PIC Serial Communication Modules

Synchronous Serial Port (SSP Module) {clock signal is required}
- Serial Peripheral Interface (SPI)
- Inter Integrated Circuit (IIC -> I^2C) Serial Interface

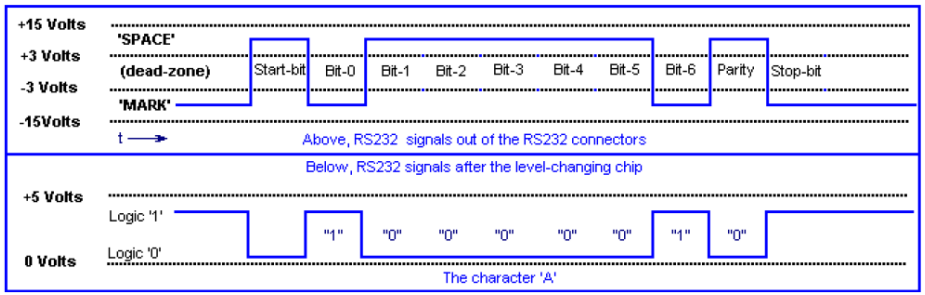
Asynchronous Communication Modules (clock signal is not required)
- Universal Asynchronous Receiver Transmitter (UART)

UART

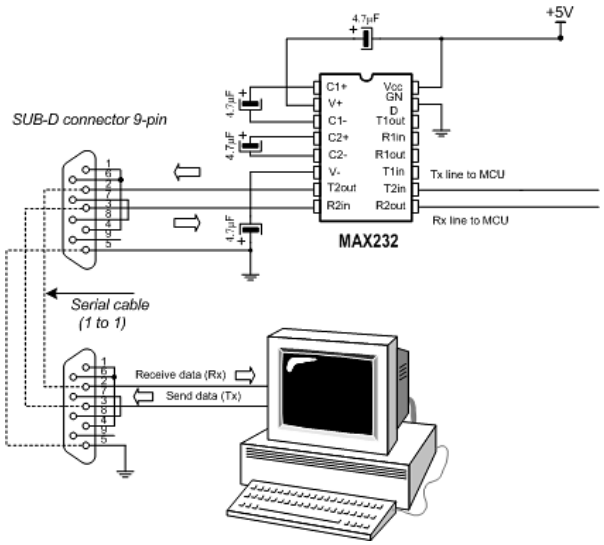
- A *Universal Asynchronous Receiver-Transmitter* (UART) is used for serial communications – usually via a cable.
- The UART generates signals with the same timing as the RS-232 standard used by the Personal Computer's COM ports.
- The UART input/output uses 0V for logic 0 and 5V for logic 1.
- The RS-232 standard (and the COM port) use +12V for logic 0 and –12V for logic 1.
- To convert between these voltages levels we need an additional integrated circuit (such as Maxim's MAX232).

Parts of an RS-232 Frame

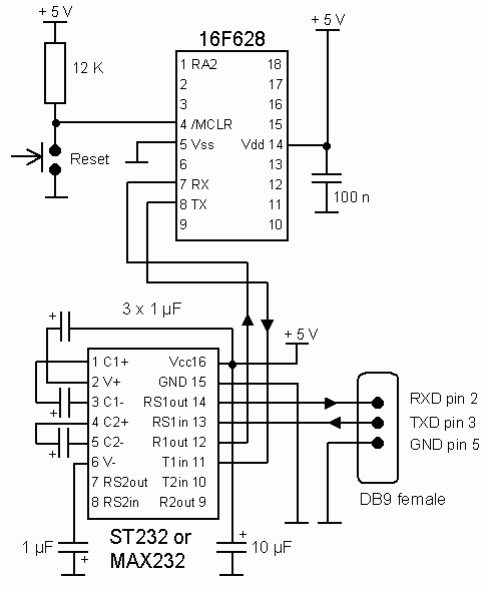
- A frame transmits a single character and is in general composed of:
  1) A start bit (always logic 0)
  2) Data bits (5, 6, 7, or 8 of them)
  3) A parity bit (optional, even or odd parity)
  4) A stop bit (always logic 1)



RS232 Level Converter



PIC & MAX232 Connection



UART

Timing Accuracy

- Since the transmitter and receiver keep track of time independently between clock recovery synchronization points, the combined inaccuracy of the transmitter and receiver's clocks can not be too large.
- For RS-232 communication, this combined inaccuracy is about 5%.
- Usually, an RC oscillator is too inaccurate and a crystal is required.

