



DKAN0008A

PIC18 Software UART Timing Requirements

11 June 2009

Introduction

Design conditions often limit the hardware peripherals available for an embedded system. Perhaps the available hardware UARTs are already allocated, or the system requires alternate functionality (creating a conflict when the UART shares a common pin). Implementing the UART in software is one common solution.

The Microchip MPLAB C18 C Compiler Libraries include software UART functions. Implementing this UART requires the user to calculate and meet the specific timing requirements for the desired baud rate. This application note provides information on the time delays required in the serial data sampling routines and the relationship between the sampling rate and the serial data.

Background

The MPLAB C18 compiler features both hardware and software libraries for each communications peripheral found in Microchip's PIC18 family of microcontroller products. The hardware libraries take full advantage of port registers and assigned port pins, while the software libraries allow the user to specify and configure GPIO pins as a communications port. Familiarity with the MPLAB IDE environment, the MPLAB C18 C Compiler, and RS232 is assumed.

The default library functions expect a host serial port configured for 8 data bits, 1 stop bit, no parity, and no flow control. Microchip provides all of the assembly files for the library, so the developer can modify the configuration and/or port pins.

Application

The C18 library provides functions for initializing ports and transmitting/receiving serial data. However, the embedded developer must provide the delay routines used by the library functions, namely *DelayTXBitUART*, *DelayRXHalfBitUART*, and *DelayRXBitUART*. These routines set delays that correspond to the bit periods during communication. This provides a means to synchronize data transfer and sample received data.

Incoming data is identified by continuously polling the receive pin. A high-to-low transition indicates the start bit of the serial data frame. A half-bit delay sets the sample point at the start bit's midpoint. After the initial half-bit delay, full-bit delays are used to sample the remaining data. This is repeated for each frame of received data. Figure 1 illustrates this technique for one frame.

For example, a nominal clock frequency F_{osc} of 4MHz and 3 instruction cycles in the polling routine results in a t_{ird} of 3us.

Receive Half-Bit and Full-Bit Delays

The receive half-bit and full-bit delays determine the number of instruction cycles required for any oscillator and bit rate combination (see also the C18 library documentation, Chapter 3.6). Equations (2) and (3) define these relationships, respectively.

$$(2) \quad RXhalfbitdelay_{instruction_cycles} = \frac{\left(\frac{2 \times F_{osc}}{8 \times baud}\right) + 1}{2} - 9_{(instruction_cycles)}$$

$$(3) \quad RXbitdelay_{instruction_cycles} = \frac{\left(\frac{2 \times F_{osc}}{4 \times baud}\right) + 1}{2} - 14_{(instruction_cycles)}$$

For example, a nominal clock frequency F_{osc} of 4MHz and a baud rate of 9600bps require a half-bit delay of 44 instruction cycles and a full-bit delay of 91 instruction cycles.

Use equations (4) and (5) to calculate the time delay provided by the half-bit and full-bit instruction cycle delays, respectively. For the example above, the half-bit time delay is 53us, and the full-bit time delay is 105us.

$$(4) \quad Thalfbit_{rx_delay} = \frac{(RXhalfbitdelay_{instruction_cycles} + 9) \times 4}{F_{osc}}$$

$$(5) \quad Tbit_{rx_delay} = \frac{(RXbitdelay_{instruction_cycles} + 14) \times 4}{F_{osc}}$$

For received data at 9600bps, the total time period (1 start bit, 8 data bits) is 937.5us, with the last data bit's center point at 885.4us. Sampling the incoming data requires 1 *Thalfbit* and 8 *Tbit* delays, totaling 893us. The sample point of the last data bit is offset by approximately 7.6us from the center of the bit period.

Clock Tolerance Error

The clock variation between the transmitter and receiver also contribute to the sampling error. Clock error $Tcte$ is the time between the sample point and the n^{th} data bit of the transmitted data frame. The sample window bounded by $Tcte_{upper}$ and $Tcte_{lower}$ varies, since microcontroller internal oscillators can vary by $\pm 2\%$ at ambient and $\pm 10\%$ over the operational temperature range. f_{max} and f_{min} are the worst case clock frequencies in the application's environment. Equations (6) and (7) determine if the final sample occurs before the end of the n^{th} bit of the data frame.

$$(6) \quad Tcte_{upper} = \frac{nbits - 0.5}{\frac{f_{rx,min}}{f_{rx,nom}} \times baud_{nom}} - \frac{nbits}{\frac{f_{tx,max}}{f_{tx,nom}} \times baud_{nom}}$$

$$(7) \quad Tcte_{upper} < 0$$

$nbits$ is the number of bits in the packet (including start and data bits). Equations (8) and (9) ensure that the sample takes place after the n^{th} bit begins.

$$(8) \quad Tcte_{lower} = \frac{nbits - 0.5}{\frac{f_{rx,max}}{f_{rx,nom}} \times baud_{nom}} - \frac{nbits - 1}{\frac{f_{tx,min}}{t_{tx,nom}} \times baud_{nom}}$$

$$(9) \quad Tcte_{lower} > 0$$

Assume a host transmit bit rate of 9600bps, 9 bits to recover (1 start bit, 8 data bits), and a 4MHz clock with an accuracy of $\pm 2\%$. The receiving microcontroller's clock of 4MHz also has an accuracy of $\pm 2\%$. Using equation (6) and (8) results in $Tcte_{upper} = -15.6\mu s$ and $Tcte_{lower} = 17.7\mu s$, meeting the timing boundary conditions in equation (7) and (9).

