Section 19. UART

HIGHLIGHTS

This section of the manual contains the following major topics:

19.1 Introduction 19-2
19.2 Control Registers 19-3
19.3 UART Baud Rate Generator (BRG) 19-8
19.4 UART Configuration 19-10
19.5 UART Transmitter 19-11
19.6 UART Receiver 19-14
19.7 Using the UART for 9-bit Communication 19-18
19.8 Receiving Break Characters 19-19
19.9 Initialization 19-20
19.10 Other Features of the UART 19-21
19.11 UART Operation During CPU Sleep and Idle Modes 19-21
19.12 Registers Associated with UART Module 19-22
19.13 Design Tips 19-23
19.14 Related Application Notes 19-24
19.15 Revision History 19-25

19.1 Introduction

The Universal Asynchronous Receiver Transmitter (UART) module is one of the serial I/O modules available in the dsPIC30F device family. The UART is a full-duplex asynchronous system that can communicate with peripheral devices, such as personal computers, RS-232 and RS-485 interfaces.

The primary features of the UART module are:

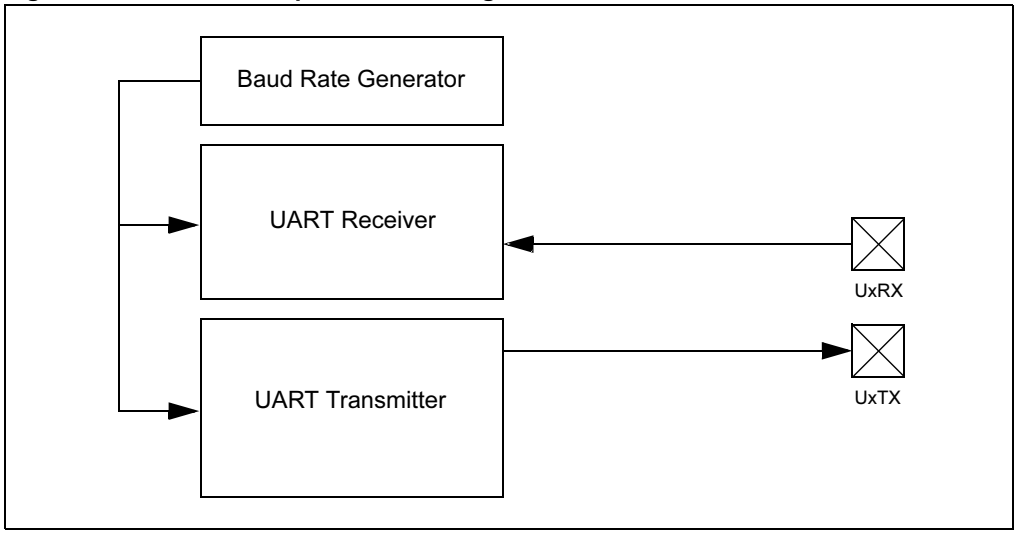
- Full-duplex 8- or 9-bit data transmission through the UxTX and UxRX pins
- Even, Odd or No Parity options (for 8-bit data)
- One or two Stop bits
- Fully integrated Baud Rate Generator with 16-bit prescaler
- Baud rates ranging from 29 bps to 1.875 Mbps at FCY = 30 MHz
- 4-deep First-In-First-Out (FIFO) transmit data buffer
- 4-deep FIFO receive data buffer
- Parity, Framing and Buffer Overrun error detection
- Support for 9-bit mode with Address Detect (9th bit = 1)
- Transmit and Receive Interrupts
- Loopback mode for diagnostic support

Note:	Each dsPIC30F device variant may have one or more UART modules. An 'x' used in the names of pins, control/status bits and registers denotes the particular module. Refer to the specific device data sheets for more details.

A simplified block diagram of the UART is shown in Figure 19-1. The UART module consists of the key important hardware elements:

- Baud Rate Generator
- Asynchronous Transmitter
- Asynchronous Receiver

Figure 19-1: UART Simplified Block Diagram



19.2 Control Registers

Register 19-1: UxMODE: UARTx Mode Register

Upper Byte:							
R/W-0	U-0	R/W-0	U-0	U-0	R/W-0	U-0	U-0
UARTEN	—	USIDL	—	reserved	ALTIO	reserved	reserved
bit 15							bit 8

Lower Byte:							
R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0
WAKE	LPBACK	ABAUD	—	—	PDSEL<1:0>		STSEL
bit 7							bit 0

bit 15 **UARTEN:** UART Enable bit
1 = UART is enabled. UART pins are controlled by UART as defined by UEN<1:0> and UTXEN control bits.
0 = UART is disabled. UART pins are controlled by corresponding PORT, LAT, and TRIS bits.

bit 14 **Unimplemented:** Read as '0'

bit 13 **USIDL:** Stop in Idle Mode bit
1 = Discontinue operation when device enters Idle mode
0 = Continue operation in Idle mode

bit 12 **Unimplemented:** Read as '0'

bit 11 **Reserved:** Write '0' to this location

bit 10 **ALTIO:** UART Alternate I/O Selection bit
1 = UART communicates using UxATX and UxARX I/O pins
0 = UART communicates using UxTX and UxRX I/O pins

Note: The alternate UART I/O pins are not available on all devices. See device data sheet for details.

bit 9-8 **Reserved:** Write '0' to these locations

bit 7 **WAKE:** Enable Wake-up on Start bit Detect During Sleep Mode bit
1 = Wake-up enabled
0 = Wake-up disabled

bit 6 **LPBACK:** UART Loopback Mode Select bit
1 = Enable Loopback mode
0 = Loopback mode is disabled

bit 5 **ABAUD:** Auto Baud Enable bit
1 = Input to Capture module from UxRX pin
0 = Input to Capture module from ICx pin

bit 4-3 **Unimplemented:** Read as '0'

bit 2-1 **PDSEL<1:0>:** Parity and Data Selection bits
11 = 9-bit data, no parity
10 = 8-bit data, odd parity
01 = 8-bit data, even parity
00 = 8-bit data, no parity

bit 0 **STSEL:** Stop Selection bit
1 = 2 Stop bits
0 = 1 Stop bit

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

Register 19-2: UxSTA: UARTx Status and Control Register

Upper Byte:							
R/W-0	U-0	U-0	U-0	R/W-0	R/W-0	R-0	R-1
UTXISEL	—	—	—	UTXBRK	UTXEN	UTXBF	TRMT
bit 15							bit 8

Lower Byte:							
R/W-0	R/W-0	R/W-0	R-1	R-0	R-0	R/C-0	R-0
URXISEL<1:0>		ADDEN	RIDLE	PERR	FERR	OERR	URXDA
bit 7							bit 0

bit 15 **UTXISEL:** Transmission Interrupt Mode Selection bit
1 = Interrupt when a character is transferred to the Transmit Shift register and as result, the transmit buffer becomes empty
0 = Interrupt when a character is transferred to the Transmit Shift register (this implies that there is at least one character open in the transmit buffer)

bit 14-12 **Unimplemented:** Read as '0'

bit 11 **UTXBRK:** Transmit Break bit
1 = UxTX pin is driven low, regardless of transmitter state
0 = UxTX pin operates normally

bit 10 **UTXEN:** Transmit Enable bit
1 = UART transmitter enabled, UxTX pin controlled by UART (if UARTEN = 1)
0 = UART transmitter disabled, any pending transmission is aborted and buffer is reset. UxTX pin controlled by PORT.

