Section 18. USART

HIGHLIGHTS

This section of the manual contains the following major topics:

18.1 Introduction

The Universal Synchronous Asynchronous Receiver Transmitter (USART) module is one of the two serial I/O modules (other is the SSP module). The USART is also known as a Serial Communications Interface or SCI. The USART can be configured as a full duplex asynchronous system that can communicate with peripheral devices such as CRT terminals and personal computers, or it can be configured as a half duplex synchronous system that can communicate with peripheral devices such as A/D or D/A integrated circuits, Serial EEPROMs etc.

The USART can be configured in the following modes:

- Asynchronous (full duplex)
- Synchronous - Master (half duplex)
- Synchronous - Slave (half duplex)

The SPEN bit (RCSTA<7>), and the TRIS bits, have to be set in order to configure the TX/CK and RX/DT pins for the USART.

18.2 Control Registers

Register 18-1: TXSTA: Transmit Status and Control Register

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R-1	R/W-0
CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D
bit 7							bit 0

bit 7 **CSRC:** Clock Source Select bit
Asynchronous mode
Don't care

Synchronous mode
1 = Master mode (Clock generated internally from BRG)
0 = Slave mode (Clock from external source)

bit 6 **TX9**: 9-bit Transmit Enable bit
1 = Selects 9-bit transmission
0 = Selects 8-bit transmission

bit 5 **TXEN**: Transmit Enable bit
1 = Transmit enabled
0 = Transmit disabled

Note: SREN/CREN overrides TXEN in SYNC mode.

bit 4 **SYNC**: USART Mode Select bit
1 = Synchronous mode
0 = Asynchronous mode

bit 3 **Unimplemented:** Read as '0'

bit 2 **BRGH**: High Baud Rate Select bit
Asynchronous mode
1 = High speed
0 = Low speed

Synchronous mode
Unused in this mode

bit 1 **TRMT**: Transmit Shift Register Status bit
1 = TSR empty
0 = TSR full

bit 0 **TX9D:** 9th bit of transmit data. Can be parity bit.

Legend	
R = Readable bit	W = Writable bit
U = Unimplemented bit, read as ‘0’	- n = Value at POR reset

18

USART

Register 18-2: RCSTA: Receive Status and Control Register

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R-0	R-0	R-0
SPEN	RX9	SREN	CREN	—	FERR	OERR	RX9D
bit 7							bit 0

bit 7 **SPEN:** Serial Port Enable bit
1 = Serial port enabled (Configures RX/DT and TX/CK pins as serial port pins)
0 = Serial port disabled

bit 6 **RX9**: 9-bit Receive Enable bit
1 = Selects 9-bit reception
0 = Selects 8-bit reception

bit 5 **SREN**: Single Receive Enable bit
Asynchronous mode
Don’t care

Synchronous mode - master
1 = Enables single receive
0 = Disables single receive
This bit is cleared after reception is complete.

Synchronous mode - slave
Unused in this mode

bit 4 **CREN**: Continuous Receive Enable bit
Asynchronous mode
1 = Enables continuous receive
0 = Disables continuous receive

Synchronous mode
1 = Enables continuous receive until enable bit CREN is cleared (CREN overrides SREN)
0 = Disables continuous receive

bit 3 **Unimplemented:** Read as '0'

bit 2 **FERR**: Framing Error bit
1 = Framing error (Can be updated by reading RCREG register and receive next valid byte)
0 = No framing error

bit 1 **OERR**: Overrun Error bit
1 = Overrun error (Can be cleared by clearing bit CREN)
0 = No overrun error

bit 0 **RX9D:** 9th bit of received data, can be parity bit.

Legend	
R = Readable bit	W = Writable bit
U = Unimplemented bit, read as ‘0’	- n = Value at POR reset

18.3 USART Baud Rate Generator (BRG)

The BRG supports both the Asynchronous and Synchronous modes of the USART. It is a dedicated 8-bit baud rate generator. The SPBRG register controls the period of a free running 8-bit timer. In asynchronous mode bit BRGH (TXSTA<2>) also controls the baud rate. In synchronous mode bit BRGH is ignored. Table 18-1 shows the formula for computation of the baud rate for different USART modes which only apply in master mode (internal clock).

Given the desired baud rate and Fosc, the nearest integer value for the SPBRG register can be calculated using the formula in Table 18-1, where X equals the value in the SPBRG register (0 to 255). From this, the error in baud rate can be determined.

Table 18-1: Baud Rate Formula

SYNC	BRGH = 0 (Low Speed)	BRGH = 1 (High Speed)
0	(Asynchronous) Baud Rate = FOSC/(64(X+1))	Baud Rate= FOSC/(16(X+1))
1	(Synchronous) Baud Rate = FOSC/(4(X+1))	NA

X = value in SPBRG (0 to 255)

Example 18-1 shows the calculation of the baud rate error for the following conditions:

FOSC = 16 MHz
Desired Baud Rate = 9600
BRGH = 0
SYNC = 0

Example 18-1: Calculating Baud Rate Error

$$\text{Desired Baud rate} = \text{Fosc} / (64 (X + 1))$$
$$9600 = 16000000 / (64 (X + 1))$$
$$X = \lfloor 25.042 \rfloor = 25$$
$$\text{Calculated Baud Rate} = 16000000 / (64 (25 + 1)) = 9615$$
$$\text{Error} = \frac{\text{(Calculated Baud Rate - Desired Baud Rate)}}{\text{Desired Baud Rate}} = (9615 - 9600) / 9600 = 0.16\%$$

It may be advantageous to use the high baud rate (BRGH = 1) even for slower baud clocks. This is because the FOSC / (16(X + 1)) equation can reduce the baud rate error in some cases.

Writing a new value to the SPBRG register causes the BRG timer to be reset (or cleared). This ensures the BRG does not wait for a timer overflow before outputting the new baud rate.

Table 18-2: Registers Associated with Baud Rate Generator

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other resets
TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
RCSTA	SPEN	RX9	SREN	CREN	—	FERR	OERR	RX9D	0000 -00x	0000 -00x
SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, - = unimplemented read as '0'. Shaded cells are not used by the BRG.

18
USART

Table 18-3: Baud Rates for Synchronous Mode

BAUD RATE (Kbps)	Fosc = 20 MHz KBAUD	% ERROR	SPBRG value (decimal)	16 MHz KBAUD	% ERROR	SPBRG value (decimal)	10 MHz KBAUD	% ERROR	SPBRG value (decimal)	7.15909 MHz KBAUD	% ERROR	SPBRG value (decimal)
0.3	NA	-	-	NA	-	-	NA	-	-	NA	-	-
1.2	NA	-	-	NA	-	-	NA	-	-	NA	-	-
2.4	NA	-	-	NA	-	-	NA	-	-	NA	-	-
9.6	NA	-	-	NA	-	-	9.766	+1.73	255	9.622	+0.23	185
19.2	19.53	+1.73	255	19.23	+0.16	207	19.23	+0.16	129	19.24	+0.23	92
76.8	76.92	+0.16	64	76.92	+0.16	51	75.76	-1.36	32	77.82	+1.32	22
96	96.15	+0.16	51	95.24	-0.79	41	96.15	+0.16	25	94.20	-1.88	18
300	294.1	-1.96	16	307.69	+2.56	12	312.5	+4.17	7	298.3	-0.57	5
500	500	0	9	500	0	7	500	0	4	NA	-	-
HIGH	5000	-	0	4000	-	0	2500	-	0	1789.8	-	0
LOW	19.53	-	255	15.625	-	255	9.766	-	255	6.991	-	255

