AN946

# Interfacing the MCP2122 to the Host Controller


Author: *Mark Palmer*
*Microchip Technology Inc.*


## INTRODUCTION

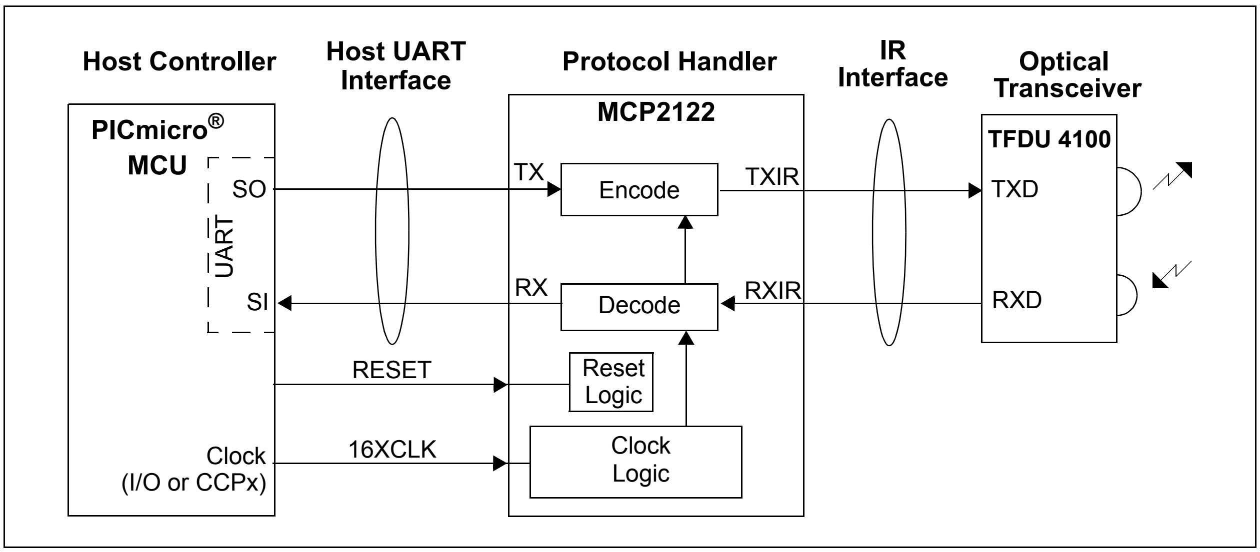
The MCP2122 has up to four signals that can be interfaced to the embedded systems Host Controller. These signals are the transmitted and received data (TX and RX), the 16XCLK signal and the RESET signal.

The MCP2122 is pinout compatible with the Agilent® HSDL-7000 Encoder/Decoder.

The 16x clock is used for the baud clock timing (transmit and receive). There are 16 16x clocks (16XCLK) for each bit time. In systems that have already been designed with the HSDL-7000, the 16XCLK signal is already present. For systems that are to be designed, the generation of a 16x clock signal will need to be done. For embedded systems using a PICmicro® microcontroller unit (MCU) as the Host Controller, the Capture/Compare/PWM (CCP) and Timer2 modules can be used to generate the 16XCLK signal. This is accomplished using the CCP module in PWM mode.

This application note will discuss methods with which to interface the MCP2122 to a Host Controller and how to use the PICmicro MCU's CCP and Timer2 modules to generate the 16XCLK signal.

**FIGURE 1: TYPICAL MCP2122 SYSTEM BLOCK DIAGRAM**



## HOST UART INTERFACE

The Host UART interface has two signals, the Transmit signal (TX) and the Receive signal (RX). These signals are connected to the Host Controller's UART, which can either be a hardware module or implemented via firmware.

## IR INTERFACE

The IR interface has two signals, the Transmit signal (TXIR) and the Receive signal (RXIR). These signals are connected to the optical transceiver circuitry, which can either be an integrated device or implemented with discrete components.

## 16XCLK

For the MCP2122 state machine to operate, the device must be clocked. The frequency of the clock on the 16XCLK pin must be 16 times the desired baud rate (see Table 1). Figure 2 shows the relationship of the 16XCLK signal to either the UART signals (TX and RX signals).

The source of this clock could be from any number of sources, including:

- Board System Clock
- Host controller-generated clock
  - Firmware-generated clock on I/O pin
  - Hardware-generated clock using time-based module (such as the PICmicro MCU CCP module)

The clock source is one of the differences between the MCP2122 and the MCP2120. The MCP2120 requires a crystal to create the device clock. The MCP2120's internal 16XCLK signal (device baud rate) is then specified by the state on the BAUD2:BAUD0 pins. The MCP2120's BAUD2:BAU0 pins can be configured so that the baud rate is determined by the state of these pins (Hardware mode) or controlled by the Host Controller (Software mode).

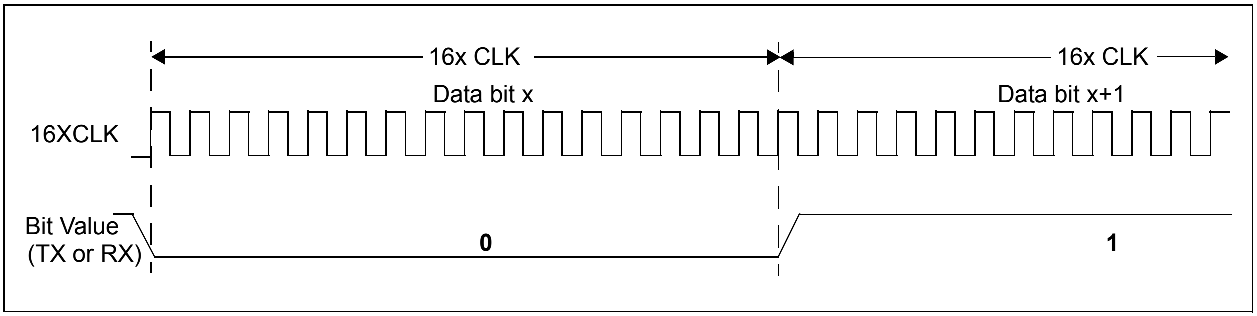
### Board System Clock

The Board System Clock is the easiest since it is already available. All that needs to be done is to connect the clock to the 16XCLK pin. This means that the baud rate will be 1/16th of the frequency of this clock.

**TABLE 1: UART/IR BAUD RATE AND 16XCLK**

| UART/IR Baud Rate | 16XCLK | Comment |
|---|---|---|
| 9,600 | 153,600 | |
| 19,200 | 307,200 | |
| 38,400 | 614,400 | |
| 57,600 | 921,600 | |
| 115,200 | 1,843,200 | Maximum device baud rate |

**FIGURE 2: 16XCLK TO TX OR RX SIGNAL**



### Host Controller Firmware-Generated Clock

The Host Controller's firmware can implement a 16XCLK signal. However, this may add complexity of timing-related issues for other parts of the application firmware. As the desired baud rate increases (relative to the system clock), the number of "free" instruction cycles performing other firmware operations decreases.

Every 16XCLK clock (pulse) has a rising edge and a falling edge that must be generated. With a PICmicro MCU, the rising edge can be generated with the BSF instruction and the falling edge with the BCF instruction. So two instructions are required in each 16XCLK clock (pulse). Table 3 shows the available PICmicro MCU instruction cycles (at a given device frequency) for the desired Host UART/IR interface baud rate. This probably means that any application that was planning to generate the 16XCLK in firmware would not want to exceed an IR baud rate of 19200. In most cases, 9600 would probably be desired to ensure sufficient CPU bandwidth to accomplish other tasks.