bit 9 **UTXBF:** Transmit Buffer Full Status bit (Read Only)
1 = Transmit buffer is full
0 = Transmit buffer is not full, at least one more data word can be written

bit 8 **TRMT:** Transmit Shift Register is Empty bit (Read Only)
1 = Transmit shift register is empty and transmit buffer is empty (the last transmission has completed)
0 = Transmit shift register is not empty, a transmission is in progress or queued in the transmit buffer

bit 7-6 **URXISEL<1:0>:** Receive Interrupt Mode Selection bit
11 = Interrupt flag bit is set when Receive Buffer is full (i.e., has 4 data characters)
10 = Interrupt flag bit is set when Receive Buffer is 3/4 full (i.e., has 3 data characters)
0x = Interrupt flag bit is set when a character is received

bit 5 **ADDEN:** Address Character Detect (bit 8 of received data = 1)
1 = Address Detect mode enabled. If 9-bit mode is not selected, this control bit has no effect.
0 = Address Detect mode disabled

bit 4 **RIDLE:** Receiver Idle bit (Read Only)
1 = Receiver is Idle
0 = Data is being received

bit 3 **PERR:** Parity Error Status bit (Read Only)
1 = Parity error has been detected for the current character
0 = Parity error has not been detected

bit 2 **FERR:** Framing Error Status bit (Read Only)
1 = Framing Error has been detected for the current character
0 = Framing Error has not been detected

Register 19-2: UxSTA: UARTx Status and Control Register (Continued)

bit 1 **OERR:** Receive Buffer Overrun Error Status bit (Read/Clear Only)
1 = Receive buffer has overflowed
0 = Receive buffer has not overflowed

bit 0 **URXDA:** Receive Buffer Data Available bit (Read Only)
1 = Receive buffer has data, at least one more character can be read
0 = Receive buffer is empty

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	C = Bit can be cleared
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

Register 19-3: UxRXREG: UARTx Receive Register

Upper Byte:							
U-0	U-0	U-0	U-0	U-0	U-0	U-0	R-0
—	—	—	—	—	—	—	URX8
bit 15							bit 8

Lower Byte:							
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
URX<7:0>							
bit 7							bit 0

bit 15-9 **Unimplemented:** Read as '0'

bit 8 **URX8:** Data bit 8 of the Received Character (in 9-bit mode)

bit 7-0 **URX<7:0>:** Data bits 7-0 of the Received Character

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

Register 19-4: UxTXREG: UARTx Transmit Register (Write Only)

Upper Byte:							
U-0	U-0	U-0	U-0	U-0	U-0	U-0	W-x
—	—	—	—	—	—	—	UTX8
bit 15							bit 8

Lower Byte:							
W-x	W-x	W-x	W-x	W-x	W-x	W-x	W-x
UTX<7:0>							
bit 7							bit 0

bit 15-9 **Unimplemented:** Read as '0'

bit 8 **UTX8:** Data bit 8 of the Character to be Transmitted (in 9-bit mode)

bit 7-0 **UTX<7:0>:** Data bits 7-0 of the Character to be Transmitted

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

Register 19-5: UxBRG: UARTx Baud Rate Register

Upper Byte:							
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
BRG<15:8>							
bit 15							bit 8

Lower Byte:							
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
BRG<7:0>							
bit 7							bit 0

bit 15-0 **BRG<15:0>:** Baud Rate Divisor bits

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

19.3 UART Baud Rate Generator (BRG)

The UART module includes a dedicated 16-bit baud rate generator. The UxBRG register controls the period of a free running 16-bit timer. Equation 19-1 shows the formula for computation of the baud rate.

Equation 19-1: UART Baud Rate

$$Baud\ Rate = \frac{FCY}{16 \bullet (UxBRG + 1)}$$

$$UxBRG = \frac{FCY}{16 \bullet Baud\ Rate} - 1$$

Note: FCY denotes the instruction cycle clock frequency.

Example 19-1 shows the calculation of the baud rate error for the following conditions:

- FCY = 4 MHz
- Desired Baud Rate = 9600

Example 19-1: Baud Rate Error Calculation

Desired Baud Rate = FCY/(16 (UxBRG + 1))

Solving for UxBRG value:

UxBRG = ((FCY/Desired Baud Rate)/16) – 1
UxBRG = ((4000000/9600)/16) – 1
UxBRG = [25.042] = 25

Calculated Baud Rate = 4000000/(16 (25 + 1))
= 9615

$$\text{Error} = \frac{\text{(Calculated Baud Rate – Desired Baud Rate)}}{\text{Desired Baud Rate}}$$
$$= (9615 - 9600)/9600$$
$$= 0.16\%$$

The maximum baud rate possible is FCY / 16 (for UxBRG = 0), and the minimum baud rate possible is FCY / (16 * 65536).

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

19.3.1 Baud Rate Tables

UART baud rates are provided in Table 19-1 for common device instruction cycle frequencies (FCY). The minimum and maximum baud rates for each frequency are also shown.

Table 19-1: UART Baud Rates

BAUD RATE (Kbps)	FCY = 30 MHz		BRG value (decimal)	25 MHz		BRG value (decimal)	20 MHz		BRG value (decimal)	16 MHz		BRG value (decimal)
	KBAUD	% ERROR		KBAUD	% ERROR		KBAUD	% ERROR		KBAUD	% ERROR	
0.3	0.3	0.0	6249	0.3	+0.01	5207	0.3	0.0	4166	0.3	+0.01	3332
1.2	1.1996	0.0	1562	1.2001	+0.01	1301	1.1996	0.0	1041	1.2005	+0.04	832
2.4	2.4008	0.0	780	2.4002	+0.01	650	2.3992	0.0	520	2.3981	-0.08	416
9.6	9.6154	+0.2	194	9.5859	-0.15	162	9.6154	+0.2	129	9.6154	+0.16	103
19.2	19.1327	-0.4	97	19.2901	0.47	80	19.2308	+0.2	64	19.2308	+0.16	51
38.4	38.2653	-0.4	48	38.1098	-0.76	40	37.8788	-1.4	32	38.4615	+0.16	25
56	56.8182	+1.5	32	55.8036	-0.35	27	56.8182	+1.5	21	55.5556	-0.79	17
115	117.1875	+1.9	15	111.6071	-2.95	13	113.6364	-1.2	10	111.1111	-3.38	8
250							250	0.0	4	250	0.0	3
500										500	0.0	1
MIN.	0.0286	0.0	65535	0.0238	0.0	65535	0.019	0.0	65535	0.015	0.0	65535
MAX.	1875	0.0	0	1562.5	0.0	0	1250	0.0	0	1000	0.0	0

