**INTRODUCTION**

In this application note, a PIC10F322 is being used to implement a delay block/debouncer. The delay can be set between 2 µs and 193 µs. This can be used effectively as a noise discriminator, or for switch debouncing.

When used as a delay block, the application can be used to fix low-level timing issues on signals. When used as a debouncer, it can debounce signals from a mechanical switch so that a clean signal can feed other circuitry.

The application makes use of the Configurable Logic Cell (CLC) peripheral to produce fast switching on the output (if desired). If the same application were written using port logic only, there would be multiple instruction cycles before the output would change in response to an input. Using the CLC the signal can be routed directly and only have propagation and gate delay between the input and output signals. In order to get the highest performance possible from the application, it has been written in assembly.

The code has been written in-line (without subroutines) to maximize switching performance of the application. The code uses only 43 locations of program memory (512 available), and one byte of RAM.

This application note was developed using a PIC10F322 demo board (part # AC103011). Language tool versions: MPASMWIN.exe v5.45, mplink.exe v4.43, mplib.exe v4.43.

**DELAY SETTINGS AND VALUES**

Delays can be set between 2 µs and 193 µs in 750 ns increments (using the 16 MHz internal clock). Delays can easily be made longer if desired, by reducing the clock speed or increasing the size of the delay loop. Enabling/disabling of delays is configured in the code by commenting/uncommenting the following lines of code:

```assembly
; 2 lines below enable rising and falling edge delays
#define RISING_EDGE_DELAY
#define FALLING_EDGE_DELAY
```

Delays are set in milliseconds.

**CALCULATING DELAY VALUES**

There are two options for setting delay values, and they are determined by commenting/uncommenting the `MS_DELAY` definition:

1. 1 millisecond step size – this is useful for switch debouncing with recommended settings of `FALLING_EDGE_DELAY = 100 ms`, and `RISING_EDGE_DELAY` disabled.

2. 750 ns step size

With the part running at 16 MHz, there are 250 ns per instruction cycle. The delay loop takes three instruction cycles to execute, so 750 ns are added for each incremental change in the countdown timer. There are approximately eight instructions (2 µs) that will be executed between an input change and an output change, if the shortest possible delay is selected (`RISING_EDGE_DELAY` or `FALLING_EDGE_DELAY = 1`).

```assembly
delay = 2 µs + (delay_value x 750 ns)
```

Operating current: ~ 2.4 mA (but can be reduced with lower clock speed).

**MODES OF OPERATION**

**Rising Edge Delay**

In this mode, only the rising edge has a delay, and the falling edge will drop immediately.
Falling Edge Delay (Pulse Extender)

In this mode, the rising edge will come up immediately, and the falling edge will be delayed.

Rising and Falling Edge Delay

In this mode, the user has the option to set independent rising and falling edge values.

The diagram below (Figure 1) shows the relationship between the rising and falling edges. Signal edges are marked for reference in the source assembly code (delay.asm).

FIGURE 1: MEASUREMENT OF RISING AND FALLING EDGE DELAYS
When falling edge delay is not selected, the signal is routed through the CLC and has approximately 50 ns of propagation delay. Figure 2 shows the input and output signals.

Figure 2: Propagation Delay

With the CLC block configured as a pass-through, it is possible to quickly route signals to the output when no delay is desired, and the PIC® device core (port function) will create edge delays when desired. The MUX (CLC1CON, LC1OE) selects whether the pin is driven by the CLC or by the port logic (Figure 3).

Figure 3: Block Diagram – Mux Between Port and CLC Logic Block
CONFIGURING THE CLC BLOCK WITH THE CLC DESIGNER TOOL

The PIC10F322 has only one CLC block. The CLC block is configured as a pass-through by using the ‘AND’ logic tab of the CLC Designer tool. Unused ‘OR’ gates are inverted, so that they will produce a ‘1’ at the output. The configuration is shown in the screen shot below (Figure 4).

FIGURE 4: CONFIGURATION OF THE CLC BLOCK (CLC1)

RISING EDGE DELAY

Rising edge delay is enabled, while the falling edge is provided through the CLC block (Figure 5).

FIGURE 5: RISING EDGE DELAY EXAMPLE
FALLING EDGE DELAY

Falling edge delay is enabled, while the rising edge is provided through the CLC block (Figure 6).

FIGURE 6: FALLING EDGE DELAY EXAMPLE

RISING AND FALLING EDGE DELAY

FIGURE 7: RISING AND FALLING EDGE DELAY EXAMPLE

Signal delay specified in milliseconds:

FIGURE 8: RISING AND FALLING EDGE DELAY WITH MS_DELAY ENABLED
INVERTING THE OUTPUT SIGNAL

The CLC output can be inverted with the CLC tool by clicking on the buffer output to enable the inverter (as shown circled in red in Figure 9 below).