### Host Controller Hardware-Generated Clock

Using hardware features of the Host Controller can eliminate the processing overhead of a firmware implementation, as well as allow much faster baud rates to be used.

A typical PICmicro device system implementation would consist of using the CPP module to generate the clock, with the CCP module being configured in PWM mode. The PWM frequency would be the required 16XCLK frequency (for the desired baud rate).The PWM duty cycle should be configured for about 50%. After initial configuration of the module (after a reset of the PICmicro device), no additional software overhead is required. This clock will be synchronized to the operation of the PICmicro device UART.

In the section entitled "Using the PICmicro® MCU's CCP Module to Generate the 16XCLK" discusses the details of using the PICmicro CCP module.

> **Note:** There are other techniques that could be used with PICmicro devices to generate the 16XCLK signal. These techniques include:
>
> 1. Use the hardware USART to generate the 16XCLK clock output and implement the TX and RX functions in firmware. This is done by using the USART in Synchronous Master mode with the Continuous Receive enabled (CREN bit is set). The selected clock of the Baud rate will be output on the TX/CLK pin of the PICmicro MCU.
> 2. With devices that have an accurate 8 MHz internal RC clock, select the option to output that clock and use an external circuit to create the desired 16XCLK frequency.

## RESET

The RESET pin is used to put the MCP2122 into a known state. Forcing the RESET pin low will immediately force the output pins to their default output state, as shown in Table 2.

**TABLE 2: MCP2122 DEFAULT OUTPUT PIN STATES IN DEVICE RESET**

| Input Pin | | Output Pin State | | Comments |
|---|---|---|---|---|
| Name | State | RX | TXIR | |
| RESET | L | H | L | Device in reset mode |

**TABLE 3: PICMICRO® MCU FREQUENCY/INSTRUCTION CYCLES VS. HOST UART/IR BAUD RATE**

| UART/IR Baud Rate | 16XCLK | Instruction Frequency (4 X) | Instruction Cycles @ Device Frequency (MHz) | | | Comment |
|---|---|---|---|---|---|---|
| | | | 14.7456 | 18.432 | 19.6608 | |
| 9,600 | 153,600 | 614,400 | 24 | 30 | 32 | |
| 19,200 | 307,200 | 1,228,800 | 12 | 15 | 16 | |
| 38,400 | 614,400 | 2,457,600 | 6 | 7.5 | 8 | |
| 57,600 | 921,600 | 3,686,400 | 4 | 5 | 5.33 | |
| 115,200 | 1,843,200 | 7,372,800 | 2 | 2.5 | 2.67 | Maximum baud rate |

## USING THE PICmicro® MCU'S CCP MODULE TO GENERATE THE 16XCLK

The CCP module is a hardware module that can be configured for Capture, Compare or PWM operation. This module works in conjunction with either Timer1 or Timer2, depending on the CCP mode selected.

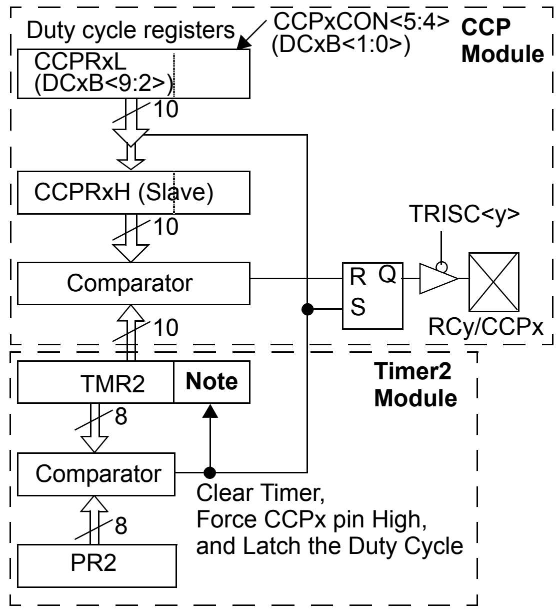
PWM mode is required for the generation of a 16XCLK. For PWM generation, the Timer2 module is used to create the time-base. A simplified block diagram of the PIC16F877A's CCP module in PWM mode is shown in Figure 3. The PIC16F877A has two CCP modules; this figure is generic for either.

A PWM output waveform is shown in Figure 4. The period of the PWM determines how often the waveform repeats, while the duty cycle determines the relationship between the time the signal is high to the time the signal is low. For the 16XCLK signal, a 50% time is desired.

The PWM module takes advantage of the PICmicro MCU's internal Q clocks (4Q clocks for an instruction cycle) for the generation and duty cycle of the PWM. This allows slower device frequencies (lower power) to be used for a given baud rate. The minimum PWM period can be an instruction cycle (TCY).

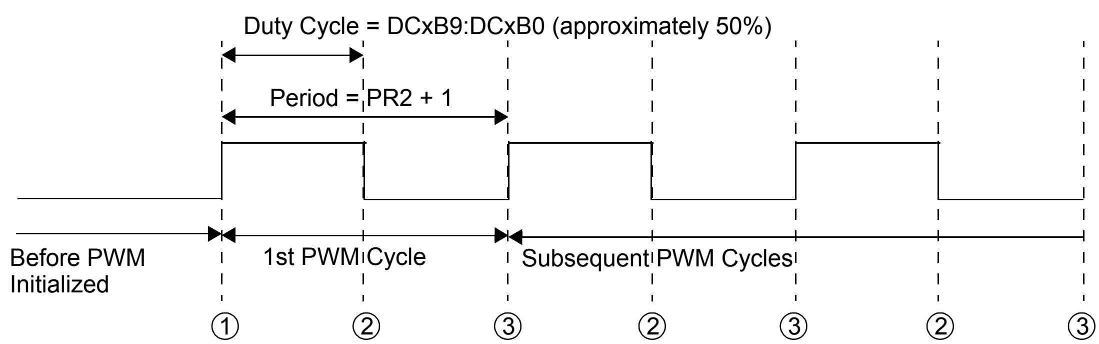
| | |
|---|---|
| **Note:** | There are some small, low-cost PICmicro devices that have the CCP module. These include the 8-pin PIC12F683 and 14-pin PIC16F684. |

**FIGURE 3: CCP MODULE - PWM MODE BLOCK DIAGRAM**




| | |
|---|---|
| **Note:** | 8-bit timer is concatenated with 2-bit internal Q clock or 2 bits of the prescaler to create 10-bit time base. |

**FIGURE 4: PWM OUTPUT WAVEFORM**




1. Timer2 is cleared and new duty cycle value is loaded from the duty cycle latch into the duty cycle slave register
2. Timer2 value equals to value in duty cycle latch register, with CCP pin driven low
3. Timer2 matches PR2 register, value from duty cycle latch is loaded into slave register, CCP pin driven high, with Timer2 being forced clear

## PWM period

The PWM period is specified by writing to the PR2 register. The PWM period can be calculated using Equation 1.

### EQUATION 1: CALCULATION FOR PWM PERIOD

$$T_{PWM\ period} = [(PR2) + 1] \bullet 4 \bullet T_{OSC} \bullet (\text{TMR2 prescale value})$$

Where:

PR2 = Value in PR2 Register

$T_{OSC}$ = Oscillator Clock

PWM frequency is defined as 1/ [PWM period].