BAUD RATE (Kbps)	FCY = 12 MHz		BRG value (decimal)	10 MHz		BRG value (decimal)	8 MHz		BRG value (decimal)	7.68 MHz		BRG value (decimal)
	KBAUD	% ERROR		KBAUD	% ERROR		KBAUD	% ERROR		KBAUD	% ERROR	
0.3	0.3	0.0	2499	0.3	0.0	2082	0.2999	-0.02	1666	0.3	0.0	1599
1.2	1.2	0.0	624	1.1996	0.0	520	1.199	-0.08	416	1.2	0.0	399
2.4	2.3962	-0.2	312	2.4038	+0.2	259	2.4038	+0.16	207	2.4	0.0	199
9.6	9.6154	-0.2	77	9.6154	+0.2	64	9.6154	+0.16	51	9.6	0.0	49
19.2	19.2308	+0.2	38	18.9394	-1.4	32	19.2308	+0.16	25	19.2	0.0	24
38.4	37.5	+0.2	19	39.0625	+1.7	15	38.4615	+0.16	12			
56	57.6923	-2.3	12	56.8182	+1.5	10	55.5556	-0.79	8			
115			6									
250	250	0.0	2				250	0.0	1			
500							500	0.0	0			
MIN.	0.011	0.0	65535	0.010	0.0	65535	0.008	0.0	65535	0.007	0.0	65535
MAX.	750	0.0	0	625	0.0	0	500	0.0	0	480	0.0	0

BAUD RATE (Kbps)	FCY = 5 MHz		BRG value (decimal)	4 MHz		BRG value (decimal)	3.072 MHz		BRG value (decimal)	1.8432 MHz		BRG value (decimal)
	KBAUD	% ERROR		KBAUD	% ERROR		KBAUD	% ERROR		KBAUD	% ERROR	
0.3	0.2999	0.0	1041	0.3001	0.0	832	0.3	0.0	639	0.3	0.0	383
1.2	1.2019	+0.2	259	1.2019	+0.2	207	1.2	0.0	159	1.2	0.0	95
2.4	2.4038	+0.2	129	2.4038	+0.2	103	2.4	0.0	79	2.4	0.0	47
9.6	9.4697	-1.4	32	9.6154	+0.2	25	9.6	0.0	19	9.6	0.0	11
19.2	19.5313	+1.7	15	19.2308	+0.2	12	19.2	0.0	9	19.2	0.0	5
38.4	39.0625	+1.7	7				38.4	0.0	4	38.4	0.0	2
56												
115												
250												
500												
MIN.	0.005	0.0	65535	0.004	0.0	65535	0.003	0.0	65535	0.002	0.0	65535
MAX.	312.5	0.0	0	250	0.0	0	192	0.0	0	115.2	0.0	0

19.4 UART Configuration

The UART uses standard non-return-to-zero (NRZ) format (one Start bit, eight or nine data bits, and one or two Stop bits). Parity is supported by the hardware, and may be configured by the user as even, odd or no parity. The most common data format is 8 bits, no parity and one Stop bit (denoted as 8, N, 1), which is the default (POR) setting. The number of data bits and Stop bits, and the parity, are specified in the PDSEL<1:0> (UxMODE<2:1>) and STSEL (UxMODE<0>) bits. An on-chip dedicated 16-bit baud rate generator can be used to derive standard baud rate frequencies from the oscillator. The UART transmits and receives the LSb first. The UART's transmitter and receiver are functionally independent, but use the same data format and baud rate.

19.4.1 Enabling the UART

The UART module is enabled by setting the UARTEN (UxMODE<15>) bit and UTXEN (UxSTA<10>) bit. Once enabled, the UxTX and UxRX pins are configured as an output and an input, respectively, overriding the TRIS and PORT register bit settings for the corresponding I/O port pins. The UxTX pin is at logic '1' when no transmission is taking place.

Note:	The UTXEN bit should not be set until the UARTEN bit has been set. Otherwise, UART transmissions will not be enabled.

19.4.2 Disabling the UART

The UART module is disabled by clearing the UARTEN (UxMODE<15>) bit. This is the default state after any Reset. If the UART is disabled, all UART pins operate as port pins under the control of their corresponding PORT and TRIS bits.

Disabling the UART module resets the buffers to empty states. Any data characters in the buffers are lost, and the baud rate counter is reset.

All error and status flags associated with the UART module are reset when the module is disabled. The URXDA, OERR, FERR, PERR, UTXEN, UTXBRK and UTXBF bits are cleared, whereas RIDLE and TRMT are set. Other control bits, including ADDEN, URXISEL<1:0>, UTXISEL, as well as the UxMODE and UxBRG registers, are not affected.

Clearing the UARTEN bit while the UART is active will abort all pending transmissions and receptions and reset the module as defined above. Re-enabling the UART will restart the UART in the same configuration.

19.4.3 Alternate UART I/O Pins

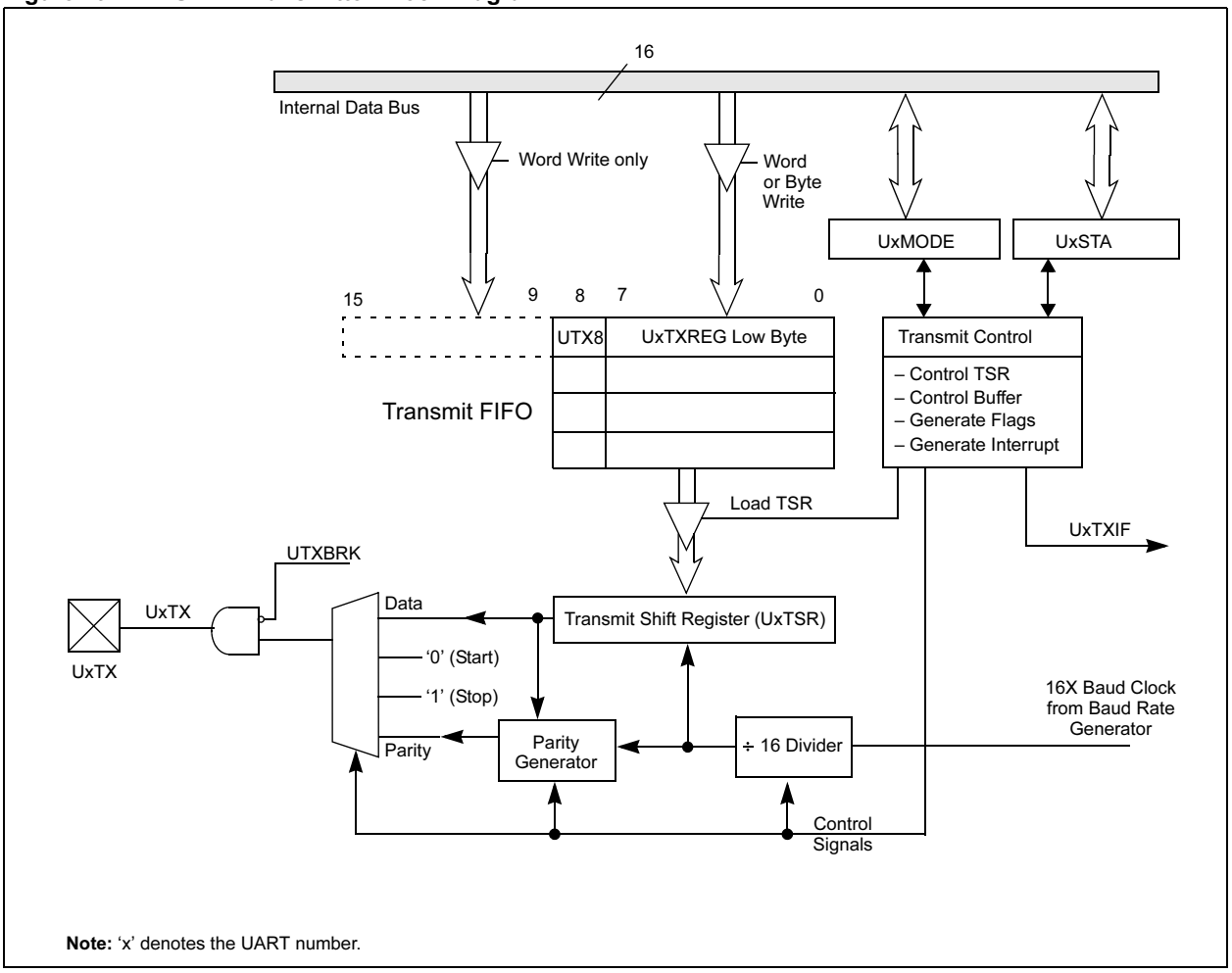
Some dsPIC30F devices have an alternate set of UART transmit and receive pins that can be used for communications. The alternate UART pins are useful when the primary UART pins are shared by other peripherals. The alternate I/O pins are enabled by setting the ALTIO bit (UxMODE<10>). If ALTIO = 1, the UxATX and UxARX pins (alternate transmit and alternate receive pins, respectively) are used by the UART module, instead of the UxTX and UxRX pins. If ALTIO = 0, the UxTX and UxRX pins are used by the UART module.