BAUD RATE (Kbps)	Fosc = 5.0688 MHz KBAUD	% ERROR	SPBRG value (decimal)	4 MHz KBAUD	% ERROR	SPBRG value (decimal)	3.579545 MHz KBAUD	% ERROR	SPBRG value (decimal)	1 MHz KBAUD	% ERROR	SPBRG value (decimal)	32.768 kHz KBAUD	% ERROR	SPBRG value (decimal)
0.3	NA	-	-	NA	-	-	NA	-	-	NA	-	-	0.303	+1.14	26
1.2	NA	-	-	NA	-	-	NA	-	-	1.202	+0.16	207	1.170	-2.48	6
2.4	NA	-	-	NA	-	-	NA	-	-	2.404	+0.16	103	NA	-	-
9.6	9.6	0	131	9.615	+0.16	103	9.622	+0.23	92	9.615	+0.16	25	NA	-	-
19.2	19.2	0	65	19.231	+0.16	51	19.04	-0.83	46	19.24	+0.16	12	NA	-	-
76.8	79.2	+3.13	15	76.923	+0.16	12	74.57	-2.90	11	83.34	+8.51	2	NA	-	-
96	97.48	+1.54	12	1000	+4.17	9	99.43	+3.57	8	NA	-	-	NA	-	-
300	316.8	+5.60	3	NA	-	-	298.3	-0.57	2	NA	-	-	NA	-	-
500	NA	-	-	NA	-	-	NA	-	-	NA	-	-	NA	-	-
HIGH	1267	-	0	100	-	0	894.9	-	0	250	-	0	8.192	-	0
LOW	4.950	-	255	3.906	-	255	3.496	-	255	0.9766	-	255	0.032	-	255

Table 18-4: Baud Rates for Asynchronous Mode (BRGH = 0)

BAUD RATE (Kbps)	FOSC = 20 MHz KBAUD	% ERROR	SPBRG value (decimal)	16 MHz KBAUD	% ERROR	SPBRG value (decimal)	10 MHz KBAUD	% ERROR	SPBRG value (decimal)	7.15909 MHz KBAUD	% ERROR	SPBRG value (decimal)
0.3	NA	-	-	NA	-	-	NA	-	-	NA	-	-
1.2	1.221	+1.73	255	1.202	+0.16	207	1.202	+0.16	129	1.203	+0.23	92
2.4	2.404	+0.16	129	2.404	+0.16	103	2.404	+0.16	64	2.380	-0.83	46
9.6	9.469	-1.36	32	9.615	+0.16	25	9.766	+1.73	15	9.322	-2.90	11
19.2	19.53	+1.73	15	19.23	+0.16	12	19.53	+1.73	7	18.64	-2.90	5
76.8	78.13	+1.73	3	83.33	+8.51	2	78.13	+1.73	1	NA	-	-
96	104.2	+8.51	2	NA	-	-	NA	-	-	NA	-	-
300	312.5	+4.17	0	NA	-	-	NA	-	-	NA	-	-
500	NA	-	-	NA	-	-	NA	-	-	NA	-	-
HIGH	312.5	-	0	250	-	0	156.3	-	0	111.9	-	0
LOW	1.221	-	255	0.977	-	255	0.6104	-	255	0.437	-	255

BAUD RATE (Kbps)	FOSC = 5.0688 MHz KBAUD	% ERROR	SPBRG value (decimal)	4 MHz KBAUD	% ERROR	SPBRG value (decimal)	3.579545 MHz KBAUD	% ERROR	SPBRG value (decimal)	1 MHz KBAUD	% ERROR	SPBRG value (decimal)	32.768 kHz KBAUD	% ERROR	SPBRG value (decimal)
0.3	0.31	+3.13	255	0.3005	-0.17	207	0.301	+0.23	185	0.300	+0.16	51	0.256	-14.67	1
1.2	1.2	0	65	1.202	+1.67	51	1.190	-0.83	46	1.202	+0.16	12	NA	-	-
2.4	2.4	0	32	2.404	+1.67	25	2.432	+1.32	22	2.232	-6.99	6	NA	-	-
9.6	9.9	+3.13	7	NA	-	-	9.322	-2.90	5	NA	-	-	NA	-	-
19.2	19.8	+3.13	3	NA	-	-	18.64	-2.90	2	NA	-	-	NA	-	-
76.8	79.2	+3.13	0	NA	-	-	NA	-	-	NA	-	-	NA	-	-
96	NA	-	-	NA	-	-	NA	-	-	NA	-	-	NA	-	-
300	NA	-	-	NA	-	-	NA	-	-	NA	-	-	NA	-	-
500	NA	-	-	NA	-	-	NA	-	-	NA	-	-	NA	-	-
HIGH	79.2	-	0	62.500	-	0	55.93	-	0	15.63	-	0	0.512	-	0
LOW	0.3094	-	255	3.906	-	255	0.2185	-	255	0.0610	-	255	0.0020	-	255

Table 18-5: Baud Rates for Asynchronous Mode (BRGH = 1)

BAUD RATE (Kbps)	FOSC = 20 MHz KBAUD	% ERROR	SPBRG value (decimal)	16 MHz KBAUD	% ERROR	SPBRG value (decimal)	10 MHz KBAUD	% ERROR	SPBRG value (decimal)	7.15909 MHz KBAUD	% ERROR	SPBRG value (decimal)
9.6	9.615	+0.16	129	9.615	+0.16	103	9.615	+0.16	64	9.520	-0.83	46
19.2	19.230	+0.16	64	19.230	+0.16	51	18.939	-1.36	32	19.454	+1.32	22
38.4	37.878	-1.36	32	38.461	+0.16	25	39.062	+1.7	15	37.286	-2.90	11
57.6	56.818	-1.36	21	58.823	+2.12	16	56.818	-1.36	10	55.930	-2.90	7
115.2	113.636	-1.36	10	111.111	-3.55	8	125	+8.51	4	111.860	-2.90	3
250	250	0	4	250	0	3	NA	-	-	NA	-	-
625	625	0	1	NA	-	-	625	0	0	NA	-	-
1250	1250	0	0	NA	-	-	NA	-	-	NA	-	-

BAUD RATE (Kbps)	FOSC = 5.0688 MHz KBAUD	% ERROR	SPBRG value (decimal)	4 MHz KBAUD	% ERROR	SPBRG value (decimal)	3.579545 MHz KBAUD	% ERROR	SPBRG value (decimal)	1 MHz KBAUD	% ERROR	SPBRG value (decimal)	32.768 kHz KBAUD	% ERROR	SPBRG value (decimal)
9.6	9.6	0	32	NA	-	-	9.727	+1.32	22	8.928	-6.99	6	NA	-	-
19.2	18.645	-2.94	16	1.202	+0.17	207	18.643	-2.90	11	20.833	+8.51	2	NA	-	-
38.4	39.6	+3.12	7	2.403	+0.13	103	37.286	-2.90	5	31.25	-18.61	1	NA	-	-
57.6	52.8	-8.33	5	9.615	+0.16	25	55.930	-2.90	3	62.5	+8.51	0	NA	-	-
115.2	105.6	-8.33	2	19.231	+0.16	12	111.860	-2.90	1	NA	-	-	NA	-	-
250	NA	-	-	NA	-	-	223.721	-10.51	0	NA	-	-	NA	-	-
625	NA	-	-	NA	-	-	NA	-	-	NA	-	-	NA	-	-
1250	NA	-	-	NA	-	-	NA	-	-	NA	-	-	NA	-	-

18.4 USART Asynchronous Mode

In this mode, the USART uses standard nonreturn-to-zero (NRZ) format (one start bit, eight or nine data bits and one stop bit). The most common data format is 8-bits. An on-chip dedicated 8-bit baud rate generator can be used to derive standard baud rate frequencies from the oscillator. The USART transmits and receives the LSb first. The USART's transmitter and receiver are functionally independent but use the same data format and baud rate. The baud rate generator produces a clock either x16 or x64 of the bit shift rate, depending on the BRGH bit (TXSTA<2>). Parity is not supported by the hardware, but can be implemented in software (stored as the ninth data bit). Asynchronous mode is stopped during SLEEP.

