Contents

[**Introduction:** 2](#_Toc94278442)

[**USART Baud Rate Generator (BRG):** 4](#_Toc94278443)

[**USART Asynchronous Mode:** 5](#_Toc94278444)

[USART Asynchronous Transmitter: 6](#_Toc94278445)

[**Steps to follow when setting up a Asynchronous Transmission:** 6](#_Toc94278446)

[**USART Asynchronous Receiver:** 7](#_Toc94278447)

[**Steps to follow when setting up an Asynchronous Reception:** 7](#_Toc94278448)

[**The HC-05 Bluetooth Module:** 7](#_Toc94278449)

[**Code:** 8](#_Toc94278450)

[**PIC ASYCHRONOUS UART MODE (RECIEVER):** 8](#_Toc94278451)

[Figure 1 TXSTA data sheet 3](#_Toc94279040)

[Figure 2 RCSTA data sheet 4](#_Toc94279041)

[Figure 3 registers associated with baud generator 5](#_Toc94279042)

[Figure 4 Baud rate formula 5](#_Toc94279043)

[Figure 5 Baud rates for asynchronous mode when BRGH=0 5](#_Toc94279044)

[Figure 6 Baud rates for asynchronous mode when BRGH=1 6](#_Toc94279045)

[Figure 7 Asynchronous transmission registers 7](#_Toc94279046)

[Figure 8 Synchronous slave reception registers 8](#_Toc94279047)

# **Introduction:**

The Universal Synchronous Asynchronous Receiver Transmitter (USART) module is one of the two serial I/O modules. 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), and the TRIS bits, must be set to configure the TX/CK and RX/DT pins for the USART.

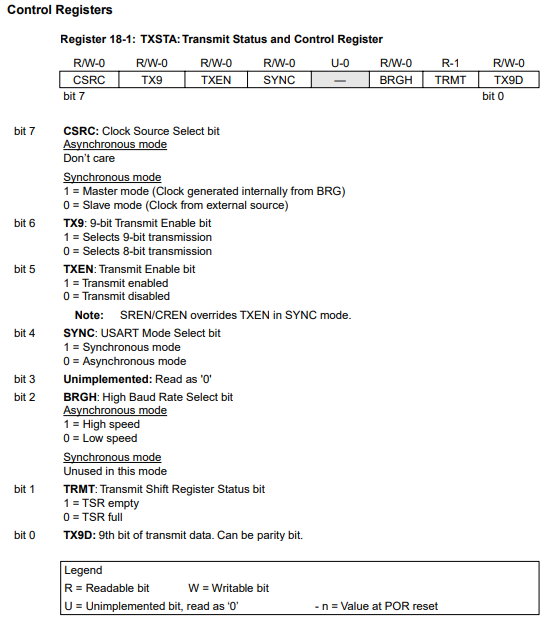


Figure TXSTA data sheet

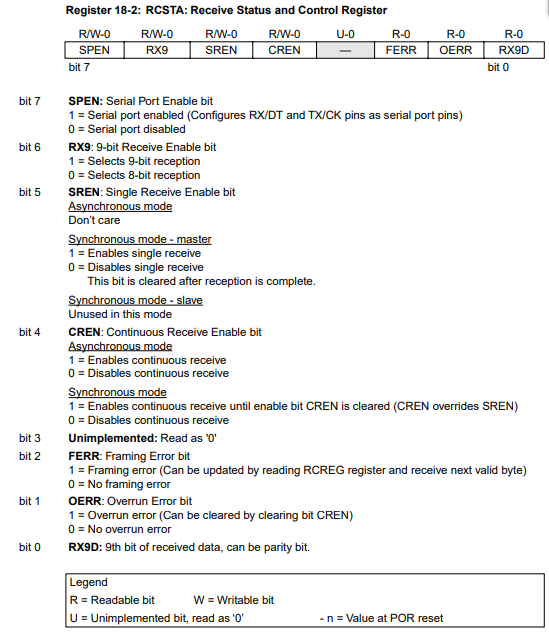


Figure RCSTA data sheet

# **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 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. 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.