UART

Bit Time and Baud Rate

- The *bit time* (units of time) is the time from the start of one serial data bit value to the start of another.
- The inverse of the bit time is the *baud rate* (units of frequency, Hz – however the units are normally omitted).
- RS-232 baud rates are integer multiples and sub-multiples of 9600 (Hz).

Synchronization Point

- RS-232 idles at a logic 1.
- A frame always starts with either an idle-to-start-bit or a stop-bit-to-start-bit transition.
- Either way, the frame always starts with a logic 1 to logic 0 transition.
- The UART looks for 1→0 for the start of the first frame and then looks for 1→0 after 9.5BT for the start of each subsequent frame.

UART

Sampling Points

- The synchronization point is at the start of the frame (always a 1 to 0 transition).
- The 8 received data values are sampled 1.5BT, 2.5BT, … , 8.5BT after the synchronization point (BT = bit time).
- The stop bit is sampled 9.5BT after the synchronization point (if it is not a logic 1, this is a *framing error*).

UART

TXSTA – TRANSMIT STATUS AND CONTROL REGISTER (ADDRESS: 98h)

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

CSRC: Clock Source Select bit (for Synchronous mode only)
TX9: 9-bit Transmit Enable bit
1 = Selects 9-bit transmission
0 = Selects 8-bit transmission
TXEN: Transmit Enable bit
1 = Transmit enabled
0 = Transmit disabled
SYNC: USART Mode Select bit
1 = Synchronous mode
0 = Asynchronous mode
BRGH: High Baud Rate Select bit (For Asynchronous mode only)
1 = High speed
0 = Low speed
TRMT: Transmit Shift Register Status bit
1 = TSR empty
0 = TSR full
TX9D: 9th bit of transmit data. Can be parity bit.

UART

RCSTA – RECEIVE STATUS AND CONTROL REGISTER (ADDRESS: 18h)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-0	R-x
SPEN	RX9	SREN	CREN	ADEN	FERR	OERR	RX9D
bit 7							bit 0

SPEN: Serial Port Enable bit
1 = Serial port enabled
0 = Serial port disabled
RX9: 9-bit Receive Enable bit
1 = Selects 9-bit reception
0 = Selects 8-bit reception
SREN: Single Receive Enable bit (Synchronous mode only)
CREN: Continuous Receive Enable bit (in Asynchronous mode)
1 = Enables continuous receive
0 = Disables continuous receive
ADEN: Address Detect Enable bit (for Asynchronous 9-bit mode only)
FERR: Framing Error bit (read RCREG register and receive next valid byte to clear)
1 = Framing error
0 = No framing error
OERR: Overrun Error bit
1 = Overrun error (Can be cleared by clearing bit CREN)
0 = No overrun error
RX9D: 9th bit of received data (Can be parity bit)

PIC UART

Follow these steps when setting up an Asynchronous Transmission:

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 bit SYNC and setting bit SPEN.

3. If interrupts are desired, then set enable bit TXIE. **(Optional)**

4. If 9-bit transmission is desired, then set transmit bit TX9.

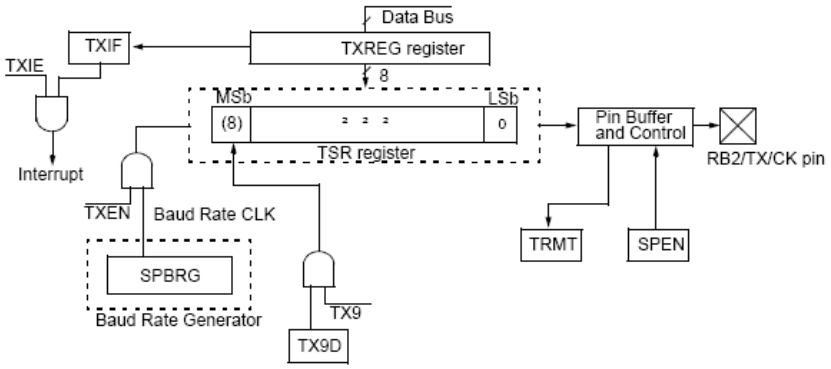
5. Enable the transmission by setting bit TXEN, which will also set bit TXIF.