Asynchronous mode is selected by clearing the SYNC bit (TXSTA<4>).

The USART Asynchronous module consists of the following important elements:

- Baud Rate Generator
- Sampling Circuit
- Asynchronous Transmitter
- Asynchronous Receiver

18.4.1 USART Asynchronous Transmitter

The USART transmitter block diagram is shown in Figure 18-1. The heart of the transmitter is the transmit (serial) shift register (TSR). The shift register obtains its data from the read/write transmit buffer, TXREG. The TXREG register is loaded with data in software. The TSR register is not loaded until the STOP bit has been transmitted from the previous load. As soon as the STOP bit is transmitted, the TSR is loaded with new data from the TXREG register (if available). Once the TXREG register transfers the data to the TSR register (occurs in one T_{CY}), the TXREG register is empty and the TXIF flag bit is set. This interrupt can be enabled/disabled by setting/clearing the TXIE enable bit. The TXIF flag bit will be set regardless of the state of the TXIE enable bit and cannot be cleared in software. It will reset only when new data is loaded into the TXREG register. While the TXIF flag bit indicated the status of the TXREG register, the TRMT bit (TXSTA<1>) shows the status of the TSR register. The TRMT status bit is a read only bit which is set when the TSR register is empty. No interrupt logic is tied to this bit, so the user has to poll this bit in order to determine if the TSR register is empty.

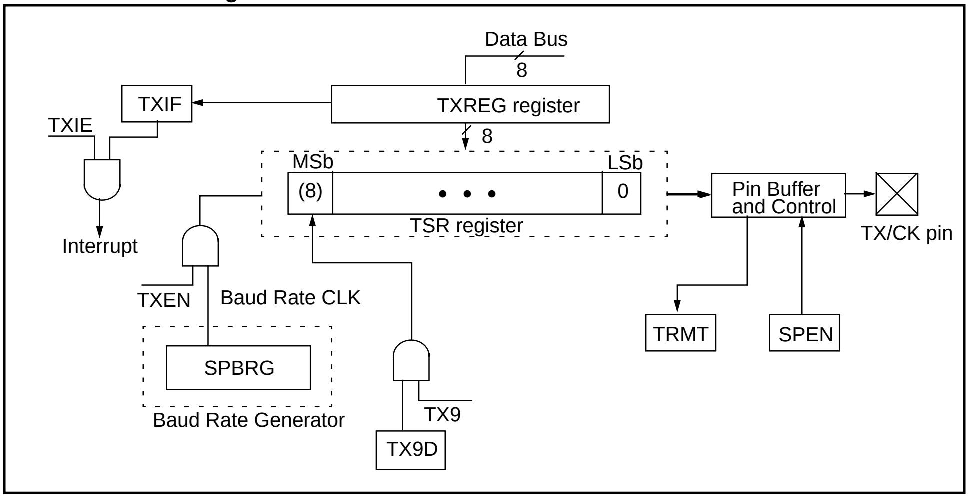
Note 1: The TSR register is not mapped in data memory so it is not available to the user.

Note 2: When the TXEN bit is set, the TXIF flag bit will also be set since the transmit buffer is not yet full (still can move transmit data to the TXREG register).

Transmission is enabled by setting the TXEN enable bit (TXSTA<5>). The actual transmission will not occur until the TXREG register has been loaded with data and the baud rate generator (BRG) has produced a shift clock (Figure 18-1). The transmission can also be started by first loading the TXREG register and then setting the TXEN enable bit. Normally when transmission is first started, the TSR register is empty, so a transfer to the TXREG register will result in an immediate transfer to TSR resulting in an empty TXREG. A back-to-back transfer is thus possible (Figure 18-3). Clearing the TXEN enable bit during a transmission will cause the transmission to be aborted and will reset the transmitter. As a result the TX/CK pin will revert to hi-impedance.

In order to select 9-bit transmission, transmit bit, TX9 (TXSTA<6>), should be set and the ninth bit should be written to the TX9D bit (TXSTA<0>). The ninth bit must be written before writing the 8-bit data to the TXREG register. This is because a data write to the TXREG register can result in an immediate transfer of the data to the TSR register (if the TSR is empty). In such a case, an incorrect ninth data bit maybe loaded in the TSR register.

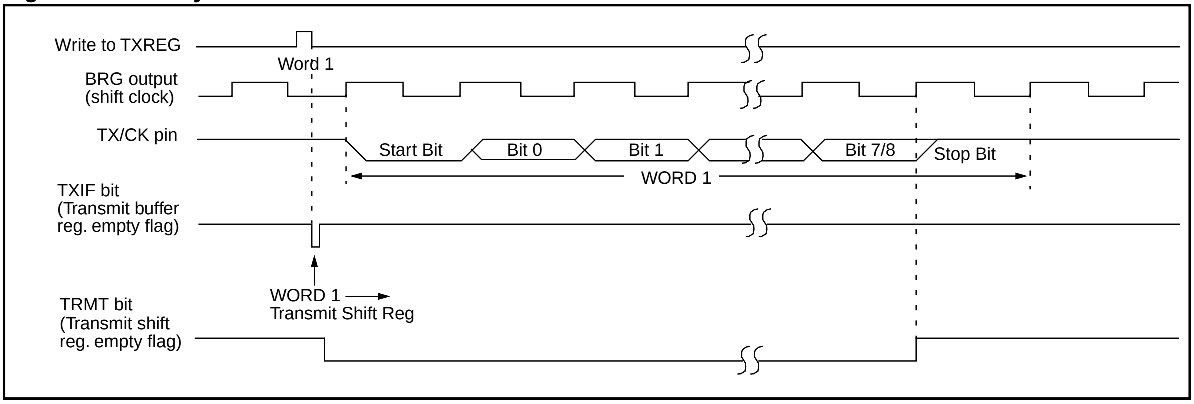
Figure 18-1: USART Transmit Block Diagram



Steps to follow when setting up a Asynchronous Transmission:

1. Initialize the SPBRG register for the appropriate baud rate. If a high speed baud rate is desired, set the BRGH bit. (Subsection **18.3 "USART Baud Rate Generator (BRG)"**)
2. Enable the asynchronous serial port by clearing the SYNC bit and setting the SPEN bit.
3. If interrupts are desired, then set the TXIE, GIE and PEIE bits.
4. If 9-bit transmission is desired, then set the TX9 bit.
5. Enable the transmission by setting the TXEN bit, which will also set the TXIF bit.
6. If 9-bit transmission is selected, the ninth bit should be loaded in the TX9D bit.
7. Load data to the TXREG register (starts transmission).