When TMR2 is equal to PR2, the following three events occur on the next increment cycle:

1. TMR2 is cleared
2. The CCPx pin is set (exception: if PWM duty cycle = 0%, the CCPx pin will not be set)
3. The PWM duty cycle is latched from CCPRxL into CCPRxH

The period of the PWM determines the frequency of the 16XCLK signal. This corresponds to the UART/IR baud rate. Table 4 shows, at the common UART/IR baud rates and PICmicro MCU frequencies, what the PR2 (PWM period) and CCPRxL (PWM duty cycle) values are.

## PWM Duty Cycle

The PWM duty cycle is specified by writing to the DCxB9:DCxB0 bits. DCxB9:DCxB2 are contained in the CCPRxL register, while DCxB1:DCxB0 are located at CCPxCON<5:4>). The CCPRxL contains the eight MSbs, while CCPxCON<5:4> contains the two LSbs. This 10-bit value is represented by DCxB9:DCxB0. Equation 2 is used to calculate the PWM duty cycle.

### EQUATION 2: CALCULATING THE PWM DUTY CYCLE

$$\text{PWM Duty Cycle} = (\text{DCxB}<9:0> \text{ bits value}) \bullet T_{OSC} \bullet (\text{TMR2 prescale value})$$

Where:

PWM Duty Cycle = PWM Duty Cycle Time

$T_{OSC}$ = Oscillator Clock

Though the DCxB<9:0> bits can be written to at any time, the duty cycle value is not latched into CCPRxH until after a match between PR2 and TMR2 occurs (which marks the end of the current period). In PWM mode, CCPRxH is a read-only register.

The CCPRxH register and a 2-bit internal latch are used to double buffer the PWM duty cycle. This double buffering is essential for glitchless PWM operation.

When CCPRxH and a 2-bit latch match the value of TMR2 concatenated with the internal 2-bit Q clock (or two bits of the TMR2 prescaler), the CCPx pin is cleared. This marks the end of the duty cycle.

**Note:** If the PWM duty cycle value is longer than the PWM period, the CCPx pin will not be cleared. This allows a duty cycle of 100%.

Table 4 shows the PR2 (PWM period) and CCPRxL (PWM duty cycle) values to configure the PWM for a given UART/IR baud rate at selected device frequencies.

### TABLE 4: PICmicro® MCU FREQUENCY VS. HOST UART/IR BAUD RATE

| UART/IR Baud Rate | 16XCLK Frequency | Instruction Frequency (4 X) | TMR2 Prescaler | Device Frequency (MHz) | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | 14.7456 | | 7. 3728 | | 3.6864 | |
| | | | | PR2 | CCPRxL (1) | PR2 | CCPRxL (1) | PR2 | CCPRxL (1) |
| 9,600 | 153,600 | 614,400 | 1 | 23 | 11 | 11 | 5 | 5 | 2 |
| 19,200 | 307,200 | 1,228,800 | 1 | 11 | 5 | 5 | 2 | 2 | 1 |
| 38,400 | 614,400 | 2,457,600 | 1 | 5 | 2 | 2 | 1 | — (4) | — |
| 57,600 | 921,600 | 3,686,400 | 1 | 3 | 1 | 1 | 0 (2) | 0 | 0 (3) |
| 115,200 | 1,843,200 | 7,372,800 | 1 | 1 | 0 (2) | 0 | 0 (3) | — (5) | — |

**Note 1:** The CCPxX:CCPxY bits are always set to ‘11’b

**2:** CCPxX:CCPxY = ‘00’b will supply approximately a 50% duty cycle when PR2 = 1 and CCPRxL = 1.

**3:** CCPxX:CCPxY = ‘10’b will supply a 50% duty cycle when PR2 = 0 and CCPRxL = 0.

**4:** A PR2 value can not be selected for this baud rate (too much clock error).

**5:** PWM can not be configured for this baud rate at this device frequency.

### CCP Pin

The CCPx pin is multiplexed with the PORT data latch and the corresponding TRIS bit must be cleared to make the CCPx pin an output.

### Set-up for PWM Operation

The following steps configure the CCP module for PWM operation:

1. Establish the PWM period by writing to the PR2 register.
2. Establish the PWM duty cycle by writing to the DCxB9:DCxB0 bits (CCPR1L register and CCP1CON<5:4> bits).
3. Make the CCPx pin an output by clearing the appropriate TRISC bit.
4. Establish the TMR2 prescale value and enable Timer2 by writing to T2CON.
5. Configure the CCP module for PWM operation.

### Sleep Operation

When the PIC16F877A is placed in sleep, Timer2 will not increment and the state of the module will not change. If the CCP pin is driving a value, it will continue to drive that value. When the device wakes up, it will continue from this state.

| | |
|---|---|
| **Note:** | If it is desired to place the PICmicro device in Sleep mode, it is good practice to force the PWM pin to a known state before going to “sleep”. This can be done by disabling the PWM (CCP and Timer2 modules) and forcing the I/O pin to the desired state. |

### Effects of a Reset

When the PIC16F877A is reset, the Timer2 and CCP modules are forced off. The Timer2 and CCP module will need to be reconfigured for PWM operation.

## SUMMARY

This application note has shown how a Host Controller can be interfaced to the MCP2122. Several methods have been discussed, with a detailed discussion on how to use the PICmicro MCU’s CCP module. The source code for the example PIC16F877A code is provided in **Appendix A**. **Appendix B** shows screen captures of the PWM (16XCLK) and data bit timings from the PIC16F877A code. This should help bootstrap the development of any new applications that plan on using the MCP2122.

## APPENDIX A: PIC16F877A SOURCE CODE - MCP2122TX.ASM

### FIGURE A-1: MCP2122TX.ASM - PAGE 1

```
    LIST    C=132
    include P16F877A.inc
        ERRORLEVEL -302, -306
;
/////////////////////////////////////////////////////////////////////////////////////////
;                         Software License Agreement
;
;  The software supplied herewith by Microchip Technology Incorporated
;  (the "Company") is intended and supplied to you, the Companyís
;  customer, for use solely and exclusively with products manufactured
;  by the Company.
;
;  The software is owned by the Company and/or its supplier, and is
;  protected under applicable copyright laws. All rights are reserved.
;  Any use in violation of the foregoing restrictions may subject the
;  user to criminal sanctions under applicable laws, as well as to
;  civil liability for the breach of the terms and conditions of this
;  license.
;
;  THIS SOFTWARE IS PROVIDED IN AN "AS IS" CONDITION. NO WARRANTIES,
;  WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING, BUT NOT LIMITED
;  TO, IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
;  PARTICULAR PURPOSE APPLY TO THIS SOFTWARE. THE COMPANY SHALL NOT,
;  IN ANY CIRCUMSTANCES, BE LIABLE FOR SPECIAL, INCIDENTAL OR
;  CONSEQUENTIAL DAMAGES, FOR ANY REASON WHATSOEVER.
/////////////////////////////////////////////////////////////////////////////////////////
;*******************************************************************************
;
; MCP2122 Transmit Data Code
;    This code will generate the 16x clock and continuously transmit a
;    known byte value (bit stream).
;    The variables "BaudRate" and "TXValue" must first be selected and the program then
;    assembled and downloaded for these configurations to become effective.
;    After Reset of the PICmicro(r) MCU, the device baud rate will determine the
;    values to load into the UART and CCP1 (PWM mode)/Timer2
;    registers. The TXValue will be the value that is continuously transmitted.
;
;    Device:    PIC16F877A
;    Platform:  PICDEM™ 2 Plus
;    Device Frequency: Fosc = 14.7456MHz
;
;    PORTC is used to interface to the MCP2122 device. The signals on this
;    port are: TX, RX, 16XCLK, and RESET
;
;*******************************************************************************
; Revision History
;  1.0  06/10/04  Initial Release
;
```