19.5 UART Transmitter

The UART transmitter block diagram is shown in Figure 19-2. The heart of the transmitter is the Transmit Shift register (UxTSR). The Shift register obtains its data from the transmit FIFO buffer, UxTXREG. The UxTXREG register is loaded with data in software. The UxTSR register is not loaded until the Stop bit has been transmitted from the previous load. As soon as the Stop bit is transmitted, the UxTSR is loaded with new data from the UxTXREG register (if available).

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

Figure 19-2: UART Transmitter Block Diagram



Note: 'x' denotes the UART number.

Transmission is enabled by setting the UTXEN enable bit (UxSTA<10>). The actual transmission will not occur until the UxTXREG register has been loaded with data and the Baud Rate Generator (UxBRG) has produced a shift clock (Figure 19-2). The transmission can also be started by first loading the UxTXREG register and then setting the UTXEN enable bit. Normally when transmission is first started, the UxTSR register is empty, so a transfer to the UxTXREG register will result in an immediate transfer to UxTSR. Clearing the UTXEN bit during a transmission will cause the transmission to be aborted and will reset the transmitter. As a result, the UxTX pin will revert to a high-impedance state.

In order to select 9-bit transmission, the PDSEL<1:0> bits (UxMODE<2:1>) should be set to '11' and the ninth bit should be written to the UTX9 bit (UxTXREG<8>). A word write should be performed to UxTXREG so that all nine bits are written at the same time.

Note:	There is no parity in the case of 9-bit data transmission.

19.5.1 Transmit Buffer (UxTXB)

Each UART has a 4-deep, 9-bit wide FIFO transmit data buffer. The UxTXREG register provides user access to the next available buffer location. The user may write up to 4 words in the buffer. Once the UxTXREG contents are transferred to the UxTSR register, the current buffer location becomes available for new data to be written and the next buffer location is sourced to the UxTSR register. The UTXBF (UxSTA<9>) status bit is set whenever the buffer is full. If a user attempts to write to a full buffer, the new data will not be accepted into the FIFO.

The FIFO is reset during any device Reset, but is not affected when the device enters a Power Saving mode or wakes up from a Power Saving mode.

19.5.2 Transmit Interrupt

The transmit interrupt flag (UxTXIF) is located in the corresponding interrupt flag status (IFS) register. The UTXISEL control bit (UxSTA<15>) determines when the UART will generate a transmit interrupt.

1. If UTXISEL = 0, an interrupt is generated when a word is transferred from the transmit buffer to the Transmit Shift register (UxTSR). This implies that the transmit buffer has at least one empty word. Since an interrupt is generated after the transfer of each individual word, this mode is useful if interrupts can be handled frequently (i.e., the ISR is completed before the transmission of the next word).
2. If UTXISEL = 1, an interrupt is generated when a word is transferred from the transmit buffer to the Transmit Shift register (UxTSR) and the transmit buffer is empty. Since an interrupt is generated only after all 4 words have been transmitted, this 'Block Transmit' mode is useful if the user's code cannot handle interrupts quickly enough (i.e., the ISR is completed before the transmission of the next word).

The UxTXIF bit will be set when the module is first enabled.

The user should clear the UxTXIF bit in the ISR.

Switching between the two Interrupt modes during operation is possible.

Note:	When the UTXEN bit is set, the UxTXIF flag bit will also be set if UTXISEL = 0, since the transmit buffer is not yet full (can move transmit data to the UxTXREG register).

While the UxTXIF flag bit indicates the status of the UxTXREG register, the TRMT bit (UxSTA<8>) shows the status of the UxTSR register. The TRMT status bit is a read only bit, which is set when the UxTSR 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 UxTSR register is empty.

19.5.3 Setup for UART Transmit

Steps to follow when setting up a transmission:

1. Initialize the UxBRG register for the appropriate baud rate (**Section 19.3 "UART Baud Rate Generator (BRG)"**).
2. Set the number of data bits, number of Stop bits, and parity selection by writing to the PDSEL<1:0> (UxMODE<2:1>) and STSEL (UxMODE<0>) bits.
3. If transmit interrupts are desired, set the UxTXIE control bit in the corresponding Interrupt Enable Control register (IEC). Specify the interrupt priority for the transmit interrupt using the UxTXIP<2:0> control bits in the corresponding Interrupt Priority Control register (IPC). Also, select the Transmit Interrupt mode by writing the UTXISEL (UxSTA<15>) bit.
4. Enable the UART module by setting the UARTEN (UxMODE<15>) bit.
5. Enable the transmission by setting the UTXEN (UxSTA<10>) bit, which will also set the UxTXIF bit. The UxTXIF bit should be cleared in the software routine that services the UART transmit interrupt. The operation of the UxTXIF bit is controlled by the UTXISEL control bit.
6. Load data to the UxTXREG register (starts transmission). If 9-bit transmission has been selected, load a word. If 8-bit transmission is used, load a byte. Data can be loaded into the buffer until the UxTXBF status bit (UxSTA<9>) is set.

Note:	The UTXEN bit should not be set until the UARTEN bit has been set. Otherwise, UART transmissions will not be enabled.