Figure 18-2: Asynchronous Master Transmission



18

USART

Figure 18-3: Asynchronous Master Transmission (Back to Back)

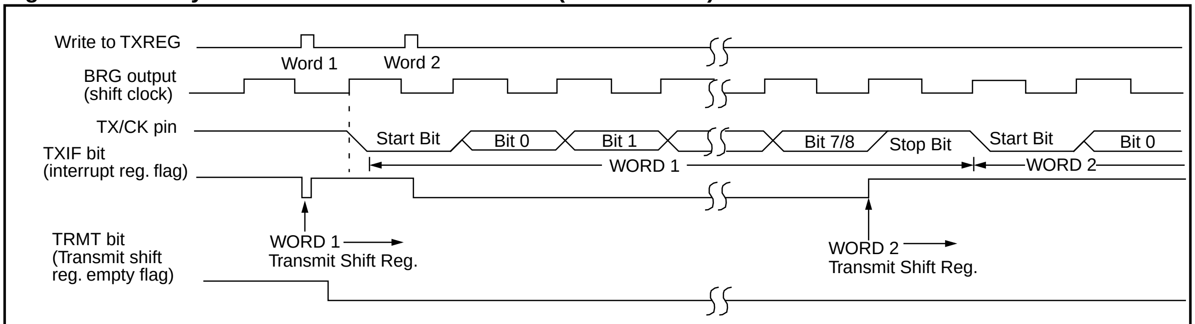


Table 18-6: Registers Associated with Asynchronous Transmission

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets
PIR	TXIF (1)								0	0
RCSTA	SPEN	RX9	SREN	CREN	—	FERR	OERR	RX9D	0000 -00x	0000 -00x
TXREG	TX7	TX6	TX5	TX4	TX3	TX2	TX1	TX0	0000 0000	0000 0000
PIE	TXIE (1)								0	0
TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, - = unimplemented locations read as '0'.
Shaded cells are not used for Asynchronous Transmission.
Note 1: The position of this bit is device dependent.

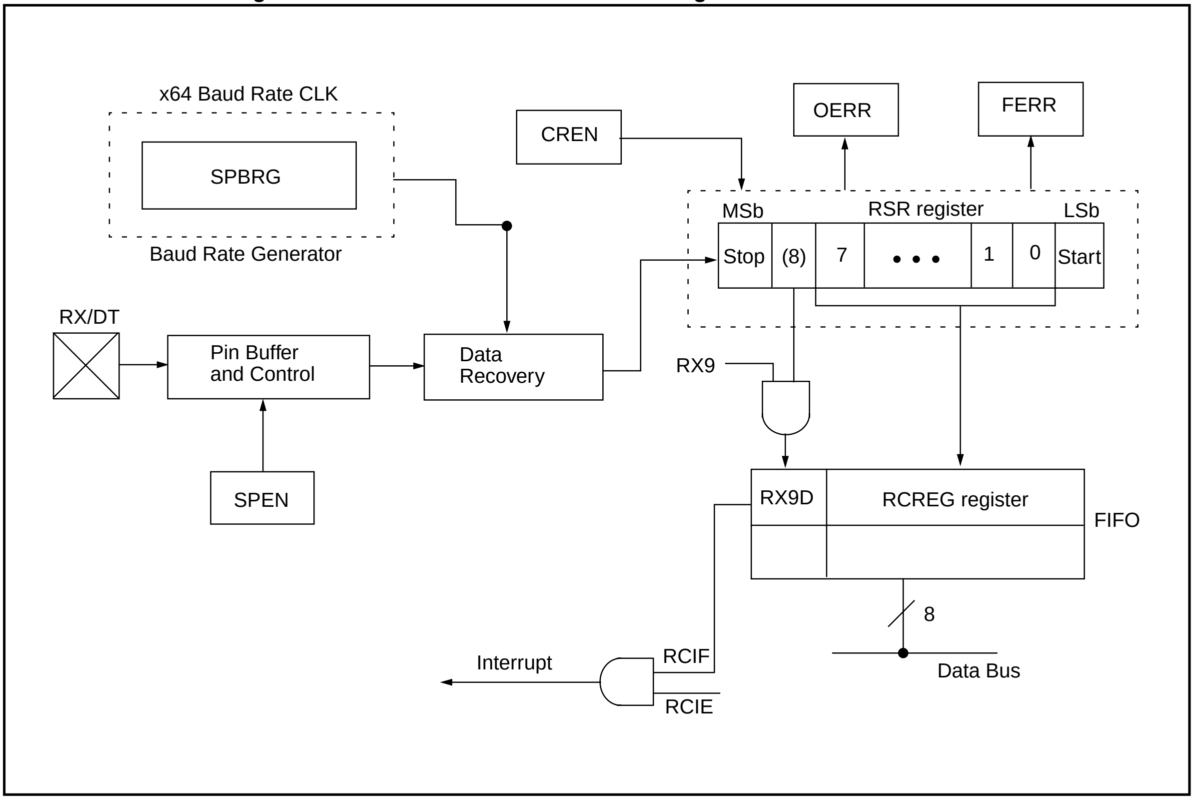
18.4.2 USART Asynchronous Receiver

The receiver block diagram is shown in Figure 18-4. The data is received on the RX/DT pin and drives the data recovery block. The data recovery block is actually a high speed shifter operating at x16 times the baud rate, whereas the main receive serial shifter operates at the bit rate or at FOSC.

Once Asynchronous mode is selected, reception is enabled by setting the CREN bit (RCSTA<4>).

The heart of the receiver is the receive (serial) shift register (RSR). After sampling the RX/TX pin for the STOP bit, the received data in the RSR is transferred to the RCREG register (if it is empty). If the transfer is complete, the RCIF flag bit is set. The actual interrupt can be enabled/disabled by setting/clearing the RCIE enable bit. The RCIF flag bit is a read only bit which is cleared by the hardware. It is cleared when the RCREG register has been read and is empty. The RCREG is a double buffered register, i.e. it is a two deep FIFO. It is possible for two bytes of data to be received and transferred to the RCREG FIFO and a third byte begin shifting to the RSR register. On the detection of the STOP bit of the third byte, if the RCREG register is still full then overrun error bit, OERR (RCSTA<1>), will be set. The word in the RSR will be lost. The RCREG register can be read twice to retrieve the two bytes in the FIFO. The OERR bit has to be cleared in software. This is done by resetting the receive logic (the CREN bit is cleared and then set). If the OERR bit is set, transfers from the RSR register to the RCREG register are inhibited, so it is essential to clear the OERR bit if it is set. Framing error bit, FERR (RCSTA<2>), is set if a stop bit is detected as a low level. The FERR bit and the 9th receive bit are buffered the same way as the receive data. Reading the RCREG will load the RX9D and FERR bits with new values, therefore it is essential for the user to read the RCSTA register before reading the next RCREG register in order not to lose the old (previous) information in the FERR and RX9D bits.