FIGURE 9: INVERTING THE OUTPUT SIGNAL

The example below (Figure 10) has no delay enabled, so the input signal is routed through the CLC block.

FIGURE 10: SIGNAL NOT INVERTED, ZERO DELAY
FIGURE 11: SIGNAL INVERTED, ZERO DELAY
Software License Agreement

The software supplied herewith by Microchip Technology Incorporated (the “Company”) is intended and supplied to you, the Company’s 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.

; File: CLC_pass_through.inc
; Generated by CLC Designer, Version: 1.0.0.3
; Date: 6/4/2012 11:49 AM
; Device: PIC10(L)F320/2
; BANKSEL CLC1GLS0    - this instruction not needed for PIC10LF322
   movlw  H'02'
   movwf  CLC1GLS0
   movlw  H'00'
   movwf  CLC1GLS1
   movlw  H'00'
   movwf  CLC1GLS2
   movlw  H'00'
   movwf  CLC1GLS3
   movlw  H'02'
   movwfc  CLC1SEL0
   movlw  H'65'
   movwf  CLC1SEL1
   movlw  H'0E'
   movwfc  CLC1POL
   movlw  H'C2'
   movwf  CLC1CON
APPENDIX B: DELAY.ASM

Software License Agreement
The software supplied herewith by Microchip Technology Incorporated (the “Company”) is intended and supplied to you, the Company's 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.

; /**************************************************************************
; * FileName:delay.asm
; * Dependencies:
; * Processor:PIC10F322
; * Hardware:
; * Compiler:MPASM 5.45 or later
; * Company:Microchip Technology, Inc.
; *
; * Copyright © 2007-2012 Microchip Technology Inc. All rights reserved.
; *
; * You may use this software, and any derivatives, exclusively with Microchip’s products.
; * Microchip and its licensors retain all ownership and intellectual property rights in
; * the accompanying software and in all derivatives hereto. This software and any
; * accompanying information is for suggestion only. It does not modify Microchip’s
; * standard warranty for its products. You agree that you are solely responsible for
; * testing the software and determining its suitability. Microchip has no obligation
; * to modify, test, certify, or support the software.
; *
; * THIS SOFTWARE IS SUPPLIED BY MICROCHIP "AS IS". NO WARRANTIES, WHETHER EXPRESS,
; * IMPLIED OR STATUTORY, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF NON-
; * INFRINGEMENT,
; * MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE APPLY TO THIS SOFTWARE, ITS
; * INTERACTION WITH MICROCHIP’S PRODUCTS, COMBINATION WITH ANY OTHER PRODUCTS, OR USE IN
; * ANY APPLICATION.
; *
; * IN NO EVENT, WILL MICROCHIP BE LIABLE, WHETHER IN CONTRACT, WARRANTY, TORT
; * (INCLUDING NEGLIGENCE OR BREACH OF STATUTORY DUTY), STRICT LIABILITY, INDEMNITY,
; * CONTRIBUTION, OR OTHERWISE, FOR ANY INDIRECT, SPECIAL, PUNITIVE, EXEMPLARY, INCIDENTAL
; * OR CONSEQUENTIAL LOSS, DAMAGE, FOR COST OR EXPENSE OF ANY KIND WHATSOEVER RELATED TO
; * THE SOFTWARE, HOWSOEVER CAUSED, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY
; * OR THE DAMAGES ARE FORESEEABLE. TO THE FULLEST EXTENT ALLOWABLE BY LAW, MICROCHIP'S
; * TOTAL LIABILITY ON ALL CLAIMS IN ANY WAY RELATED TO THIS SOFTWARE WILL NOT EXCEED THE
; * AMOUNT OF FEES, IF ANY, THAT YOU HAVE PAID DIRECTLY TO MICROCHIP FOR THIS SOFTWARE.
; *
; * MICROCHIP PROVIDES THIS SOFTWARE CONDITIONALLY UPON YOUR ACCEPTANCE OF THESE TERMS.
; *
; */
#include <p10f322.inc>

list p=p10f322

;; Define parameters here

;; 2 lines below enable rising and falling edge delays
#define RISING_EDGE_DELAY
#define FALLING_EDGE_DELAY

;; specify length of falling and rising edge below
RISE_EDGE_DELAY   equ   0x05
FALL_EDGE_DELAY   equ   0x04

#define MS_DELAY

;; Configuration Fuses

__config _FOSC_INTOSC & _BOREN_OFF & _WDTE_OFF & _PWRT_OFF & _MCLRE_OFF & _CP_OFF
 & _WRT_OFF

;; pin-out
;; 1 - RA0 - pin available for use as scope trigger.
;; 2 - VSS
;; 3 - RA1 - Data Out / CLC1
;; 4 - RA2 - Data In / CLC1IN2
;; 5 - VDD
;; 6 - MCLR

#define input_signal PORTA,2
#define output_signal LATA,1

; single RAM location used for countdown_timer
countdown_timer  equ   0x51

;; additional RAM location required if millisecond delay is enabled.
#ifdef MS_DELAY
ms_timer        equ   0x52 ; delay counter for specifying time delay in milliseconds.
#endif

org   0x00

start

bcf      TRISA,0 ; RA0 output - this can be used as a scope trigger.

bcf      TRISA,1 ; RA1 output
clrf     ANSELA ; all pins are digital.

movlw    0x70
movwf     OSCCON ; 16 MHz clock - change this value for longer delay times

; and to lower current consumption.

