```
;RS232.LIB  v1.0   Last modified on 26 Feb. 2006
;
;Description
;       This subroutine provides calls for RS-232 functionality, including:
;       - variable Baud rates and data bits
;       - receive bit setup
;       - Request to Send bit setup
;       - wait for a serial start bit
;       - receive a serial character
;
;Variables
;       Each of the variables below must be equated to a file register by
;       the calling program:
;       Counter         -       Used by bit time delay loop
;       BitCounter   -       Counts number of received bits
;       Receive         -       Stores the received serial byte
;       Transmit        -       Stores the serial byte to be transmitted
;
;Use
;       To receive serial characters:
;       1. Call Receive_Port to initialize serial_input as an input and CTS to output
;       2. Call Receive_Wait to wait for and receive one serial byte
;       3. (Optional) Check the Receive register for 00h indicating a
;          serial framing error has occurred.
;       To transmit serial characters:
;       1. Call Transmit_Port to initialize serial_output as an output
;       2. Move the data byte to be sent to the Transmit register
;       3. Call Transmit_Data to wait send the serial byte
;
;RS232.LIB Hardware Equates

#define         Serial_Input    PORTA,4         ;Serial input pin
#define         Serial_Output   PORTB,0         ;Serial input pin
#define         CTS             PORTB,2         ;Clear to send pin

;Software Equates
;       DataBits and Bit_Time may be commented out to allow the calling
;       program to select equates governing the number of data bits as
;       well as the received baud rate. Or, change DataBits and Bit_Time
;       below to your defaults.

#define         DataBits        0x08            ;8 data bits

;Set Bit_Time, below, with a value from the table corresponding to your
;PIC's clock speed and the serial Baud rate required.

;Clock|1MHz     |2MHz     |4MHz     |8MHz     |10MHz    |16MHz    |20MHz    |
;Baud----------------------------------------------------------------------------
;300    |0xCE     |-        |-        |-        |-        |-        |-        |
;600    |0x65     |0xCE     |-        |-        |-        |-        |-        |
;1200   |0x31     |0x65     |0xCE     |-        |-        |-        |-        |
;2400   |0x17     |0x31     |0x65     |0xCE     |0xFF*    |-        |-        |
;4800   |0x0A     |0x17     |0x31     |0x65     |0x7F     |0xCE     |0xFF*    |
;9600   |0x04*    |0x0A     |0x17     |0x31     |0x3E     |0x65     |0x7F     |
;14400  |-        |0x06     |0x0F     |0x20     |0x29     |0x43     |0x54     |
;19200  |-        |0x04*    |0x0A     |0x17     |0x1E     |0x31     |0x3E     |
;28800  |-        |-        |0x06     |0x0F     |0x13     |0x20     |0x29     |
;38400  |-        |-        |0x04*    |0x0A     |0x0E     |0x17     |0x1E     |
;57600  |-        |-        |-        |0x06     |0x08     |0x0F     |0x13     |
;----------------------------------------------------------------------------
;*Timing inaccuracies using these delay constants may cause serial errors.

#define         Bit_Time        0x08            ;Serial Bit delay from table above (0x1E)
#define         Half_Bit        Bit_Time / 2    ;Half of the Bit delay

Receive_Port                                    ;Sets Serial_Input to input.
                BSF     STATUS,RP0              ;Select memory register page 1
                MOVLW   B'00010000'             ;Load W with bit to make RA.4 input
                IORWF   TRISA                   ;and OR with Port A tristate reg.
                BCF     TRISB,2                 ; CTS paa port b.2 (output)
                BCF     STATUS,RP0              ;Return to memory register page 0
                bsf     CTS
```

```
                RETURN
Receive_Wait                                ;Waits for an RS-232 start bit indicated by Serial_Input going low.
                BCF     CTS                 ; enable CTS signal
rec_wait1
                BTFSC   Serial_Input        ;Check serial input pin
                GOTO    Rec_Wait1           ;If high, wait for low


Receive_Data
        ;Wait for serial-input to change from high to low. When it does,
        ;delay for half the Bit_Time and confirm the presence of the
        ;start bit. Then wait for the Bit_Time and read each bit into
        ;Carry. Rotate Carry into the Receive byte and repeat the
        ;delay, Carry and rotate until number of DataBits have been
        ;received. The Receive register stores the received byte. If
        ;framing error occurs (only a simple check for a stop bit is
        ;done) the contents of the Receive regsiter will be 00h.

                MOVLW   DataBits            ;Load W with number of data bits
                MOVWF   BitCounter          ;and save in BitCounter register
                MOVLW   Half_Bit            ;Load W with half of bit delay time
                CALL    BitDelay            ;and wait for 1/2 bit

                BTFSC   Serial_Input        ;Check for low start bit again
                GOTO    Receive_Data        ;If high, error occurred-keep waiting
                BSF     CTS                 ;Disable CTS signal
NextR_bit
                MOVLW   Bit_Time            ;Load W with bit delay time
                CALL    BitDelay            ;and wait until middle of next bit
                BTFSS   Serial_Input        ;Check serial input pin for 1
                BCF     STATUS,C            ;If serial input is 0, clear Carry
                BTFSC   Serial_Input        ;Check serial input pin for 0
                BSF     STATUS,C            ;If serial input is 1, set Carry
                RRF     Receive             ;Rotate Carry into received data byte
                DECFSZ  BitCounter          ;Decrement bit counter & check for 0
                GOTO    NextR_bit           ;If not 0, get the next bit

                MOVLW   Bit_Time            ;Load W with bit delay time
                CALL    BitDelay            ;and wait until middle of stop bit
                BTFSS   Serial_Input        ;Check for high stop bit
                GOTO    Receive_Error       ;If low, we have a framing error
                RETURN                      ;Otherwise, return

Receive_Error                               ;Simply clears Receive buffer if no stop bit is found.
                CLRF    Receive             ;If a framing error occurs, clear
                RETURN                      ;Receive register befor returning

Transmit_Port                               ;Sets Serial_output to output.
                BSF     STATUS,RP0          ;Select memory register page 1
                MOVLW   B'11111110'         ;Load W with bit to make Serial_output to output
                ANDWF   TRISB               ;and OR with Port B tristate reg.
                BCF     STATUS,RP0          ;Return to memory register page 0
                BSF     Serial_Output       ;Set serial line high
                RETURN


Transmit_Data
        ;Drop PortA.4 from high to low to indicate the Start Bit and
        ;delay for one Bit_Time. Rotate the Transmit buffer right
        ;into Carry and set or clear the serial output pin based on
        ;Carry. Wait for another bit time and continue rotating and
        ;transmitting until all eight bits have been sent. Finally,
        ;send a stop bit.

                MOVWF   Transmit            ;Save character in W to buffer
                MOVLW   DataBits            ;Load W with number of data bits
                MOVWF   BitCounter          ;and save in BitCounter regsiter
                BCF     Serial_Output       ;Send Start bit
                NOP                         ;and pad routine to be same length
                NOP                         ;as :Next_Bit code so Bit_Time is
                NOP                         ;accurate
```

```
145:                NOP
146:                NOP
147:                NOP
148:                NOP
149: NextT_Bit
150:                MOVLW   Bit_Time        ;Load W with bit delay time
151:                CALL    BitDelay        ;and wait one bit duration
152:                RRF     Transmit        ;Rotate Transmit byte into C
153:                BTFSS   STATUS,C        ;Check Carry for a 1
154:                BCF     Serial_Output   ;If C=0, clear serial output
155:                BTFSC   STATUS,C        ;Check Carry for a 0
156:                BSF     Serial_Output   ;If C=1, set serial output
157:                DECFSZ  BitCounter      ;Decrement bit counter & check for 0
158:                GOTO    NextT_Bit       ;If not 0, get the next bit
159:                MOVLW   Bit_Time        ;Load W with bit delay time
160:                CALL    BitDelay        ;and wait until end of last bit
161:                BSF     Serial_Output   ;Set serial line high for Stop bit
162:                MOVLW   Bit_Time        ;Load W with bit delay time
163:                CALL    BitDelay        ;and wait a bit
164:                RETURN
165:
166: BitDelay                               ;RS-232 Bit time period delay
167:                MOVWF   Counter         ;Move delay time in W to Counter
168: Loop           NOP                     ;Pad loop to 4 cycles
169:                DECFSZ  Counter,1       ;Decrement bit counter
170:                GOTO    Loop            ;and do it until zero
171:                RETURN
```