With the transmit clock accuracy decreased to $\pm 4\%$, $Tcte_{upper} = 2.0\mu s$ and $Tcte_{lower} = 0\mu s$, failing the boundary requirements, resulting in communication errors at the worst case clock frequencies.

Total Receive Error

Total receiver error $Ttre$ is the sum of the initial response delay, the half-bit and full-bit delays, and the clock tolerance error. In effect, the sample window, bounded by the clock variation, is shifted in time due to the accumulation of timing errors. This shift can cause the sample window to fall outside the data bit boundaries, resulting in communication errors. Equations (10) and (11) determine if the final sample occurs before the end of the n^{th} bit of the data frame. Equations (12) and (13) ensure that the sample takes place after the n^{th} bit begins.

$$(10) \quad Ttre_{upper} = Tcte_{upper} + \frac{f_{rx,nom}}{f_{rx,min}} \left(t_{ird} + T_{halfbit_{rx_delay}} + \left(T_{bit_{rx_delay}} (nbits - 1) \right) \right)$$

$$(11) \quad Ttre_{upper} < \frac{nbits - 0.5}{baud_{nom}}$$

$$(12) \quad Ttre_{lower} = Tcte_{lower} + \frac{f_{rx,nom}}{f_{rx,max}} \left(t_{ird} + T_{halfbit_{rx_delay}} + \left(T_{bit_{rx_delay}} (nbits - 1) \right) \right)$$

$$(13) \quad Ttre_{lower} > \frac{nbits - 0.5}{baud_{nom}}$$

Expanding on the prior example, assume a host transmit bit rate of 9600bps, 9 bits to recover (1 start bit, 8 data bits), and a 4MHz clock with an accuracy of $\pm 2\%$. The receiving microcontroller's 4MHz clock has an accuracy of $\pm 2\%$. Applying the results from equations (1), (4), (5), (6) and (8), in equations (10) and (12) results in $Ttre_{upper} = -15.6\mu s$ and $Ttre_{lower} = 17.7\mu s$, meeting the timing boundary conditions in equation (11) and (13).

Accumulated timing effects can move the sample point to the edge of the bit period. Noise on the line or capacitive effects altering the signal edge rate can further reduce the sample margin until data is not reliably recovered.

Transmission Full-Bit Delay Error

Data transmission from the target also introduces timing errors from the full-bit delay inaccuracy and clock tolerance. The transmit full-bit delay, equation (14), determines the number of instruction cycles required for any oscillator and bit rate combination. Equation (15) calculates the time delay provided by the full-bit instruction cycle delay.

$$(14) \quad TXbitdelay_{instruction_cycles} = \frac{\left(\frac{2 \times F_{osc}}{4 \times baud}\right) + 1}{2} - 12_{(instruction_cycles)}$$

$$(15) \quad Tbit_{tx_delay} = \frac{(TXbitdelay_{instruction_cycles} + 12) \times 4}{F_{osc}}$$

For a microcontroller clock frequency F_{osc} of 4MHz and a baud rate of 9600bps, a full-bit delay of 93 instruction cycles is required. The full-bit time delay is 105us.

Transmission timing errors accumulate as the start bit, 8 data bits, and stop bit are transmitted and may impact the ability of the host receiver to recover the data.

Conclusion

A conservative timing margin must be maintained to ensure reliable data transfer. If high bit rates are required, a tight clock tolerance enhances the timing margin. Increasing the clock frequency reduces the software half-bit and full-bit delay inaccuracies. Reducing the serial data rate provides longer bit periods, increasing the timing margin.

Using software UART is not without pitfalls. But, once the contributing error factors are understood and ameliorated, a reliable serial communication port can be implemented using any pair of GPIO pins.

Additional Information

MPLAB® C18 C Compiler Libraries, Chapter 3.6; Microchip Technology, Inc.

Disclaimer

This document is for informational use only and is subject to change without prior notice. Digi-Key makes no commitment to update or keep current the information contained herein. Digi-Key does not guarantee or warrant that any information provided is accurate, complete, or correct and disclaims any and all liability associated with the use of the information contained herein. The use of this information and Digi-Key's liability is subject to Digi-Key's standard Terms & Conditions which can be found at www.digi-key.com by clicking on the Terms & Conditions link at the bottom of the web page.

No license, whether express, implied, arising by estoppel or otherwise is granted under any intellectual property or other rights of Digi-Key or others.

Trademarks

DIGI-KEY® is a registered trademark of Digi-Key Corporation. All other trademarks, service marks and product names contained herein are the sole property of their respective owner and their use is for informational purposes only and does not imply any endorsement, recommendation, sponsorship or approval by the trademark owner of the contents.

Copyright

Use of this document is limited to customer's internal business use for the evaluation and purchase of products. No permission is granted to the user to copy, print, store, distribute, transmit, display in public or modify the content of this document in any way for any other purpose.

© Copyright 2009 Digi-Key Corporation. All rights reserved.