Figure 19-3: Transmission (8-bit or 9-bit Data)

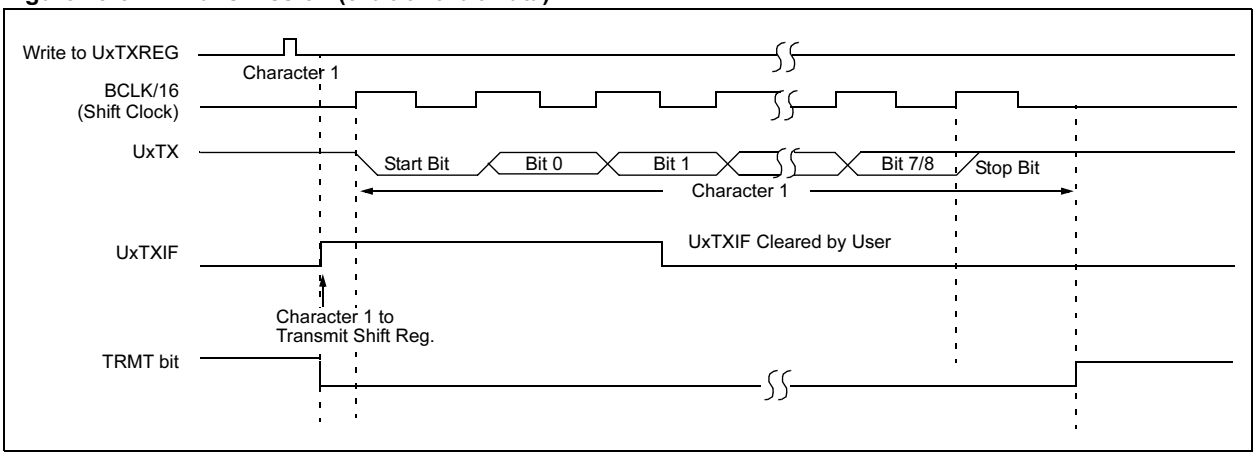
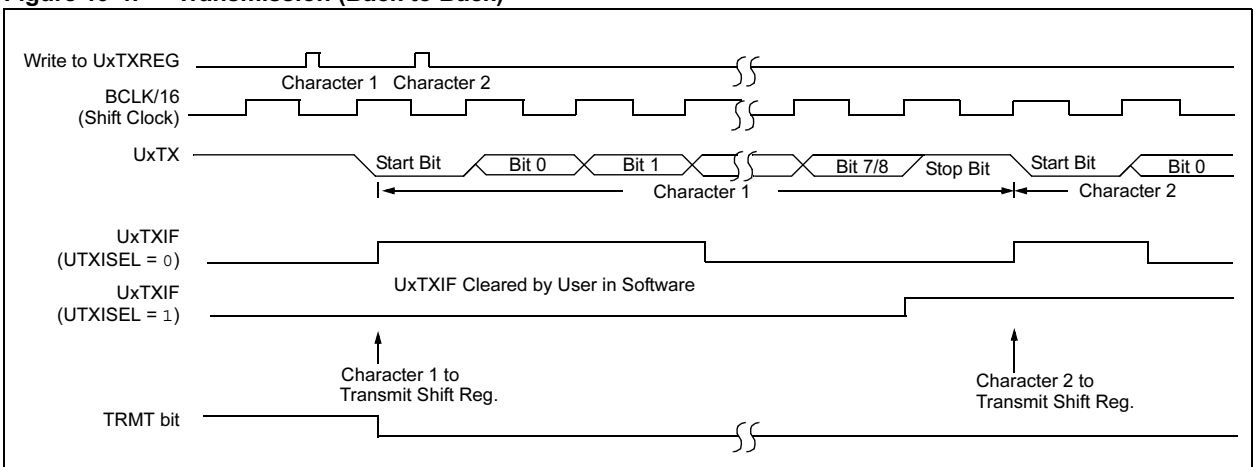


Figure 19-4: Transmission (Back to Back)



Note: This timing diagram shows two consecutive transmissions.

19.5.4 Transmission of Break Characters

Setting the UTXBRK bit (UxSTA<11>) will force the UxTX line to '0'. UTXBRK overrides any other transmitter activity. The user should wait for the transmitter to be Idle (TRMT = 1) before setting UTXBRK.

To send a break character, the UTXBRK bit must be set by software and remain set for a minimum of 13 baud clocks. The baud clock periods are timed in software. The UTXBRK bit is then cleared by software to generate the Stop bit. The user must wait at least one or two baud clocks to ensure a valid Stop bit(s) before loading the UTXBUF again or renewing transmitter activity.

Note:	Sending a break character does not generate a transmitter interrupt.

19.6 UART Receiver

The receiver block diagram is shown in Figure 19-5. The heart of the receiver is the Receive (Serial) Shift register (UxRSR). The data is received on the UxRX pin and is sent to the data recovery block. The data recovery block operates at 16 times the baud rate, whereas the main receive serial shifter operates at the baud rate. After sampling the UxRX pin for the Stop bit, the received data in UxRSR is transferred to the receive FIFO (if it is empty).

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

The data on the UxRX pin is sampled three times by a majority detect circuit to determine if a high or a low level is present at the UxRX pin. Figure 19-5 shows the sampling scheme.

19.6.1 Receive Buffer (UxRXB)

The UART receiver has a 4-deep, 9-bit wide FIFO receive data buffer. UxRXREG is a memory mapped register that provides access to the output of the FIFO. It is possible for 4 words of data to be received and transferred to the FIFO and a fifth word to begin shifting to the UxRSR register before a buffer overrun occurs.

19.6.2 Receiver Error Handling

If the FIFO is full (four characters) and a fifth character is fully received into the UxRSR register, the overrun error bit, OERR (UxSTA<1>), will be set. The word in UxRSR will be kept, but further transfers to the receive FIFO are inhibited as long as the OERR bit is set. The user must clear the OERR bit in software to allow further data to be received.

If it is desired to keep the data received prior to the overrun, the user should first read all five characters, then clear the OERR bit. If the five characters can be discarded, the user can simply clear the OERR bit. This effectively resets the receive FIFO and all prior received data is lost.

Note:	The data in the receive FIFO should be read prior to clearing the OERR bit. The FIFO is reset when OERR is cleared, which causes all data in the buffer to be lost.

The framing error bit, FERR (UxSTA<2>), is set if a Stop bit is detected as a logic low level.

The parity error bit, PERR (UxSTA<3>), is set if a parity error has been detected in the data word at the top of the buffer (i.e., the current word). For example, a parity error would occur if the parity is set to be even, but the total number of ones in the data has been detected to be odd. The PERR bit is irrelevant in the 9-bit mode. The FERR and PERR bits are buffered along with the corresponding word and should be read before reading the data word.

19.6.3 Receive Interrupt

The UART receive interrupt flag (UxRXIF) is located in the corresponding Interrupt Flag Status (IFS) register. The URXISEL<1:0> (UxSTA<7:6>) control bits determine when the UART receiver generates an interrupt.

a) If URXISEL<1:0> = 00 or 01, an interrupt is generated each time a data word is transferred from the Receive Shift register (UxRSR) to the receive buffer. There may be one or more characters in the receive buffer.

b) If URXISEL<1:0> = 10, an interrupt is generated when a word is transferred from the Receive Shift register (UxRSR) to the receive buffer and as a result, the receive buffer contains 3 or 4 characters.

c) If URXISEL<1:0> = 11, an interrupt is generated when a word is transferred from the Receive Shift register (UxRSR) to the receive buffer and as a result, the receive buffer contains 4 characters (i.e., becomes full).