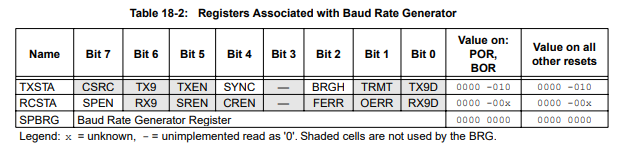


Figure registers associated with baud generator

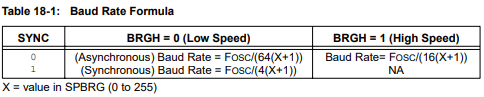


Figure Baud rate formula

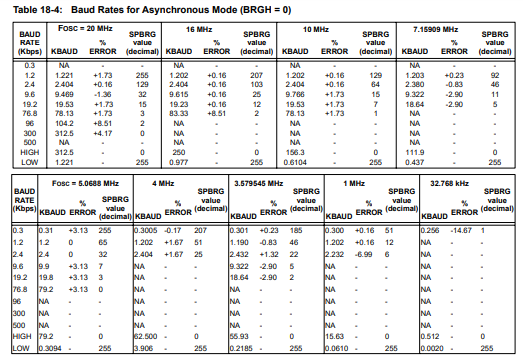


Figure Baud rates for asynchronous mode when BRGH=0

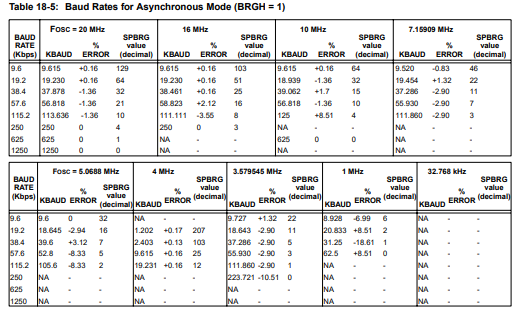


Figure Baud rates for asynchronous mode when BRGH=1

**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). Asynchronous mode is stopped during SLEEP. Asynchronous mode is selected by clearing the SYNC bit (TXSTA). The USART Asynchronous module consists of the following important elements:

• Baud Rate Generator

• Sampling Circuit

• Asynchronous Transmitter

• Asynchronous Receiver

## USART Asynchronous Transmitter:

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 TCY), 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) 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 must poll this bit in order to determine if the TSR register is empty. Transmission is enabled by setting the TXEN enable bit (TXSTA). 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. 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.

## **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.

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).

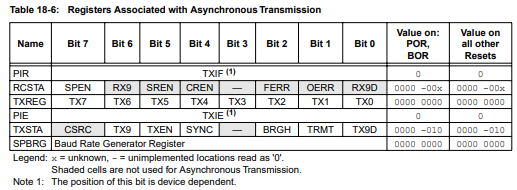


Figure Asynchronous transmission registers

## **USART Asynchronous Receiver:**

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, 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), 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), is set if a stop bit is detected as a low level.

## **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.

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.

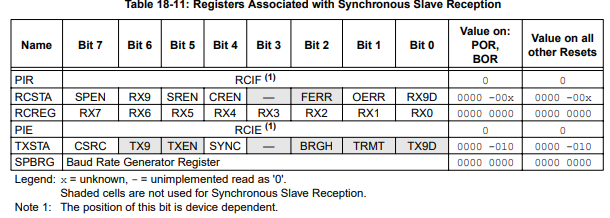


Figure Synchronous slave reception registers

# **The HC-05 Bluetooth Module:**

The Bluetooth is a wireless technology standard for exchanging data over short distances (using short-wavelength UHF radio waves in the ISM band from 2.4 to 2.485 GHz) from fixed and mobile devices and building personal area networks (PANs). The communication range is approximately 9 Meters (30 feet). We’ll be using the HC-05 Bluetooth module which communicates with microcontrollers over the serial UART bus. Finally, download “Bluetooth Controller HC-05” on an android device and start sending data.

# **Code:**

## **PIC ASYCHRONOUS UART MODE (RECIEVER):**

#INCLUDE "P16F877A.INC" ;

\_\_CONFIG \_CP\_OFF & \_WDT\_OFF & \_PWRTE\_OFF & \_BODEN\_OFF & \_LVP\_OFF & \_HS\_OSC ;

RES\_VECT CODE 0x00 ; processor reset vector

GOTO CONFI ; go to beginning of program

INT\_VECT CODE 0X04 ; interrupt vector

GOTO ISR ; go to interrupt service routine

CONFI BSF STATUS,RP0

MOVLW H'00'

MOVWF TRISD ;PORT D OUTPUT

MOVLW H'FF'

MOVWF TRISC ;PORT C UART MODE

MOVLW D'25'

MOVWF SPBRG ; SET SPBRG TO 9600

MOVLW H'04'

MOVWF TXSTA

MOVLW H'C0' ;GIE ENABLED ;PEIE ENABLED

MOVWF INTCON

MOVLW H'20' ;RCIE ENABLED

MOVWF PIE1

BCF STATUS,RP0

MOVLW H'90'

MOVWF RCSTA

GOTO MAIN

ISR MOVF RCREG,W ; RECIEVE DATA

MOVWF PORTD ; OUTPUT TO PORT D

BCF PIE1,RCIF

RETFIE

MAIN GOTO MAIN

END