**FIGURE A-2: MCP2122TX.ASM - PAGE 2**

```
;******************************************************************************
;
; PICDEM 2 Plus Requirements
;    Clock Frequency:  14.7456 MHz
;    UART Baud:    9600,     19200,     38400,     57600,     115200
;    ==>
;    16XCLK:     153600,    307200,    614400,    921600,    1843200
;
;    Instructions/bit (@ 115200 Baud) = 14.7456/1.8432 = 8 (80 instrctions/byte)
;
; PIC16F877 PORT Functions
;    PORTA
;      Function         ---   ---   NA    NA    NA    NA    NA    NA
;      TRIS Direction   ---   ---    O     O     O     O     O     O
;      Initial value    ---   ---    H     H     H     H     H     H
;
;    PORTB
;      Function         NA    NA    NA    NA    LCD3  LCD2  LCD1  LCD0
;      TRIS Direction    O     O     O     O     O     O     O     O
;      Initial value     H     H     H     H     H     H     H     H
;
;    PORTC
;      Function         RX    TX  RST213X NA    NA   CCP1   NA   NA
;      TRIS Direction    I     I     O     O     O     O     O     O
;      Initial value    ---   ---    H     H     H     H     H     H
;
;    PORTD
;      Function         NA    NA    NA    NA    NA    NA    NA    NA
;      TRIS Direction    O     O     O     O     O     O     O     O
;      Initial value     H     H     H     H     H     H     H     H
;
;    PORTE
;      Function         ---   ---   ---   ---   ---   NA    NA    NA
;      TRIS Direction   ---   ---   ---   ---   ---    O     O     O
;      Initial value    ---   ---   ---   ---   ---    H     H     H
;
;******************************************************************************
;
#define reset   H'00'   ;Reset vector
;******************************************************************************
;   Configuration Bits
    __CONFIG _CP_OFF & _PWRTE_ON & _HS_OSC & _WDT_OFF & _BODEN_OFF & _LVP_OFF
                     & _WRT_HALF & _CPD_OFF & _DEBUG_ON
    __IDLOCS H'0010'

; for PIC16F877   __CONFIG _CP_OFF & _PWRTE_ON & _HS_OSC & _WDT_OFF & _BODEN_OFF
                                   & _LVP_OFF & _WRT_ENABLE_ON & _CPD_OFF & _DEBUG_ON
```

**FIGURE A-3: MCP2122TX.ASM - PAGE 3**

```
;*******************************************************************************
;   PortA Bits
;
#define rxd       PORTC, 7        ; input, serial data from MCP2122
#define txd       PORTC, 6        ; output, serial data to MCP2122

#define RESET213x PORTC, 5        ; output, used to reset the MCP2122
                                  ;         high for normal operation,
                                  ;         low to RESET device
;
;*******************************************************************************
; Program Definitions
;*******************************************************************************
;
ddra    EQU     B'00000000'     ; Data Direction for PORTA (output port)
ddrb    EQU     B'00000000'     ; Data Direction for PORTB (output port)
ddrc    EQU     B'11000000'     ; Data Direction for PORTC (input/output port)
ddrd    EQU     B'00000000'     ; Data Direction for PORTD (output port)
ddre    EQU     B'00000000'     ; Data Direction for PORTE (output port)
;
cfgopt  equ     B'11001000'     ; option reg setup
;
;*******************************************************************************
;   Constants
;
;BaudRate  EQU   D'9600'         ;
;BaudRate  EQU   D'19200'        ;
;BaudRate  EQU   D'38400'        ;
;BaudRate  EQU   D'57600'        ;
BaudRate  EQU   D'115200'       ;


;TXValue            EQU     0xFF                ; Value to transmit
;TXValue            EQU     0x00                ;
;TXValue            EQU     0x7E                ;
;TXValue            EQU     0x81                ;
;TXValue            EQU     0xFE                ;
TXValue            EQU     0x01                ;
;TXValue            EQU     0xAA                ;
;TXValue            EQU     0x55                ;
;TXValue            EQU     0xC3                ;
;TXValue            EQU     0x3C                ;

;
; Host UART Data Rate/BRG Value (BRGH = 1)
;                         SPBRG Value
;    Baud Rate    @ 20MHz   @ 16MHz   @ 14MHz   @ 4MHz
;    9600          129       103       95        25
;    19200          64        51       47        12
;    57600          21        16       15       N.A.
;   115200          10         8        7       N.A.
;
;                        SPBRG Value
B9600at14xMHz      EQU    D'95'
B19200at14xMHz     EQU    D'47'
B38400at14xMHz     EQU    D'23'
B57600at14xMHz     EQU    D'15'
B115200at14xMHz    EQU    D'07'
;
```

**FIGURE A-4: MCP2122TX.ASM - PAGE 4**

```
; PWM Values to generate 16XCLK
;    (For the duty cycle register, the CCP1X and CCP1Y bits are set to "1")
;
;                        PR2 Value     CCPR1L Value
PR2B9600at14xMHz   EQU    D'23'     ; D'11'
PR2B19200at14xMHz  EQU    D'11'     ; D'05'
PR2B38400at14xMHz  EQU    D'05'     ; D'02'
PR2B57600at14xMHz  EQU    D'03'     ; D'01'
PR2B115200at14xMHz EQU    D'01'     ; D'00'
;
    if (BaudRate == D'9600')
UARTBaud           EQU     B9600at14xMHz       ;
PWMPR2Value        EQU     PR2B9600at14xMHz    ;
    endif
    if (BaudRate == D'19200')
UARTBaud           EQU     B19200at14xMHz      ;
PWMPR2Value        EQU     PR2B19200at14xMHz   ;
    endif
    if (BaudRate == D'38400')
UARTBaud           EQU     B38400at14xMHz      ;
PWMPR2Value        EQU     PR2B38400at14xMHz   ;
    endif
    if (BaudRate == D'57600')
UARTBaud           EQU     B57600at14xMHz      ;
PWMPR2Value        EQU     PR2B57600at14xMHz   ;
    endif
    if (BaudRate == D'115200')
UARTBaud           EQU     B115200at14xMHz     ;
PWMPR2Value        EQU     PR2B115200at14xMHz  ;
    endif

;******************************************************************************
;   Registers
;
    cblock    H'20'
        temp
    endc
;
;******************************************************************************
        org     H'00'         ; use 00h as reset vector
        goto    START
;
;******************************************************************************
;   Start Routine
;   The post-reset setup is done here
;******************************************************************************
START   clrf    STATUS                    ; Bank 0
        movlw   0xFF                      ; Force PORTB to display High when configured as Output
        movwf   PORTB                     ;
        bsf     STATUS, RP0               ; Bank 1
        movlw   ddra
        movwf   TRISA                     ; Configure PORTA
        movlw   ddrb
        movwf   TRISB                     ; Configure PORTB
        movlw   ddrc
        movwf   TRISC                     ; Configure PORTC
        movlw   ddrd
        movwf   TRISD                     ; Configure PORTD
        movlw   ddre
        movwf   TRISE                     ; Configure PORTE
        movlw   cfgopt                    ; setup option reg
        movwf   OPTION_REG
;
```