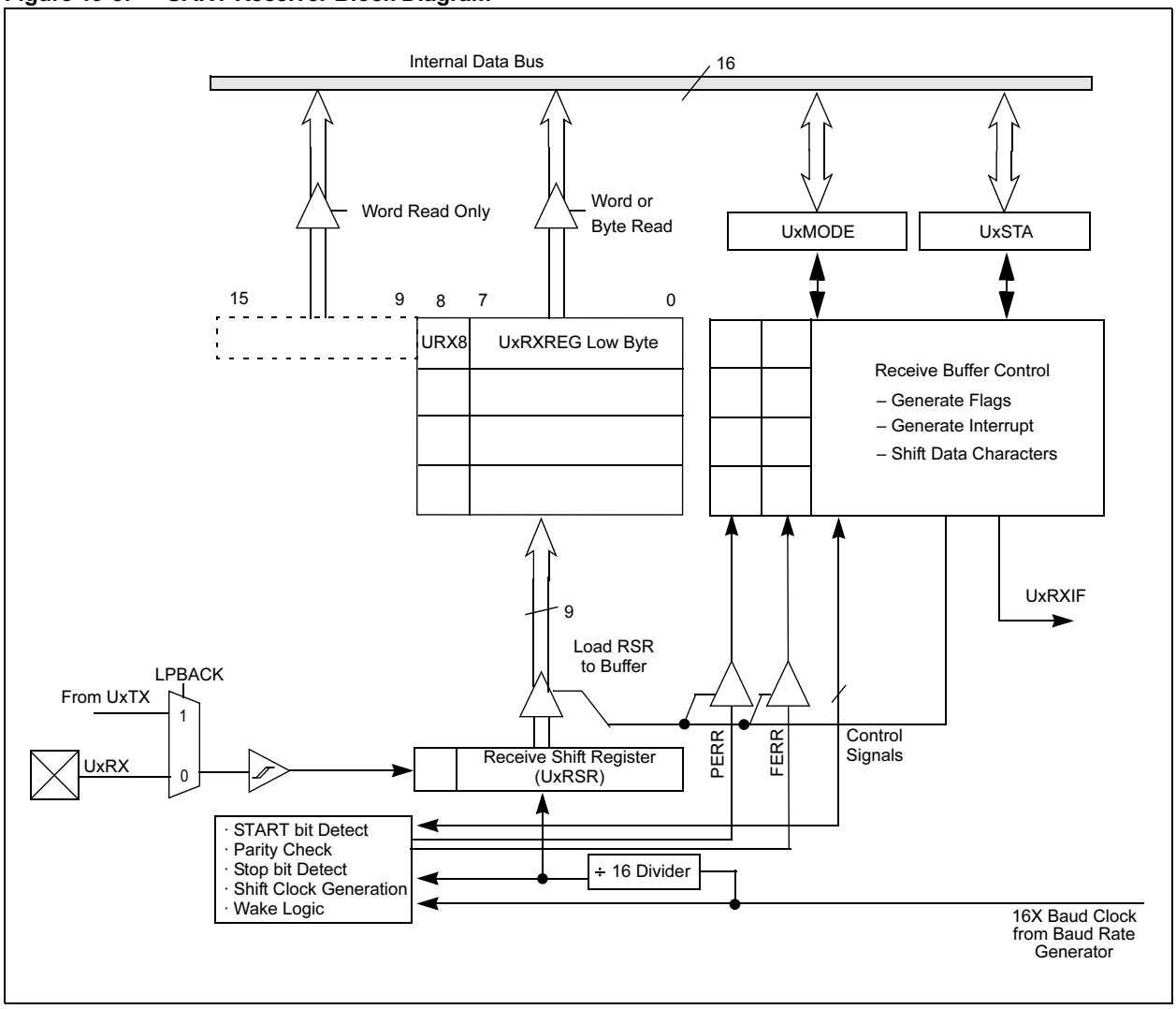
Switching between the three Interrupt modes during operation is possible.

While the URXDA and UxRXIF flag bits indicate the status of the UxRXREG register, the RIDLE bit (UxSTA<4>) shows the status of the UxRSR register. The RIDLE status bit is a read only bit, which is set when the receiver is Idle (i.e., the UxRSR 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 UxRSR is Idle.

The URXDA bit (UxSTA<0>) indicates whether the receive buffer has data or whether the buffer is empty. This bit is set as long as there is at least one character to be read from the receive buffer. URXDA is a read only bit.

Figure 19-5 shows a block diagram of the UART receiver.

Figure 19-5: UART Receiver Block Diagram

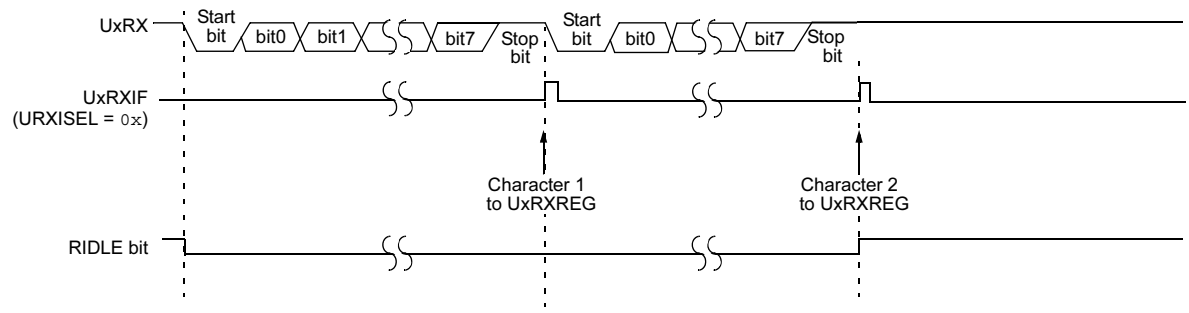


19.6.4 Setup for UART Reception

Steps to follow when setting up a Reception:

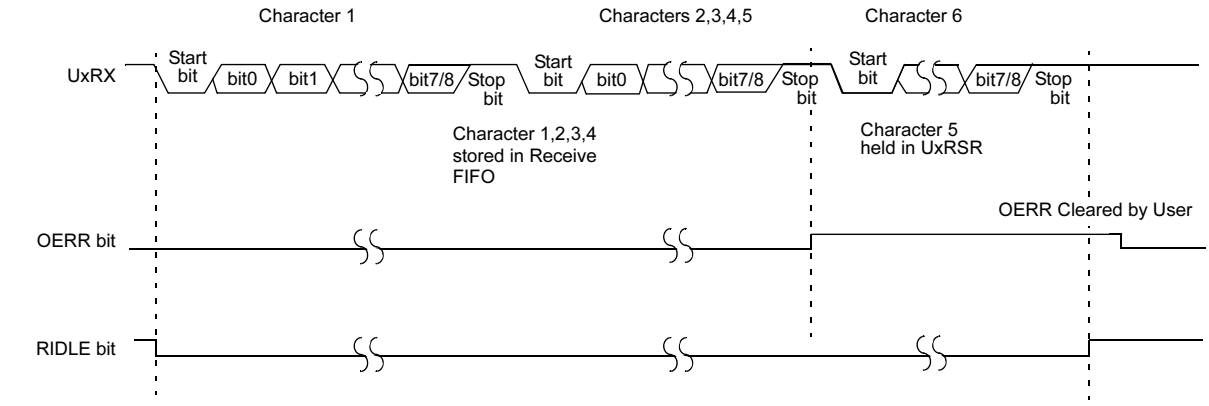
1. Initialize the UxBRG register for the appropriate baud rate (**Section 19.3 "UART Baud Rate Generator (BRG)"**).
2. Set the number of data bits, number of Stop bits and parity selection by writing to the PDSEL<1:0> (UxMODE<2:1>) and STSEL (UxMODE<0>) bits.
3. If interrupts are desired, then set the UxRXIE bit in the corresponding Interrupt Enable Control register (IEC). Specify the interrupt priority for the interrupt using the UxRXIP<2:0> control bits in the corresponding Interrupt Priority Control register (IPC). Also, select the Receive Interrupt mode by writing to the URXISEL<1:0> (UxSTA<7:6>) bits.
4. Enable the UART module by setting the UARTEN (UxMODE<15>) bit.
5. Receive interrupts will depend on the URXISEL<1:0> control bit settings. If receive interrupts are not enabled, the user can poll the URXDA bit. The UxRXIF bit should be cleared in the software routine that services the UART receive interrupt.
6. Read data from the receive buffer. If 9-bit transmission has been selected, read a word. Otherwise, read a byte. The URXDA status bit (UxSTA<0>) will be set whenever data is available in the buffer.