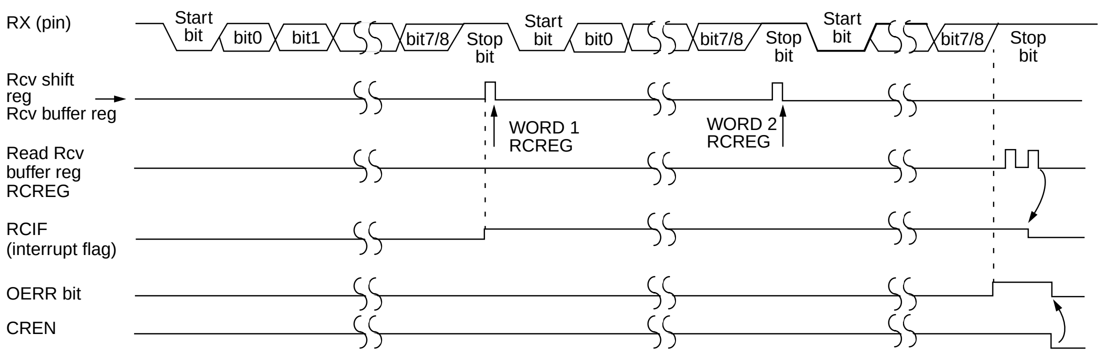
Figure 18-4: USART Receive Block Diagram



Steps to follow when setting up an Asynchronous Reception:

1. Initialize the SPBRG register for the appropriate baud rate. If a high speed baud rate is desired, set bit BRGH. (Subsection **18.3 "USART Baud Rate Generator (BRG)"**).
2. Enable the asynchronous serial port by clearing the SYNC bit, and setting the SPEN bit.
3. If interrupts are desired, then set the RCIE, GIE and PEIE bits.
4. If 9-bit reception is desired, then set the RX9 bit.
5. Enable the reception by setting the CREN bit.
6. The RCIF flag bit will be set when reception is complete and an interrupt will be generated if the RCIE bit was set.
7. Read the RCSTA register to get the ninth bit (if enabled) and determine if any error occurred during reception.
8. Read the 8-bit received data by reading the RCREG register.
9. If any error occurred, clear the error by clearing the CREN bit.

Figure 18-5: Asynchronous Reception

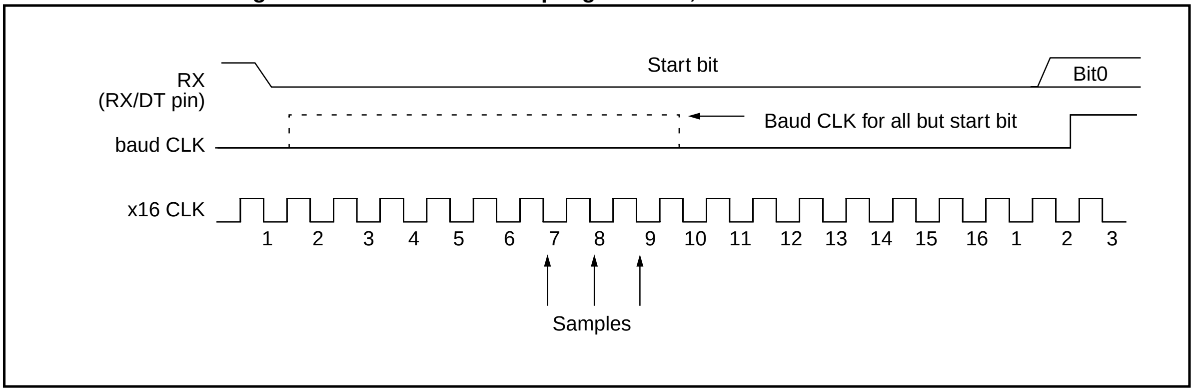


Note: This timing diagram shows three words appearing on the RX input. The RCREG (receive buffer) is read after the third word, causing the OERR (overrun) bit to be set.

18.4.3 Sampling

The data on the RX/DT pin is sampled three times by a majority detect circuit to determine if a high or a low level is present at the RX pin. Figure 18-6 shows the waveform for the sampling circuit. The sampling operates the same regardless of the state of the BRGH bit, only the source of the x16 clock is different.

Figure 18-6: RX Pin Sampling Scheme, BRGH = 0 or BRGH = 1



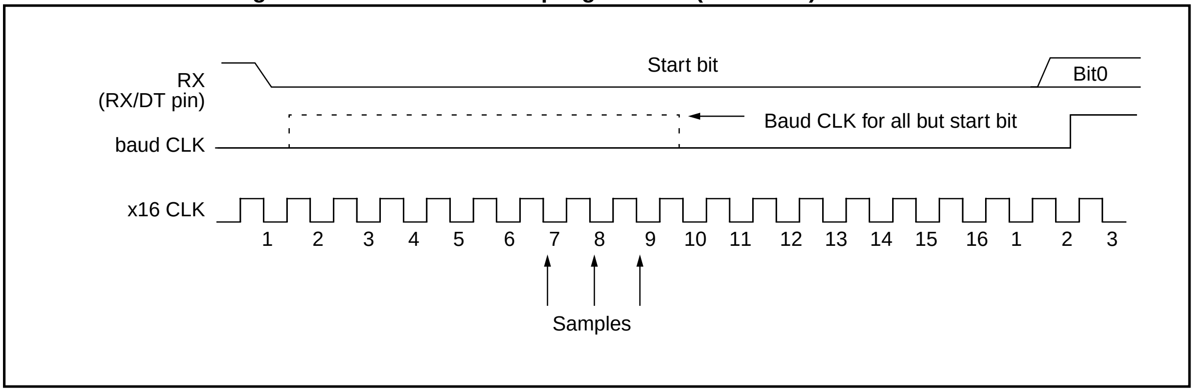
18.4.3.1 Device Exceptions

All new devices will use the sampling scheme shown in Figure 18-6. Devices that have an exception to the above sampling scheme are:

- PIC16C63
- PIC16C65
- PIC16C65A
- PIC16C73
- PIC16C73A
- PIC16C74
- PIC16C74A

These devices have a sampling circuitry that works as follows. If the BRGH bit (TXSTA<2>) is clear (i.e., at the low baud rates), the sampling is done on the seventh, eighth and ninth falling edges of a x16 clock (Figure 18-7). If bit BRGH is set (i.e., at the high baud rates), the sampling is done on the 3 clock edges preceding the second rising edge after the first falling edge of a x4 clock (Figure 18-8 and Figure 18-9).

Figure 18-7: RX Pin Sampling Scheme (BRGH = 0)



18
USART

Figure 18-8: RX Pin Sampling Scheme (BRGH = 1)

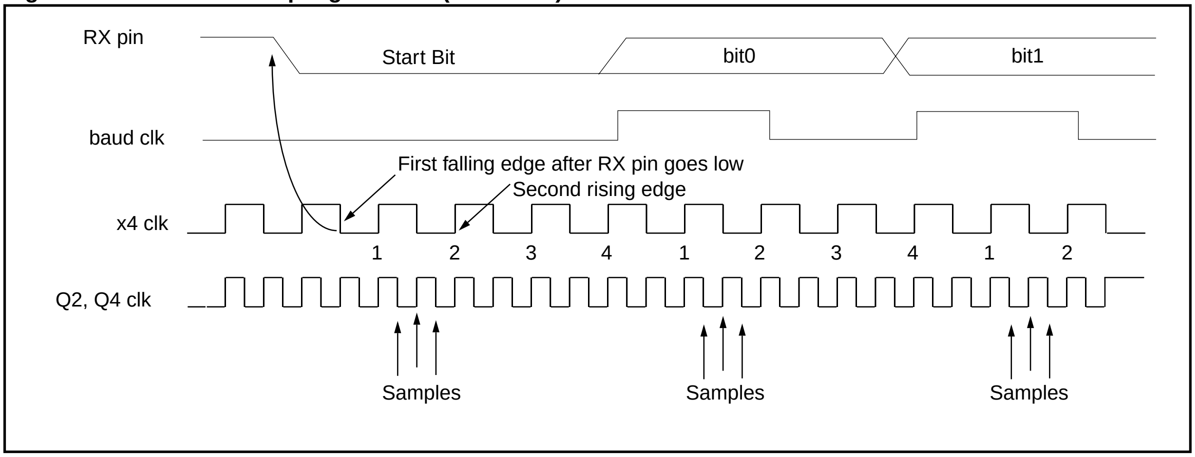


Figure 18-9: RX Pin Sampling Scheme (BRGH = 1)