**FIGURE A-5: MCP2122TX.ASM - PAGE 5**

```
;*****************************************************************************
;   Set the Operational Baud Rate
;     Initialize UART and 16XCLK (TMR2 and CCP1 modules)
;
;*****************************************************************************
;
INITBAUD
        CLRF    STATUS                     ; Bank 0
        CLRF    CCP1CON                    ; CCP1 Module Off
        CLRF    T2CON                      ; Timer 2 Off, Prescaler = 1, Postscaler = 1
                                           ;
        BSF     STATUS, RP0                ; Bank 1
        MOVLW   UARTBaud                   ; Initialize UART
        MOVWF   SPBRG                      ;
        movlw   0x24                       ; BRGH = 1, 8-bit, TX Enabled, Async.
        movwf   TXSTA                      ;
                                           ;
        CLRF    STATUS                     ; Bank 0
        MOVLW   0x90                       ; Enable serial port, continuous receive
        MOVWF   RCSTA                      ;
                                           ; Initialize PWM Frequency and Duty Cycle
        CLRF    TMR2                       ; TMR2 = 00h
        MOVLW   0xFF
        MOVWF   CCP1CON                    ; PWM mode, CCP1X:CCP1Y = 11b
        MOVLW   (PWMPR2Value-1)/2          ;
        MOVWF   CCPR1L                     ; Load the PWM Duty Cycle value
        BSF     STATUS, RP0                ; Bank 1
        MOVLW   PWMPR2Value                ;
        MOVWF   PR2                        ; Load the PWM Period value
        CLRF    STATUS                     ; Bank 0
        BSF     T2CON, TMR2ON              ; Turn On Timer 2, so 16XCLK is generated
        CLRF    PORTB                      ; clear outputs (Display on LEDs)
;
;*****************************************************************************
;   Transmit Value Routine
;
;*****************************************************************************
;
TXData
        CLRF    STATUS                     ; NO, Bank 0
        BCF     RESET213x                  ; Force MCP2122 RESET pin Low (MCP2122 in RESET)
        NOP                                ; Hold MCP2122 in reset for some time
        NOP
        NOP
        NOP
        BSF     RESET213x                  ; Force MCP2122 RESET pin High (Normal Operation)
;
TXValueLP1
        bsf     STATUS, RP0                ; Bank 1
UARTWaitLP
        btfss   TXSTA, TRMT                ; check if UART ready to transmit
        goto    UARTWaitLP                 ; not ready, wait
        bcf     STATUS, RP0                ; Bank 0
        MOVLW   TXValue                    ; This is the value of the byte to send
        movwf   TXREG                      ; send the byte
        GOTO    TXValueLP1                 ;
;
LAST    NOP

    end
```

## APPENDIX B: SCREEN CAPTURE BIT TIMING

The following screen captures (from Figure B-1 to Figure B-5) show the bit time for a single data bit at the indicated baud rate. The circled 1/Δt time shows the measured baud rate.