6. If 9-bit transmission is selected, the ninth bit should be loaded in bit TX9D.

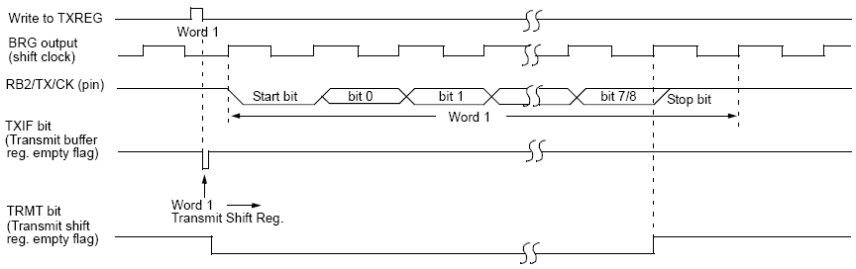
7. Load data to the TXREG register (starts transmission).

Tx pin (RB2) must be output port

UART Transmit Block



PIC UART Module



Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0Ch	PIR1	EEIF	CMIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF
18h	RCSTA	SPEN	RX9	SREN	CREN	ADEN	FERR	OERR	RX9D
19h	TXREG	USART Transmit Data Register							
8Ch	PIE1	EEIE	CMIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D
99h	SPBRG	Baud Rate Generator Register							

Baud rate calculation

TXSTA – TRANSMIT STATUS AND CONTROL REGISTER (ADDRESS: 98h)

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

BRGH = 0 (Low Speed)

Baud Rate = FOSC/(64(X+1)) ; x = value to write to SPBRG Reg

BRGH = 1 (High Speed)

Baud Rate = FOSC/(16(X+1))

Baud rate calculation

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

$$Desired\ Baud\ Rate = \frac{Fosc}{64(x+1)}$$

$$9600 = \frac{16000000}{64(x+1)}$$

$$x = 25.042$$

$$Calculated\ Baud\ Rate = \frac{16000000}{64(25+1)} = 9615$$

$$Error = \frac{(Calculated\ Baud\ Rate - Desired\ Baud\ Rate)}{Desired\ Baud\ Rate}$$

$$= \frac{9615 - 9600}{9600} = 0.16\%$$

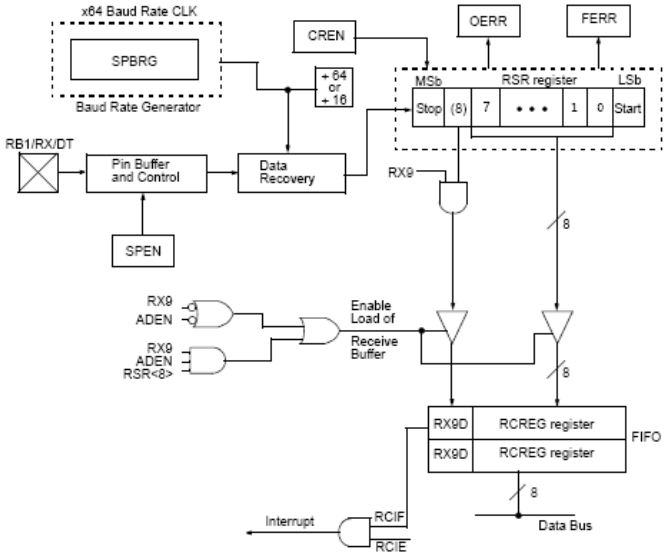
PIC USART Module

Follow these steps when setting up an Asynchronous **Reception:**

1. Initialize the SPBRG register for the appropriate baud rate.
2. Enable the asynchronous serial port by clearing bit SYNC and setting bit SPEN.
3. If interrupts are desired, then set enable bit RCIE.
4. If 9-bit reception is desired, then set bit RX9.
5. Enable the reception by setting bit CREN.
6. Flag bit RCIF will be set when reception is complete and an interrupt will be generated if enable bit RCIE 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 an OERR error occurred, clear the error by clearing enable bit CREN.

Rx pin (RB1) must be input port

UART Receive Block



PIC UART Module

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0Ch	PIR1	EEIF	CMIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF
18h	RCSTA	SPEN	RX9	SREN	CREN	ADEN	FERR	OERR	RX9D
1Ah	RCREG	USART Receive Data Register							
8Ch	PIE1	EEIE	CMIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D
99h	SPBRG	Baud Rate Generator Register							