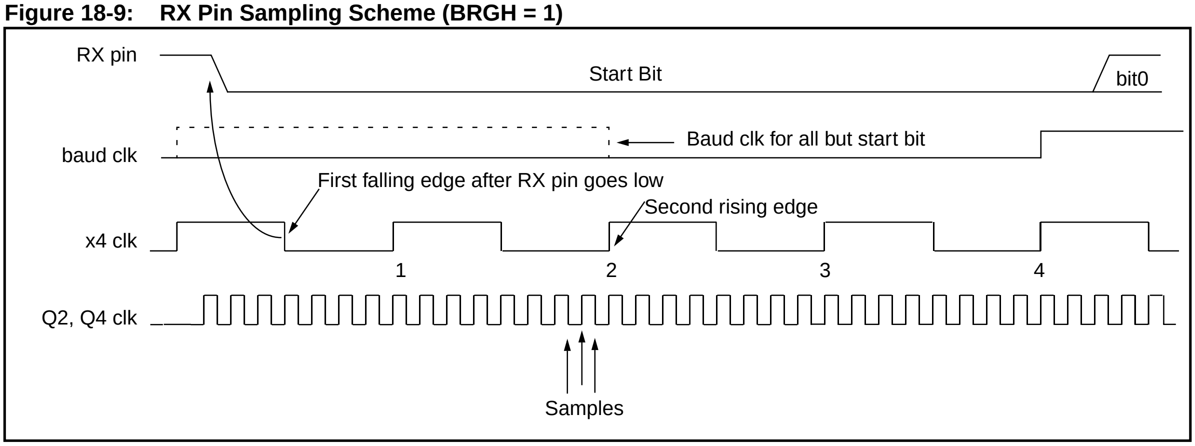


Table 18-7: Registers Associated with Asynchronous Reception

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets
PIR	RCIF (1)								`0`	`0`
RCSTA	SPEN	RX9	SREN	CREN	—	FERR	OERR	RX9D	`0000 -00x`	`0000 -00x`
RCREG	RX7	RX6	RX5	RX4	RX3	RX2	RX1	RX0	`0000 0000`	`0000 0000`
PIE	RCIE (1)								`0`	`0`
TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	`0000 -010`	`0000 -010`
SPBRG	Baud Rate Generator Register								`0000 0000`	`0000 0000`

Legend: x = unknown, - = unimplemented locations read as '0'.
Shaded cells are not used for Asynchronous Reception.

Note 1: The position of this bit is device dependent.

18.5 USART Synchronous Master Mode

In Synchronous Master mode, the data is transmitted in a half-duplex manner, i.e. transmission and reception do not occur at the same time. When transmitting data, the reception is inhibited and vice versa. Synchronous mode is entered by setting the SYNC bit (TXSTA<4>). In addition, the SPEN enable bit (RCSTA<7>) is set in order to configure the TX/CK and RX/DT I/O pins to CK (clock) and DT (data) lines respectively. The Master mode indicates that the processor transmits the master clock on the CK line. The Master mode is entered by setting the CSRC bit (TXSTA<7>).

18.5.1 USART Synchronous Master Transmission

The USART transmitter block diagram is shown in Figure 18-1. The heart of the transmitter is the transmit (serial) shift register (TSR). The shift register obtains its data from the read/write transmit buffer register TXREG. The TXREG register is loaded with data in software. The TSR register is not loaded until the last bit has been transmitted from the previous load. As soon as the last bit is transmitted, the TSR is loaded with new data from the TXREG (if available). Once the TXREG register transfers the data to the TSR register (occurs in one Tcycle), the TXREG is empty and the TXIF interrupt flag bit is set. The interrupt can be enabled/disabled by setting/clearing enable the TXIE bit. The TXIF flag bit will be set regardless of the state of the TXIE enable bit and cannot be cleared in software. It will reset only when new data is loaded into the TXREG register. While the TXIF flag bit indicates the status of the TXREG register, the TRMT bit (TXSTA<1>) shows the status of the TSR register. The TRMT bit is a read only bit which is set when the TSR is empty. No interrupt logic is tied to this bit, so the user has to poll this bit in order to determine if the TSR register is empty. The TSR is not mapped in data memory so it is not available to the user.

Transmission is enabled by setting the TXEN bit (TXSTA<5>). The actual transmission will not occur until the TXREG register has been loaded with data. The first data bit will be shifted out on the next available rising edge of the clock on the CK line. Data out is stable at the falling edge of the synchronous clock (Figure 18-10). The transmission can also be started by first loading the TXREG register and then setting the TXEN bit. This is advantageous when slow baud rates are selected, since the BRG is kept in reset when the TXEN, CREN, and SREN bits are clear. Setting the TXEN bit will start the BRG, creating a shift clock immediately. Normally when transmission is first started, the TSR register is empty, so a transfer to the TXREG register will result in an immediate transfer to TSR resulting in an empty TXREG. Back-to-back transfers are possible.

Clearing the TXEN bit during a transmission will cause the transmission to be aborted and will reset the transmitter. The DT and CK pins will revert to hi-impedance. If either of the CREN or SREN bits are set during a transmission, the transmission is aborted and the DT pin reverts to a hi-impedance state (for a reception). The CK pin will remain an output if the CSRC bit is set (internal clock). The transmitter logic is not reset although it is disconnected from the pins. In order to reset the transmitter, the user has to clear the TXEN bit. If the SREN bit is set (to interrupt an on-going transmission and receive a single word), then after the single word is received, the SREN bit will be cleared and the serial port will revert back to transmitting since the TXEN bit is still set. The DT line will immediately switch from hi-impedance receive mode to transmit and start driving. To avoid this the TXEN bit should be cleared.

In order to select 9-bit transmission, the TX9 bit (TXSTA<6>) should be set and the ninth bit should be written to the TX9D bit (TXSTA<0>). The ninth bit must be written before writing the 8-bit data to the TXREG register. This is because a data write to the TXREG can result in an immediate transfer of the data to the TSR register (if the TSR is empty). If the TSR was empty and the TXREG was written before writing the “new” value to the TX9D bit, the “present” value of of the TX9D bit is loaded.