**FIGURE B-1: BIT TIMING CAPTURE AT 9600 BAUD**

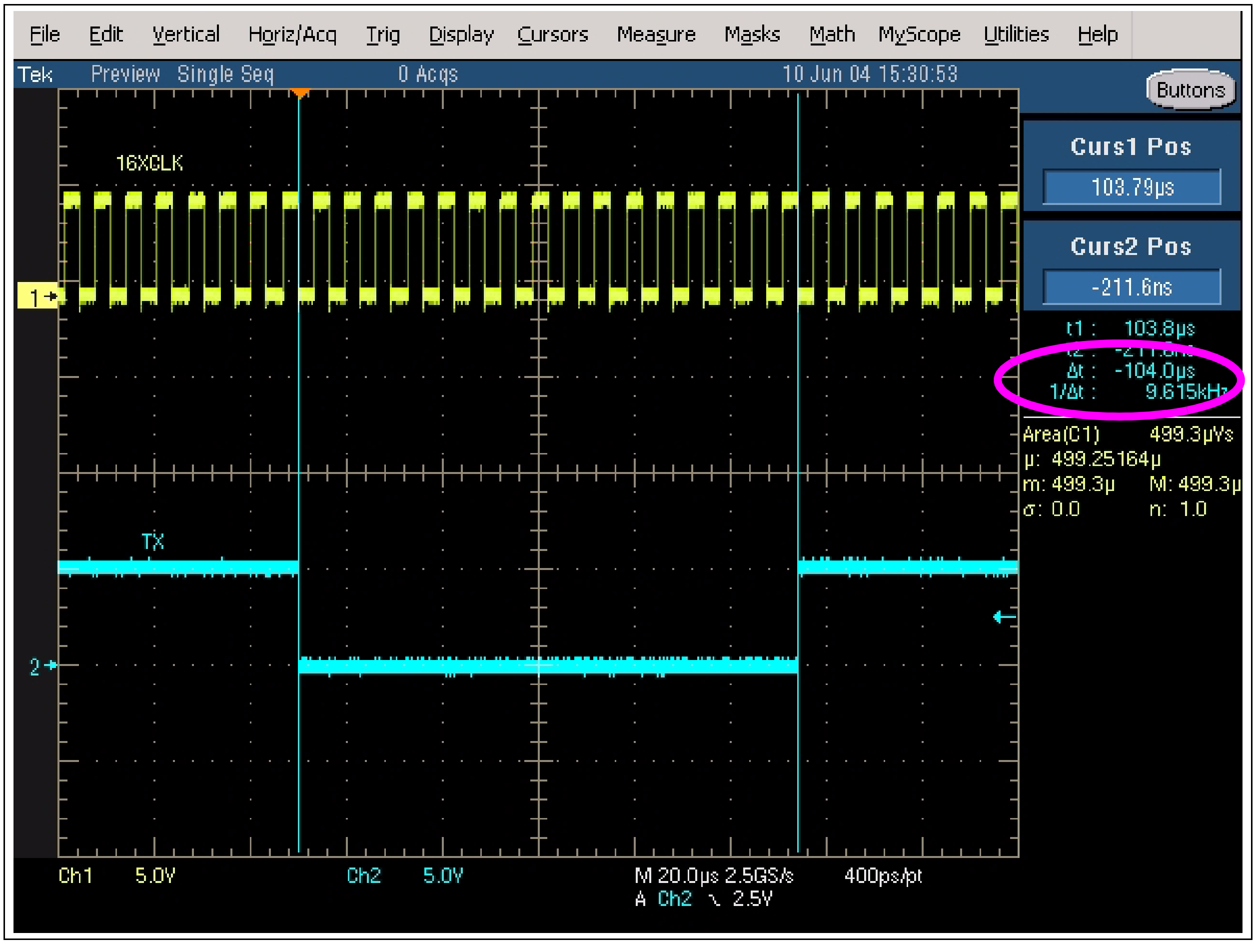


**FIGURE B-2: BIT TIMING CAPTURE AT 19200 BAUD**

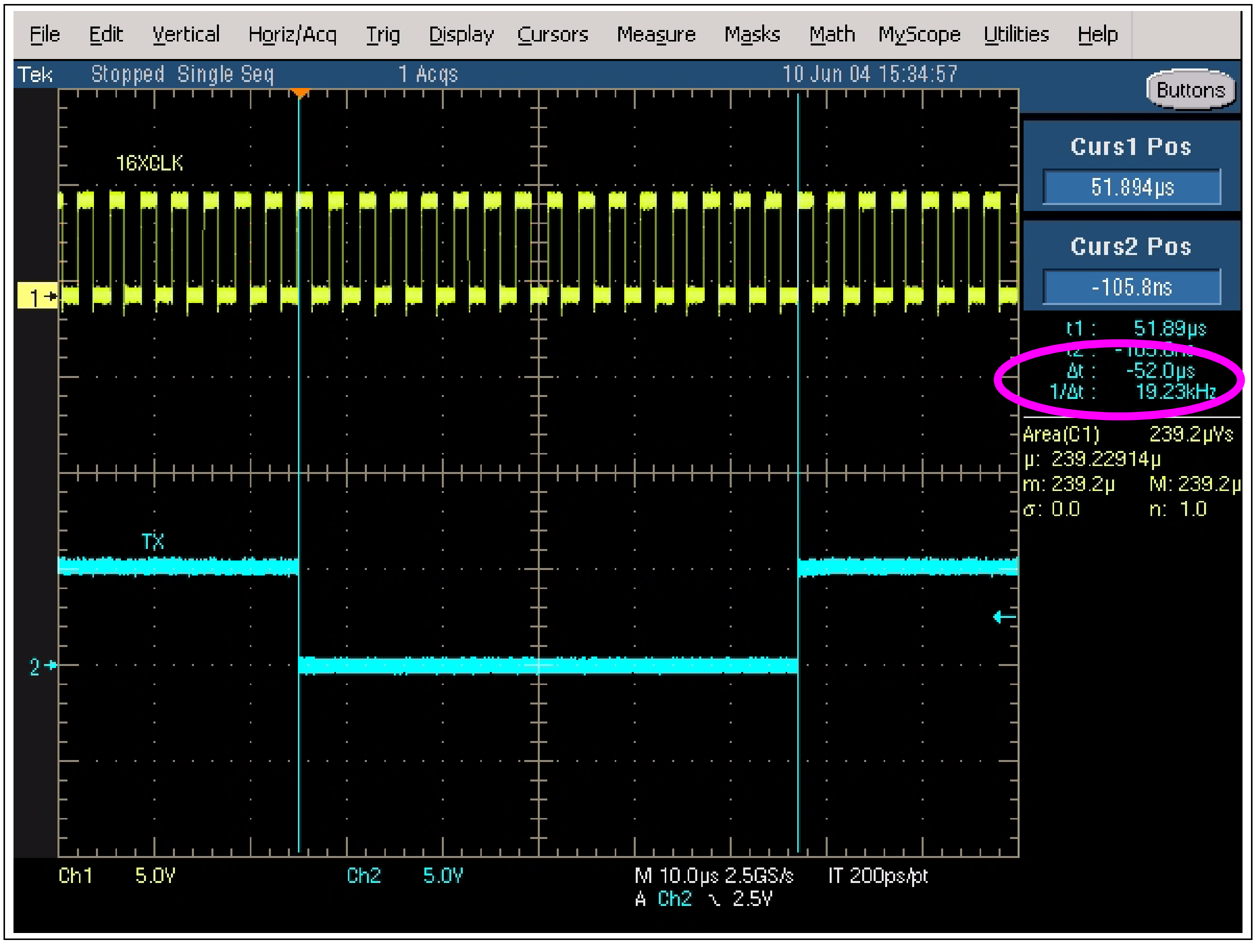


**FIGURE B-3: BIT TIMING CAPTURE AT 38400 BAUD**

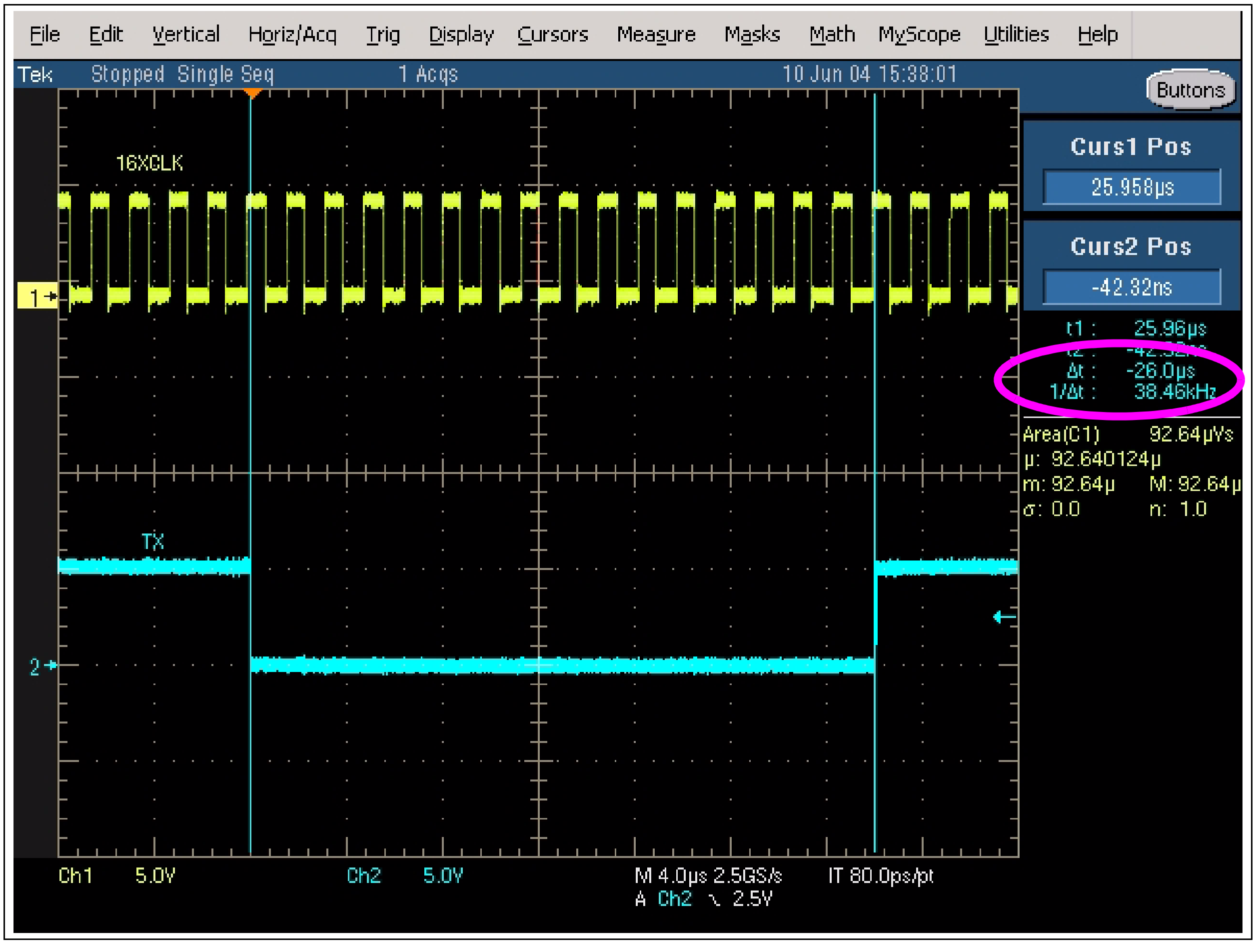


**FIGURE B-4: BIT TIMING CAPTURE AT 57600 BAUD**

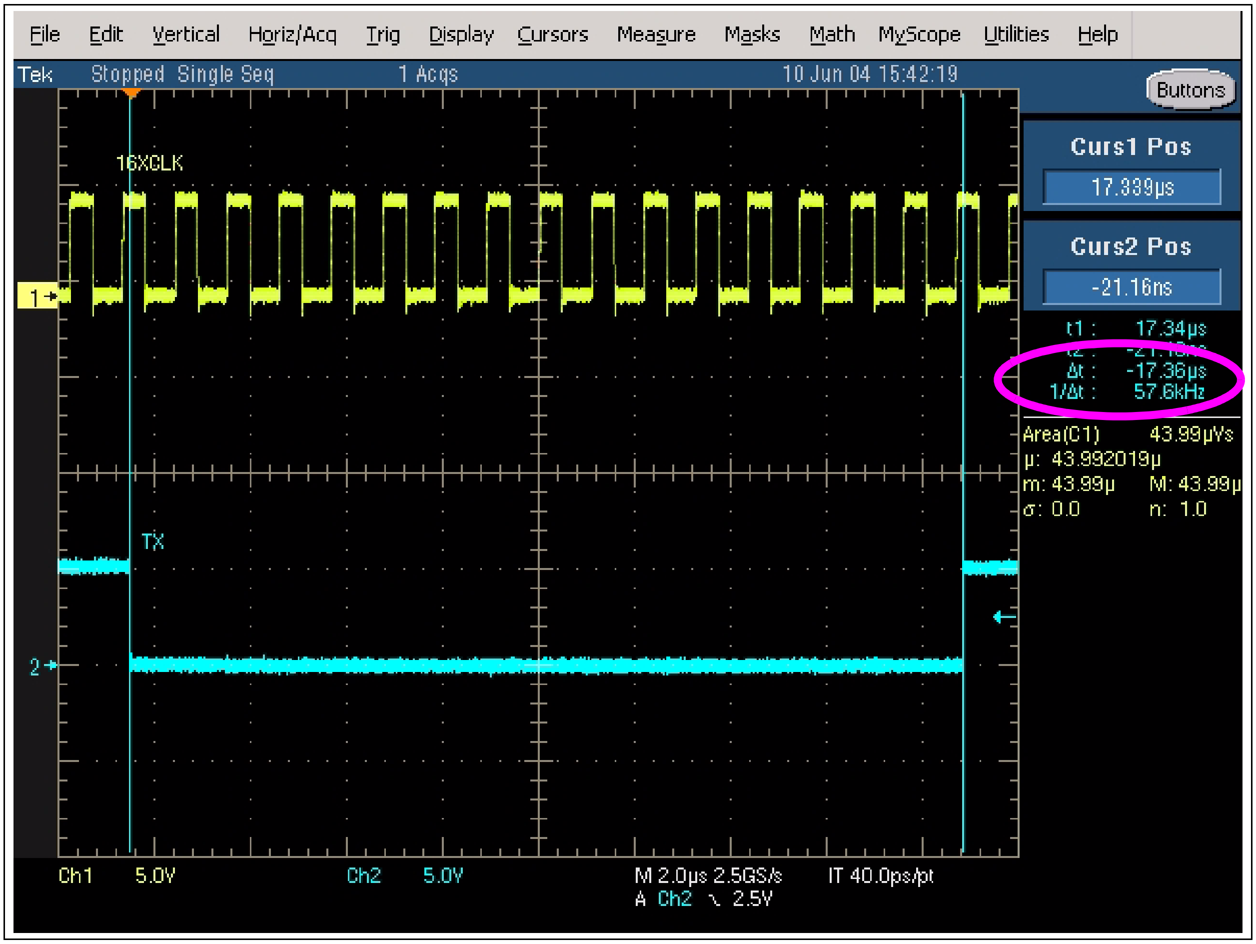
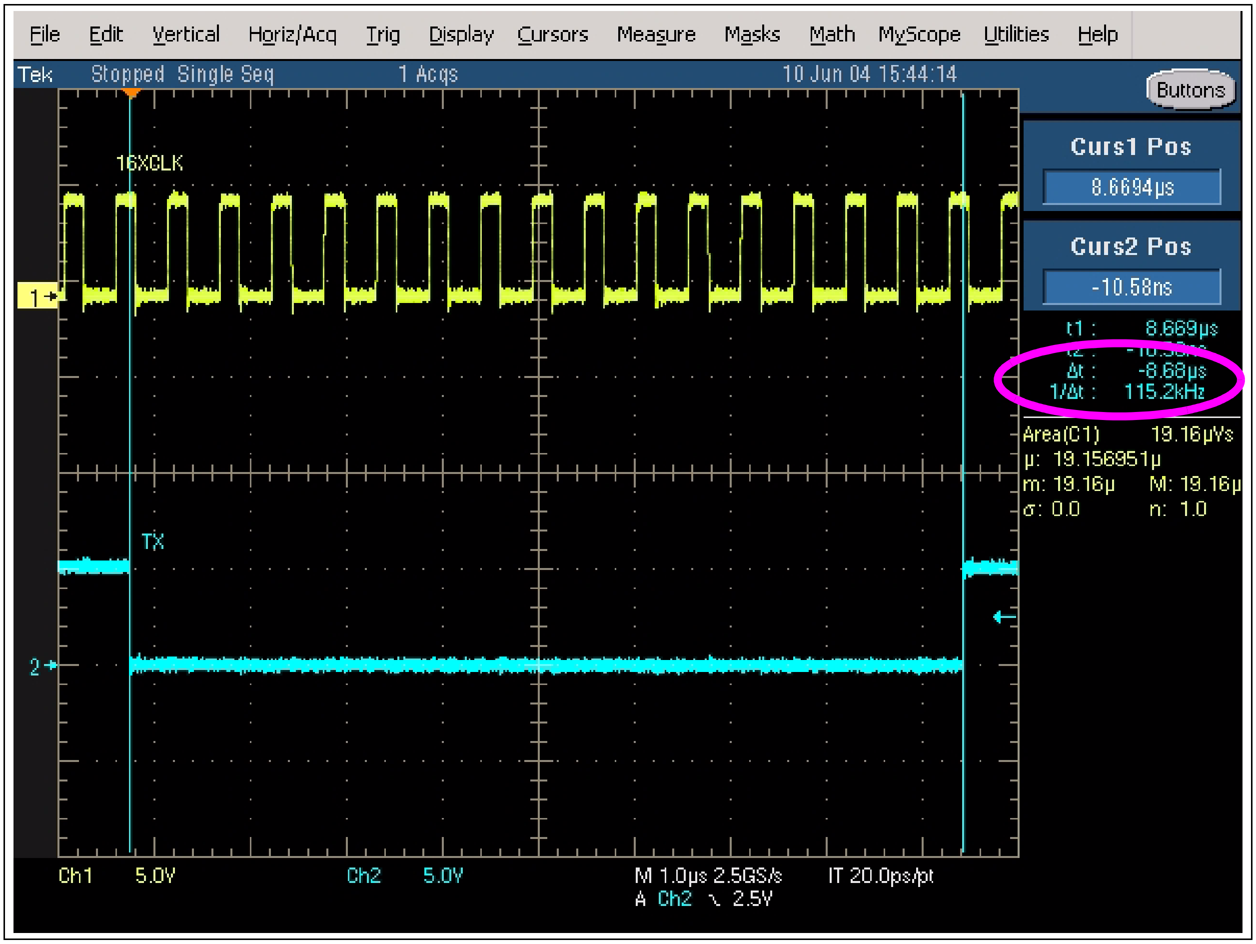


**FIGURE B-5: BIT TIMING CAPTURE AT 115200 BAUD**



**Note the following details of the code protection feature on Microchip devices:**

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Information contained in this publication regarding device applications and the like is intended through suggestion only and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. No representation or warranty is given and no liability is assumed by Microchip Technology Incorporated with respect to the accuracy or use of such information, or infringement of patents or other intellectual property rights arising from such use or otherwise. Use of Microchip's products as critical components in life support systems is not authorized except with express written approval by Microchip. No licenses are conveyed, implicitly or otherwise, under any intellectual property rights.