Figure 19-6: UART Reception



Note: This timing diagram shows 2 characters received on the UxRX input.

Figure 19-7: UART Reception with Receive Overrun



Note: This diagram shows 6 characters received without the user reading the input buffer. The 5th character received is held in the receive shift register. An overrun error occurs at the start of the 6th character.

19.7 Using the UART for 9-bit Communication

A typical multi-processor communication protocol will differentiate between data bytes and address/control bytes. A common scheme is to use a 9th data bit to identify whether a data byte is address or data information. If the 9th bit is set, the data is processed as address or control information. If the 9th bit is cleared, the received data word is processed as data associated with the previous address/control byte.

The protocol operates as follows:

- The master device transmits a data word with the 9th bit set. The data word contains the address of a slave device.
- All slave devices in the communication chain receive the address word and check the slave address value.
- The slave device that was addressed will receive and process subsequent data bytes sent by the master device. All other slave devices will discard subsequent data bytes until a new address word (9th bit set) is received.

19.7.1 ADDEN Control Bit

The UART receiver has an Address Detect mode which allows it to ignore data words with the 9th bit cleared. This reduces the interrupt overhead, since data words with the 9th bit cleared are not buffered. This feature is enabled by setting the ADDEN bit (UxSTA<5>).

The UART must be configured for 9-bit data to use the Address Detect mode. The ADDEN bit has no effect when the receiver is configured in 8-bit Data mode.

19.7.2 Setup for 9-bit Transmit

The setup procedure for 9-bit transmission is identical to the 8-bit Transmit modes, except that PDSEL<1:0> (UxMODE<2:1) should be set to '11' (see **Section 19.5.3 "Setup for UART Transmit"**).

Word writes should be performed to the UxTXREG register (starts transmission).

19.7.3 Setup for 9-bit Reception Using Address Detect Mode

The setup procedure for 9-bit reception is similar to the 8-bit Receive modes, except that PDSEL<1:0> (UxMODE<2:1) should be set to '11' (see **Section 19.6.4 "Setup for UART Reception"**).

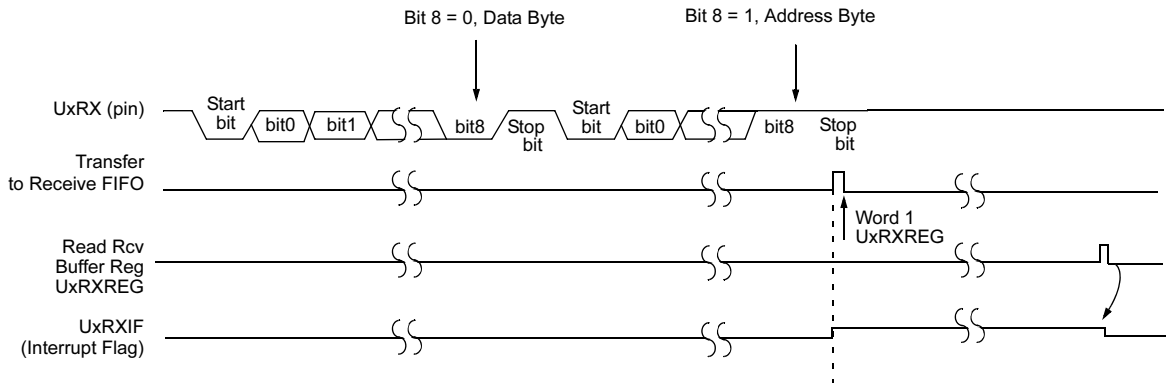
The Receive Interrupt mode should be configured by writing to the URXISEL<1:0> (UxSTA<7:6>) bits.

Note:	If the Address Detect mode is enabled (ADDEN = 1), the URXISEL<1:0> control bits should be configured so that an interrupt will be generated after every received word. Each received data word must be checked in software for an address match immediately after reception.

The procedure for using the Address Detect mode is as follows:

1. Set the ADDEN (UxSTA<5>) bit to enable address detect. Ensure that the URXISEL control bits are configured to generate an interrupt after each received word.
2. Check each 8-bit address by reading the UxRXREG register, to determine if the device is being addressed.
3. If this device has not been addressed, then discard the received word.
4. If this device has been addressed, clear the ADDEN bit to allow subsequent data bytes to be read into the receive buffer and interrupt the CPU. If a long data packet is expected, then the Receive Interrupt mode could be changed to buffer more than one data byte between interrupts.
5. When the last data byte has been received, set the ADDEN bit so that only address bytes will be received. Also, ensure that the URXISEL control bits are configured to generate an interrupt after each received word.

Figure 19-8: Reception with Address Detect (ADDEN = 1)



Note: This timing diagram shows a data byte followed by an address byte. The data byte is not read into the UxRXREG (receive buffer) because ADDEN = 1 and bit 8 = 0.

19.8 Receiving Break Characters

The receiver will count and expect a certain number of bit times based on the values programmed in the PDSEL (UxMODE<2:1>) and STSEL (UxMODE<0>) bits.

If the break is longer than 13 bit times, the reception is considered complete after the number of bit times specified by PDSEL and STSEL. The URXDA bit is set, FERR is set, zeros are loaded into the receive FIFO, interrupts are generated if appropriate and the RIDLE bit is set.

When the module receives a break signal and the receiver has detected the Start bit, the data bits and the invalid Stop bit (which sets the FERR), the receiver must wait for a valid Stop bit before looking for the next Start bit. It cannot assume that the break condition on the line is the next Start bit. Break is regarded as a character containing all '0's with the FERR bit set. The break character is loaded into the buffer. No further reception can occur until a Stop bit is received. Note that RIDLE goes high when the Stop bit has been received.

19.9 Initialization

Example 19-2 is an initialization routine for the Transmitter/Receiver in 8-bit mode. Example 19-3 shows an initialization of the Addressable UART in 9-bit Address Detect mode. In both examples, the value to load into the UxBRG register is dependent on the desired baud rate and the device frequency.

Note:	The UTXEN bit should not be set until the UARTEN bit has been set. Otherwise, UART transmissions will not be enabled.

