling of the Transmitter (writing TXEN to zero) will not become effective until ongoing and pending transmissions are completed, i.e., when the transmit Shift Register and transmit Buffer Register do not contain data to be transmitted. When disabled, the transmitter will no longer override the TxD port.

- **Bit 2 – UCSZ2: Character Size**

The UCSZ2 bits combined with the UCSZ1:0 bit in UCSRC sets the number of data bits (Char-

# USCRC

Bit	7	6	5	4	3	2	1	0	
	URSEL	UMSEL	UPM1	UPM0	USBS	UCSZ1	UCSZ0	UCPOL	USCRC
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	1	0	0	0	0	1	1	0	

- **Bit 7 - URSEL: Register Select:** This bit selects between accessing the USCRC or the UBRRH Register. It is read as one when reading USCRC. The URSEL must be one when writing the USCRC

Table 65. USBS Bit Settings

USBS	Stop Bit(s)
0	1-bit
1	2-bit

# USCRC

**Table 66.** UCSZ Bits Settings

UCSZ2	UCSZ1	UCSZ0	Character Size
0	0	0	5-bit
0	0	1	6-bit
0	1	0	7-bit
0	1	1	8-bit
1	0	0	Reserved
1	0	1	Reserved
1	1	0	Reserved
1	1	1	9-bit

# Accessing USCRC and UBRRH

- The USCRC and UBRRH share the same I/O location
- the USART Register Select (URSEL) bit, controls which one of the two registers that will be written. If URSEL is zero during a write operation, the UBRRH value will be updated. If URSEL is one, the USCRC setting will be updated.

```
...
/* Set UBRRH to 2 */
UBRRH = 0x02;
...
/* Set the USBS and the UCSZ1 bit to one, and */
/* the remaining bits to zero. */
USCRC = (1<<URSEL) | (1<<USBS) | (1<<UCSZ1);
...
```

- In the above code, initial URSEL bit is low and hence the UBRRH value gets updated. Next while writing to USCRC