;; CLC is set up here with the following include:
#include "CLC_pass_through.inc" ; Configure CLC for falling edge.

; bsf CLC1POL,LC1POL; option to invert signal through CLC block.

;; input level test

    btfss input_signal ; What is the current value of my input signal?
    goto signal_low ; Low.
    goto signal_high ; High.

;; ===== This is the start of my main loop

signal_high ; Signal just transitioned high.

#ifdef RISING_EDGE_DELAY ; If I have rising edge delay,
    movlw RISE_EDGE_DELAY
    movwf countdown_timer ; load countdown timer with rising edge delay.
    rising_edge_delay_loop ; insert 'nop's' below this line to increase delay.

    #ifdef MS_DELAY ; option for millisecond delay
    call millisecond_delay
    #endif

    #ifdef MS_DELAY
    decfsz countdown_timer,1 ; Has countdown timer expired?
    goto rising_edge_delay_loop; no, continue delay loop
    #endif

    bsf output_signal ; and drive pin high.
#endif

#ifdef FALLING_EDGE_DELAY
    bsf output_signal ; drive latch high
    bcf CLC1CON,LC1OE ; PORT -> pin
#endif

#ifndef FALLING_EDGE_DELAY
    bsf CLC1CON,LC1OE ; CLC -> pin
#endif

wait_for_falling_edge
    btfsc input_signal
    goto wait_for_falling_edge

signal_low ; Signal just transitioned low.

#ifdef FALLING_EDGE_DELAY
    movlw FALL_EDGE_DELAY
    movwf countdown_timer ; load countdown timer with falling edge delay.
#endif

falling_edge_delay_loop ; insert 'nop's' below this line to increase delay.

#ifdef MS_DELAY ; option for millisecond delay
    call millisecond_delay
#endif
ifndef MS_DELAY
  decfsz countdown_timer,1 ; Has countdown timer expired?
goto falling_edge_delay_loop ; no, continue delay loop
#endif

bcf output_signal ; and drive pin low.

#ifdef RISING_EDGE_DELAY
  bcf output_signal ; drive latch low
  bcf CLC1CON,LC1OE ; PORT -> pin
#endif

#ifdef RISING_EDGE_DELAY
  bsf CLC1CON,LC1OE ; CLC -> pin
#endif

wait_for_rising_edge
  btfss input_signal
  goto wait_for_rising_edge
  goto signal_high

;;;;;;; end of main loop

#ifndef MS_DELAY

millisecond_delay
  movlw .250 ; 250 loops x 16 cycles per loop = 4000 Tcy = 1 ms
  movwf ms_timer

millisecond_delay_loop
  nop
  nop
  nop
  nop
  nop
  nop
  nop
  nop
  nop
  nop
  nop
  nop
  nop
  decfsz ms_timer ; has countdown timer reached 0?
goto millisecond_delay_loop; No. continue looping
  decfsz countdown_timer ; Do I need to have another millisecond delay?
goto millisecond_delay ; Yes.
  retlw 0x00 ; No - done.
#endif

end
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 provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer’s risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights.

Trademarks
The Microchip name and logo, the Microchip logo, dsPIC, KEELog, KEELog logo, MPLAB, PIC, PICmicro, PICSTART, PIC³2 logo, rPIC and Uni/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.
FilterLab, Hampshire, HI-TECH C, Linear Active Thermistor, MXDEV, MXLAB, SEEVAL 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, chipKIT, chipKIT logo, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, dsSPEAK, ECAN, ECONOMONITOR, FanSense, HI-TIDE, In-Circuit Serial Programming, ICSP, Mindi, MiWi, MPASM, MPLAB Certified logo, MPLIB, MPLINK, mTouch, Omniscient Code Generation, PICC, PICC-18, PICDEM, PICDEM.net, PICkit, Pictail, REAL ICE, rfLAB, Select Mode, Total Endurance, TSHARC, UniWinDriver, WiperLock and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.
SQT is a service mark of Microchip Technology Incorporated in the U.S.A.
All other trademarks mentioned herein are property of their respective companies.
© 2012, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.
Printed on recycled paper.
ISBN: 9781620764596

QUALITY MANAGEMENT SYSTEM CERTIFIED BY DNV
═ ISO/TS 16949 ═

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company’s quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, 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.