Example 19-2: 8-bit Transmit/Receive (UART1)

```
MOV    #baudrate,W0    ; Set Baudrate
MOV    W0,U1BRG

BSET   IPC2,#U1TXIP2   ; Set UART TX interrupt priority
BCLR   IPC2,#U1TXIP1   ;
BCLR   IPC2,#U1TXIP0   ;
BSET   IPC2,#U1RXIP2   ; Set UART RX interrupt priority
BCLR   IPC2,#U1RXIP1   ;
BCLR   IPC2,#U1RXIP0   ;

CLR    U1STA

MOV    #0x8800,W0      ; Enable UART for 8-bit data,
                       ; no parity, 1 STOP bit,
                       ; no wakeup
MOV    W0,U1MODE

BSET   U1STA,#UTXEN    ; Enable transmit

BSET   IEC0,#U1TXIE    ; Enable transmit interrupts
BSET   IEC0,#U1RXIE    ; Enable receive interrupts
```

Example 19-3: 8-bit Transmit/Receive (UART1), Address Detect Enabled

```
MOV    #baudrate,W0    ; Set Baudrate
MOV    W0,U1BRG

BSET   IPC2,#U1TXIP2   ; Set UART TX interrupt priority
BCLR   IPC2,#U1TXIP1   ;
BCLR   IPC2,#U1TXIP0   ;
BSET   IPC2,#U1RXIP2   ; Set UART RX interrupt priority
BCLR   IPC2,#U1RXIP1   ;
BCLR   IPC2,#U1RXIP0   ;

BSET   U1STA,#ADDEN    ; Enable address detect

MOV    #0x8883,W0      ; UART1 enabled for 9-bit data,
                       ; no parity, 1 STOP bit,
                       ; wakeup enabled
MOV    W0,U1MODE

BSET   U1STA,#UTXEN    ; Enable transmit

BSET   IEC0,#U1TXIE    ; Enable transmit interrupts
BSET   IEC0,#U1RXIE    ; Enable receive interrupts
```

19.10 Other Features of the UART

19.10.1 UART in Loopback Mode

Setting the LPBACK bit enables this special mode, in which the UxTX output is internally connected to the UxRX input. When configured for the Loopback mode, the UxRX pin is disconnected from the internal UART receive logic. However, the UxTX pin still functions normally.

To select this mode:

1. Configure UART for the desired mode of operation.
2. Set LPBACK = 1 to enable Loopback mode.
3. Enable transmission as defined in **Section 19.5 "UART Transmitter"**.

The Loopback mode is dependent on the UEN<1:0> bits, as shown in Table 19-2.

Table 19-2: Loopback Mode Pin Function

UEN<1:0>	Pin Function, LPBACK = 1
00	UxRX input connected to UxTX; UxTX pin functions; UxRX pin ignored; UxCTS/UxRTS unused
01	UxRX input connected to UxTX; UxTX pin functions; UxRX pin ignored; UxRTS pin functions, UxCTS unused
10	UxRX input connected to UxTX; UxTX pin functions; UxRX pin ignored; UxRTS pin functions, UxCTS input connected to UxRTS; UxCTS pin ignored
11	UxRX input connected to UxTX; UxTX pin functions; UxRX pin ignored; BCLK pin functions; UxCTS/UxRTS unused

19.10.2 Auto Baud Support

To allow the system to determine baud rates of the received characters, the UxRX input can be internally connected to a selected input capture channel. When the ABAUD bit (UxMODE<5>) is set, the UxRX pin is internally connected to the input capture channel. The ICx pin is disconnected from the input capture channel.

The input capture channel used for auto baud support is device specific. Please refer to the device data sheet for further details.

This mode is only valid when the UART is enabled (UARTEN = 1) and the Loopback mode is disabled (LPBACK = 0). Also, the user must program the capture module to detect the falling and rising edges of the Start bit.

19.11 UART Operation During CPU Sleep and Idle Modes

The UART does not function in Sleep mode. If entry into Sleep mode occurs while a transmission is in progress, then the transmission is aborted and the UxTX pin is driven to logic '1'. Similarly, if entry into Sleep mode occurs while a reception is in progress, then the reception is aborted.

The UART can be used to optionally wake the dsPIC device from Sleep mode on the detection of a Start bit. If the WAKE bit (UxSTA<7>) is set, the device is in Sleep mode, and the UART receive interrupt is enabled (UxRXIE = 1), then a falling edge on the UxRX pin will generate a receive interrupt. The Receive Interrupt Select mode bit (URXISEL) has no effect for this function. The UARTEN bit must be set in order to generate a wake-up interrupt.

The USIDL bit (UxMODE<13>) selects if the module will stop operation when the device enters Idle mode, or whether the module will continue normal operation in Idle mode. If USIDL = 0, the module will continue normal operation during Idle mode. If USIDL = 1, the module will stop in Idle mode. Any transmission or reception in progress will be aborted.

19.12 Registers Associated with UART Module

Table 19-3: Registers Associated with UART1

SFR Name	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset State
U1MODE	UARTEN	—	USIDL	—	reserved	ALTIO	reserved	reserved	WAKE	LPBACK	ABAUD	—	—	PDSEL<1:0>		STSEL	0000 0000 0000 0000
U1STA	UTXISEL	—	—	—	UTXBRK	UTXEN	UTXBF	TRMT	URXISEL<1:0>		ADDEN	RIDLE	PERR	FERR	OERR	URXDA	0000 0001 0001 0000
U1TXREG	—	—	—	—	—	—	—	UTX8	Transmit Register								0000 0000 0000 0000
U1RXREG	—	—	—	—	—	—	—	URX8	Receive Register								0000 0000 0000 0000
U1BRG	Baud Rate Generator Prescaler																0000 0000 0000 0000
IFS0	CNIF	MI2CIF	SI2CIF	NVMIF	ADIF	U1TXIF	U1RXIF	SPI1IF	T3IF	T2IF	OC2IF	IC2IF	T1IF	OC1IF	IC1IF	INT0	0000 0000 0000 0000
IEC0	CNIE	MI2CIE	SI2CIE	NVMIE	ADIE	U1TXIE	U1RXIE	SPI1IE	T3IE	T2IE	OC2IE	IC2IE	T1IE	OC1IE	IC1IE	INT0IE	0000 0000 0000 0000
IPC2	—	ADIP<2:0>		—	U1TXIP<2:0>				—	U1RXIP<2:0>			—	SPI1IP<2:0>			0100 0100 0100 0100

Note: The registers associated with UART1 are shown for reference. See the device data sheet for the registers associated with other UART modules.

19.13 Design Tips

Question 1: ***The data I transmit with the UART does not get received correctly. What could cause this?***

Answer: The most common reason for reception errors is that an incorrect value has been calculated for the UART baud rate generator. Ensure the value written to the UxBRG register is correct.

Question 2: ***I am getting framing errors even though the signal on the UART receive pin looks correct. What are the possible causes?***

Answer: Ensure the following control bits have been setup correctly:

- UxBRG: UART Baud Rate register
- PDSEL<1:0>: Parity and Data Size Selection bits
- STSEL: Stop bit Selection

19.14 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 dsPIC30F Product Family, but the concepts are pertinent and could be used with modification and possible limitations. The current application notes related to the UART module are:

Title	Application Note #
Serial Port Utilities Implementing Table Read and Table Write	AN547

Note: Please visit the Microchip web site (www.microchip.com) for additional Application Notes and code examples for the dsPIC30F Family of devices.

19.15 Revision History

Revision A

This is the initial released revision of this document.

Revision B

Revision B has been expanded to include a full description of the dsPIC30F UART module.

Revision C

This revision incorporates all known errata at the time of this document update.

NOTES: