USART FUNCTIONS

The information in this document is obtained from the following Microchip manuals:
- PIC18Fxx2 Datasheet
- PICmicro® 18C MCU Family Reference Manual
- MPLAB C18 C Compiler Libraries

Function Prototypes:
For a detailed description of these functions, please see:
Section 2.10 USART Functions, in MPLAB C18 C Compiler Libraries manual.

#include <usart.h>

void OpenUSART ( unsigned char config, unsigned int SPBRG);  // Configure the USART.
char BusyUSART ( void );                                   // Is the USART transmitting?
void WriteUSART ( char data );                              // Write a byte (one character) to the USART transmit buffer.
void putcUSART ( char data );                                // putcUSART is defined as WriteUSART.
void putsUSART ( char *data );                              // Write a string from data memory to the USART, including the null character.
void putrsUSART ( const rom char *data );                   // Write a string from program memory to the USART, including the null character.
char DataRdyUSART( void );                                  // Is data available in the read buffer?
char ReadUSART   ( void );                                  // Read a byte (one character) out of the USART receive buffer.
char getcUSART   ( void );                                  // getcUSART is defined as ReadUSART.
void getsUSART ( char *buffer, unsigned char len );          // Read a fixed-length string of characters from the USART.
void CloseUSART ( void );                                   // Disable the USART.

Notes:

1. The three most useful functions in the above list are OpenUSART, getcUSART and CloseUSART.

2. The getcUSART function waits for and reads len number of characters out of the USART. There is no time out when waiting for characters to arrive.

3. For serial output, the preferred functions are putc(...), puts(...), printf(...) etc., which are listed in the Character Output Functions document.
<table>
<thead>
<tr>
<th>SPBRG [0...255]</th>
<th>PIC Baud Rate</th>
<th>PC Baud Rate</th>
<th>Percent Error</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>255</td>
<td>4883</td>
<td>4800</td>
<td>1.73%</td>
<td>Good</td>
</tr>
<tr>
<td>129</td>
<td>9615</td>
<td>9600</td>
<td>0.16%</td>
<td>Best</td>
</tr>
<tr>
<td>64</td>
<td>19231</td>
<td>19200</td>
<td>0.16%</td>
<td>Best</td>
</tr>
<tr>
<td>32</td>
<td>37879</td>
<td>38400</td>
<td>-1.36%</td>
<td>Good</td>
</tr>
<tr>
<td>21</td>
<td>56818</td>
<td>57600</td>
<td>-1.36%</td>
<td>Good</td>
</tr>
<tr>
<td>10</td>
<td>113636</td>
<td>115200</td>
<td>-1.36%</td>
<td>Good</td>
</tr>
<tr>
<td>5</td>
<td>208333</td>
<td>230400</td>
<td>-9.58%</td>
<td>Unusable</td>
</tr>
<tr>
<td>2</td>
<td>416667</td>
<td>460800</td>
<td>-9.58%</td>
<td>Unusable</td>
</tr>
<tr>
<td>1</td>
<td>625000</td>
<td>921600</td>
<td>-32.18%</td>
<td>Unusable</td>
</tr>
</tbody>
</table>

Note1: For error-free communication, the serial port baud rate tolerance should not exceed +/-1% as shown by the 'Best' values.

Note2: Most PC serial ports can tolerate a +/-5% error variation from the standard baud rate in the 'PC Baud Rate' column without encountering any communication errors as shown by the 'Good' values.