Steps to follow when setting up a Synchronous Master Transmission:

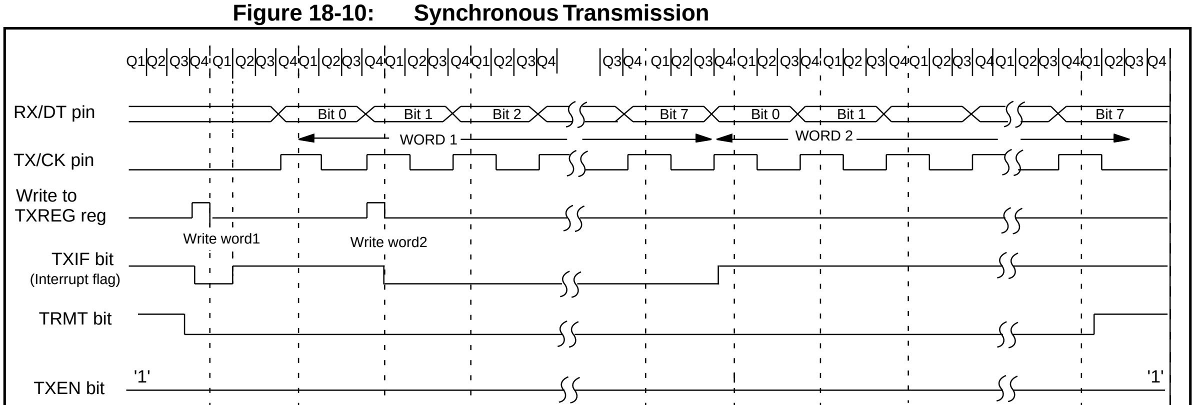
1. Initialize the SPBRG register for the appropriate baud rate (Subsection **18.3 "USART Baud Rate Generator (BRG)"**).
2. Enable the synchronous master serial port by setting the SYNC, SPEN, and CSRC bits.
3. If interrupts are desired, then set the TXIE bit.
4. If 9-bit transmission is desired, then set the TX9 bit.
5. Enable the transmission by setting the TXEN bit.
6. If 9-bit transmission is selected, the ninth bit should be loaded in the TX9D bit.
7. Start transmission by loading data to the TXREG register.

Table 18-8: Registers Associated with Synchronous Master Transmission

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets
PIR	TXIF (1)								0	0
RCSTA	SPEN	RX9	SREN	CREN	—	FERR	OERR	RX9D	0000 -00x	0000 -00x
TXREG	TX7	TX6	TX5	TX4	TX3	TX2	TX1	TX0	0000 0000	0000 0000
PIE	TXIE (1)								0	0
TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

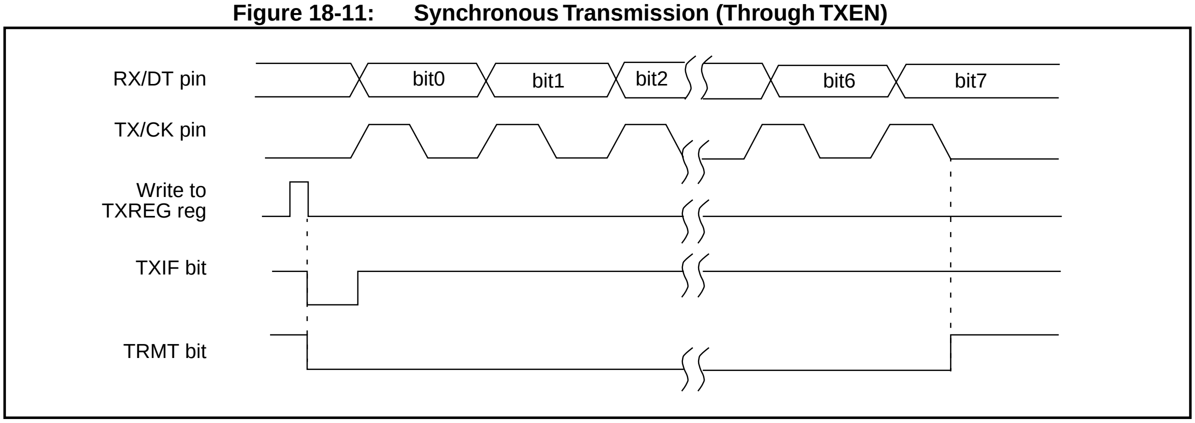
Legend: x = unknown, - = unimplemented, read as '0'.
Shaded cells are not used for Synchronous Master Transmission.
Note 1: The position of this bit is device dependent.

Figure 18-10: Synchronous Transmission



Note: Sync master mode; SPBRG = '0'. Continuous transmission of two 8-bit words.

Figure 18-11: Synchronous Transmission (Through TXEN)



18.5.2 USART Synchronous Master Reception

Once Synchronous mode is selected, reception is enabled by setting either of the SREN (RCSTA<5>) or CREN (RCSTA<4>) bits. Data is sampled on the RX/DT pin on the falling edge of the clock. If the SREN bit is set, then only a single word is received. If the CREN bit is set, the reception is continuous until the CREN bit is cleared. If both bits are set then the CREN bit takes precedence. After clocking the last serial data bit, the received data in the Receive Shift Register (RSR) is transferred to the RCREG register (if it is empty). When the transfer is complete, the RCIF interrupt flag bit is set. The actual interrupt can be enabled/disabled by setting/clearing the RCIE enable bit. The RCIF flag bit is a read only bit which is cleared by the hardware. In this case it is cleared when the RCREG register has been read and is empty. The RCREG is a double buffered register, i.e. it is a two deep FIFO. It is possible for two bytes of data to be received and transferred to the RCREG FIFO and a third byte to begin shifting into the RSR register. On the clocking of the last bit of the third byte, if the RCREG register is still full then overrun error bit, OERR (RCSTA<1>), is set and the word in the RSR is lost. The RCREG register can be read twice to retrieve the two bytes in the FIFO. The OERR bit has to be cleared in software (by clearing the CREN bit). If the OERR bit is set, transfers from the RSR to the RCREG are inhibited, so it is essential to clear the OERR bit if it is set. The 9th receive bit is buffered the same way as the receive data. Reading the RCREG register will load the RX9D bit with a new value, therefore it is essential for the user to read the RCSTA register before reading RCREG in order not to lose the old (previous) information in the RX9D bit.

Steps to follow when setting up a Synchronous Master Reception:

1. Initialize the SPBRG register for the appropriate baud rate. (Subsection **18.3 “USART Baud Rate Generator (BRG)”**)
2. Enable the synchronous master serial port by setting the SYNC, SPEN, and CSRC bits.
3. Ensure that the CREN and SREN bits are clear.
4. If interrupts are desired, then set the RCIE bit.
5. If 9-bit reception is desired, then set the RX9 bit.
6. If a single reception is required, set the SREN bit. For continuous reception set the CREN bit.
7. The RCIF bit will be set when reception is complete and an interrupt will be generated if the RCIE bit was set.
8. Read the RCSTA register to get the ninth bit (if enabled) and determine if any error occurred during reception.
9. Read the 8-bit received data by reading the RCREG register.
10. If any error occurred, clear the error by clearing the CREN bit.

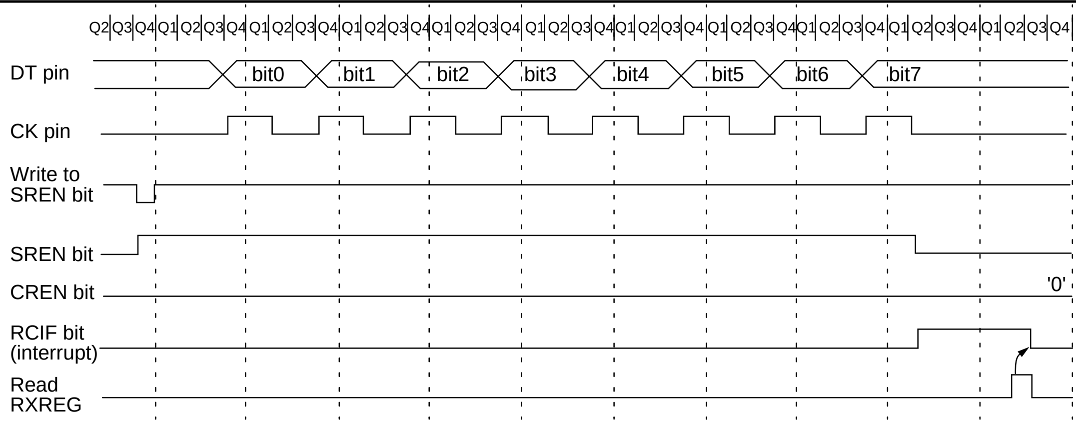
Table 18-9: Registers Associated with Synchronous Master Reception

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets
PIR	RCIF (1)								0	0
RCSTA	SPEN	RX9	SREN	CREN	—	FERR	OERR	RX9D	0000 -00x	0000 -00x
RCREG	RX7	RX6	RX5	RX4	RX3	RX2	RX1	RX0	0000 0000	0000 0000
PIE	RCIE (1)								0	0
TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, - = unimplemented read as '0'.
Shaded cells are not used for Synchronous Master Reception.