**Trademarks**

The Microchip name and logo, the Microchip logo, Accuron, dsPIC, KEELOQ, microID, MPLAB, PIC, PICmicro, PICSTART, PRO MATE, PowerSmart, rfPIC, and SmartShunt are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

AmpLab, FilterLab, MXDEV, MXLAB, PICMASTER, SEEVAL, SmartSensor and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Application Maestro, dsPICDEM, dsPICDEM.net, dsPICworks, ECAN, ECONOMONITOR, FanSense, FlexROM, fuzzyLAB, In-Circuit Serial Programming, ICSP, ICEPIC, Migratable Memory, MPASM, MPLIB, MPLINK, MPSIM, PICkit, PICDEM, PICDEM.net, PICLAB, PICtail, PowerCal, PowerInfo, PowerMate, PowerTool, rfLAB, rfPICDEM, Select Mode, Smart Serial, SmartTel and Total Endurance are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

All other trademarks mentioned herein are property of their respective companies.



Printed on recycled paper.

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
ISO/TS 16949:2002

*Microchip received ISO/TS-16949:2002 quality system certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona and Mountain View, California in October 2003. The Company's quality system processes and procedures are for its PICmicro® 8-bit MCUs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.*

MICROCHIP

# WORLDWIDE SALES AND SERVICE

## AMERICAS

**Corporate Office**
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support: 480-792-7627
Web Address: www.microchip.com

**Atlanta**
3780 Mansell Road, Suite 130
Alpharetta, GA 30022
Tel: 770-640-0034
Fax: 770-640-0307

**Boston**
2 Lan Drive, Suite 120
Westford, MA 01886
Tel: 978-692-3848
Fax: 978-692-3821

**Chicago**
333 Pierce Road, Suite 180
Itasca, IL 60143
Tel: 630-285-0071
Fax: 630-285-0075

**Dallas**
4570 Westgrove Drive, Suite 160
Addison, TX 75001
Tel: 972-818-7423
Fax: 972-818-2924

**Detroit**
Tri-Atria Office Building
32255 Northwestern Highway, Suite 190
Farmington Hills, MI 48334
Tel: 248-538-2250
Fax: 248-538-2260

**Kokomo**
2767 S. Albright Road
Kokomo, IN 46902
Tel: 765-864-8360
Fax: 765-864-8387

**Los Angeles**
18201 Von Karman, Suite 1090
Irvine, CA 92612
Tel: 949-263-1888
Fax: 949-263-1338

**San Jose**
1300 Terra Bella Avenue
Mountain View, CA 94043
Tel: 650-215-1444
Fax: 650-961-0286

**Toronto**
6285 Northam Drive, Suite 108
Mississauga, Ontario L4V 1X5, Canada
Tel: 905-673-0699
Fax: 905-673-6509

## ASIA/PACIFIC

**Australia**
Suite 22, 41 Rawson Street
Epping 2121, NSW
Australia
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

**China - Beijing**
Unit 706B
Wan Tai Bei Hai Bldg.
No. 6 Chaoyangmen Bei Str.
Beijing, 100027, China
Tel: 86-10-85282100
Fax: 86-10-85282104

**China - Chengdu**
Rm. 2401-2402, 24th Floor,
Ming Xing Financial Tower
No. 88 TIDU Street
Chengdu 610016, China
Tel: 86-28-86766200
Fax: 86-28-86766599

**China - Fuzhou**
Unit 28F, World Trade Plaza
No. 71 Wusi Road
Fuzhou 350001, China
Tel: 86-591-7503506
Fax: 86-591-7503521

**China - Hong Kong SAR**
Unit 901-6, Tower 2, Metroplaza
223 Hing Fong Road
Kwai Fong, N.T., Hong Kong
Tel: 852-2401-1200
Fax: 852-2401-3431

**China - Shanghai**
Room 701, Bldg. B
Far East International Plaza
No. 317 Xian Xia Road
Shanghai, 200051
Tel: 86-21-6275-5700
Fax: 86-21-6275-5060

**China - Shenzhen**
Rm. 1812, 18/F, Building A, United Plaza
No. 5022 Binhe Road, Futian District
Shenzhen 518033, China
Tel: 86-755-82901380
Fax: 86-755-8295-1393

**China - Shunde**
Room 401, Hongjian Building, No. 2
Fengxiangnan Road, Ronggui Town, Shunde District, Foshan City, Guangdong 528303, China
Tel: 86-757-28395507 Fax: 86-757-28395571

**China - Qingdao**
Rm. B505A, Fullhope Plaza,
No. 12 Hong Kong Central Rd.
Qingdao 266071, China
Tel: 86-532-5027355 Fax: 86-532-5027205

**India**
Divyasree Chambers
1 Floor, Wing A (A3/A4)
No. 11, O'Shaugnessey Road
Bangalore, 560 025, India
Tel: 91-80-22290061 Fax: 91-80-22290062

**Japan**
Benex S-1 6F
3-18-20, Shinyokohama
Kohoku-Ku, Yokohama-shi
Kanagawa, 222-0033, Japan
Tel: 81-45-471- 6166 Fax: 81-45-471-6122

**Korea**
168-1, Youngbo Bldg. 3 Floor
Samsung-Dong, Kangnam-Ku
Seoul, Korea 135-882
Tel: 82-2-554-7200 Fax: 82-2-558-5932 or 82-2-558-5934

**Singapore**
200 Middle Road
#07-02 Prime Centre
Singapore, 188980
Tel: 65-6334-8870 Fax: 65-6334-8850

**Taiwan**
Kaohsiung Branch
30F - 1 No. 8
Min Chuan 2nd Road
Kaohsiung 806, Taiwan
Tel: 886-7-536-4818
Fax: 886-7-536-4803

**Taiwan**
Taiwan Branch
11F-3, No. 207
Tung Hua North Road
Taipei, 105, Taiwan
Tel: 886-2-2717-7175 Fax: 886-2-2545-0139

## EUROPE

**Austria**
Durisolstrasse 2
A-4600 Wels
Austria
Tel: 43-7242-2244-399
Fax: 43-7242-2244-393

**Denmark**
Regus Business Centre
Lautrup hoj 1-3
Ballerup DK-2750 Denmark
Tel: 45-4420-9895 Fax: 45-4420-9910

**France**
Parc d'Activite du Moulin de Massy
43 Rue du Saule Trapu
Batiment A - ler Etage
91300 Massy, France
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

**Germany**
Steinheilstrasse 10
D-85737 Ismaning, Germany
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

**Italy**
Via Quasimodo, 12
20025 Legnano (MI)
Milan, Italy
Tel: 39-0331-742611
Fax: 39-0331-466781

**Netherlands**
Waegenburghtplein 4
NL-5152 JR, Drunen, Netherlands
Tel: 31-416-690399
Fax: 31-416-690340

**United Kingdom**
505 Eskdale Road
Winnersh Triangle
Wokingham
Berkshire, England RG41 5TU
Tel: 44-118-921-5869
Fax: 44-118-921-5820

05/28/04