Note 1: The position of this bit is device dependent.

Figure 18-12: Synchronous Reception (Master Mode, SREN)



Note: Timing diagram demonstrates SYNC master mode with SREN = '1' and BRG = '0'.

18.6 USART Synchronous Slave Mode

Synchronous slave mode differs from the Master mode in the fact that the shift clock is supplied externally at the TX/CK pin (instead of being supplied internally in master mode). This allows the device to transfer or receive data while in SLEEP mode. Slave mode is entered by clearing the CSRC bit (TXSTA<7>).

18.6.1 USART Synchronous Slave Transmit

The operation of the synchronous master and slave modes are identical except in the case of the SLEEP mode.

If two words are written to the TXREG and then the SLEEP instruction is executed, the following will occur:

a) The first word will immediately transfer to the TSR register and transmit.
b) The second word will remain in TXREG register.
c) The TXIF flag bit will not be set.
d) When the first word has been shifted out of TSR, the TXREG register will transfer the second word to the TSR and the TXIF flag bit will now be set.
e) If the TXIE enable bit is set, the interrupt will wake the chip from SLEEP and if the global interrupt is enabled, the program will branch to the interrupt vector (0004h).

Steps to follow when setting up a Synchronous Slave Transmission:

1. Enable the synchronous slave serial port by setting the SYNC and SPEN bits and clearing the CSRC bit.
2. Clear the CREN and SREN bits.
3. If interrupts are desired, then set the TXIE enable bit.
4. If 9-bit transmission is desired, then set the TX9 bit.
5. Enable the transmission by setting the TXEN enable bit.
6. If 9-bit transmission is selected, the ninth bit should be loaded into the TX9D bit.
7. Start transmission by loading data to the TXREG register.

Table 18-10: Registers Associated with Synchronous Slave Transmission

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets
PIR	TXIF (1)								0	0
RCSTA	SPEN	RX9	SREN	CREN	—	FERR	OERR	RX9D	0000 -00x	0000 -00x
TXREG	TX7	TX6	TX5	TX4	TX3	TX2	TX1	TX0	0000 0000	0000 0000
PIE	TXIE (1)								0	0
TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, - = unimplemented read as '0'.
Shaded cells are not used for Synchronous Slave Transmission.
Note 1: The position of this bit is device dependent.

18.6.2 USART Synchronous Slave Reception

The operation of the synchronous master and slave modes is identical except in the case of the SLEEP mode. Also, bit SREN is a don't care in slave mode.

If receive is enabled, by setting the CREN bit, prior to the SLEEP instruction, then a word may be received during SLEEP. On completely receiving the word, the RSR register will transfer the data to the RCREG register and if the RCIE enable bit bit is set, the interrupt generated will wake the chip from SLEEP. If the global interrupt is enabled, the program will branch to the interrupt vector (0004h).

Steps to follow when setting up a Synchronous Slave Reception:

1. Enable the synchronous master serial port by setting the SYNC and SPEN bits and clearing the CSRC bit.
2. If interrupts are desired, then set the RCIE enable bit.
3. If 9-bit reception is desired, then set the RX9 bit.
4. To enable reception, set the CREN enable bit.
5. The RCIF bit will be set when reception is complete and an interrupt will be generated, if the RCIE bit was set.
6. Read the RCSTA register to get the ninth bit (if enabled) and determine if any error occurred during reception.
7. Read the 8-bit received data by reading the RCREG register.
8. If any error occurred, clear the error by clearing the CREN bit.

Table 18-11: Registers Associated with Synchronous Slave Reception

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets
PIR	RCIF (1)								0	0
RCSTA	SPEN	RX9	SREN	CREN	—	FERR	OERR	RX9D	0000 -00x	0000 -00x
RCREG	RX7	RX6	RX5	RX4	RX3	RX2	RX1	RX0	0000 0000	0000 0000
PIE	RCIE (1)								0	0
TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, - = unimplemented read as '0'.
Shaded cells are not used for Synchronous Slave Reception.

Note 1: The position of this bit is device dependent.

18.7 Initialization

Example 18-2 is an initialization routine for asynchronous Transmitter/Receiver mode. Example 18-3 is for the synchronous mode. In both examples the data is 8-bits, and the value to load into the SPBRG register is dependent on the desired baud rate and the device frequency.

Example 18-2: Asynchronous Transmitter/Receiver

```
    BSF    STATUS,RP0     ; Go to Bank1
    MOVLW  <baudrate>     ; Set Baud rate
    MOVWF  SPBRG
    MOVLW  0x40           ; 8-bit transmit, transmitter enabled,
    MOVWF  TXSTA          ; asynchronous mode, low speed mode
    BSF    PIE1,TXIE      ; Enable transmit interrupts
    BSF    PIE1,RCIE      ; Enable receive interrupts
    BCF    STATUS,RP0     ; Go to Bank 0
    MOVLW  0x90           ; 8-bit receive, receiver enabled,
    MOVWF  RCSTA          ; serial port enabled
```

Example 18-3: Synchronous Transmitter/Receiver

```
    BSF    STATUS,RP0      ; Go to Bank 1
    MOVLW  <baudrate>      ; Set Baud Rate
    MOVWF  SPBRG
    MOVLW  0xB0            ; Synchronous Master,8-bit transmit,
    MOVWF  TXSTA           ; transmitter enabled, low speed mode
    BSF    PIE1,TXIE       ; Enable transmit interrupts
    BSF    PIE1,RCIE       ; Enable receive interrupts
    BCF    STATUS,RP0      ; Go to Bank 0
    MOVLW  0x90            ; 8-bit receive, receiver enabled,
    MOVWF  RCSTA           ; continuous receive, serial port enabled
```

18.8 Design Tips

Question 1: ***Using the Asynchronous mode I am getting a lot of transmission errors.***

Answer 1:

The most common reasons are

1. You are using the high speed mode (BRGH is set) on one of the devices which has an errata for this mode (PIC16C65/65A/73/73A/74/74A).
2. You have incorrectly calculated the value to load in to the SPBRG register
3. The sum of the baud errors for the transmitter and receiver is too high.

18.9 Related Application Notes

This section lists application notes that are related to this section of the manual. These application notes may not be written specifically for the Mid-Range MCU family (that is they may be written for the Base-Line, or High-End families), but the concepts are pertinent, and could be used (with modification and possible limitations). The current application notes related to this section are:

Title	Application Note #
Serial Port Utilities	AN547
Servo Control of a DC Brushless Motor	AN543

18.10 Revision History

Revision A

This is the initial released revision of the USART module description.