Frequency Counter Using PIC16C5X

INTRODUCTION

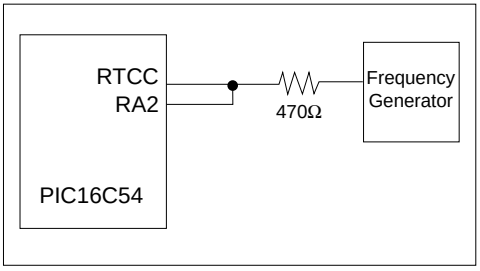
The PIC16C5X has one 8-bit timer (RTCC), which can be used with an 8-bit prescaler. The prescaler runs asynchronously, hence it can count a very high frequency. The minimum rise and fall times of the input frequency are specified to be 10nS, so the fastest clock rate the RTCC can count is 50 MHz. The prescaler must be used when measuring high frequency. Since the prescaler can be configured as a divide by 256 counter, the maximum resolution which the input frequency can be measured is 16 bits. However, the prescaler cannot be directly read like a file register. This application note depicts a unique method by which the user can "extract" the 8-bit value in the prescaler, whereby the resolution of the measurement is 16 bits with the high 8 bits in the RTCC and the low 8 bits in the prescaler.

IMPLEMENTATION

A frequency counter which can read frequencies from 50 MHz to 50 Hz was implemented in this application note to demonstrate this method of measuring the 16-bit counter value from the prescaler and RTCC.

The basic hardware for the measurement circuit is depicted in Figure 1. It consists of the frequency input at RTCC or RA4 (pin 3 in a PIC16C54). RA4 is connected to RA2. The input frequency is connected to RTCC through a 470 ohm resistor.

FIGURE 1



The RTCC is configured to measure the input frequency at RA4 of the PIC16C54. The input frequency is "gated" for a precise duration of time. Before starting this precise "gate", the RTCC is cleared (which also clears the prescaler), and the RA2 pin is configured as an input. The precise "gate" is implemented in software as an accurate delay. At the end of the delay, the RA2 pin is configured as an output going low. This will cause the input to the RTCC to be "halted" or "stopped". A 16-bit value of the input frequency is now saved in RTCC and the 8-bit prescaler. The high 8 bits are in RTCC and can be easily read. The low 8 bits have to be "shifted out". The 8-bits in the prescaler are "shifted out" by toggling RA2 with a "BSF" and "BCF" instruction. After every toggle, the value in RTCC is checked to see if the RTCC has incremented. If the number of toggles required to cause the RTCC to increment by 1 is N, then the 8-bit value in the pre-scaler can be calculated to be = (256 - N). By concatenating the calculated value and the original value in RTCC, the 16-bit value for the frequency is determined.

To measure a wide range of frequency, the following intermediate steps were taken:

Frequency Range	Precise "gate" Delay	Resolution
50 MHz - 10 MHz	1 ms	±10 KHz
10 MHz - 1 MHz	5 ms	±2 KHz
1 MHz - 100 KHz	50 ms	±200 Hz
100 KHz - 10 KHz	200 ms	±50 Hz
50 KHz - 50 Hz	50 ms†	±2 Hz

† In this case, the RTCC uses the internal 4 MHz clock and counts the number of instance of the external clock. Maximum Time required is 50 ms to make a ± 2 Hz accurate measurement for 10 KHz input frequency.

The check for the correct frequency is done automatically starting with the high frequency and going down to the low frequency. The maximum time required for each conversion is approximately 310 ms. In other words, three frequency checks are done every second.

CONCLUSION

The PIC16C5X family can be used to make a 16-bit measurement of input frequency with a small overhead of one resistor and one I/O port.

Author: Stan D'Souza
Logic Products Division

APPENDIX A

```
MPASM 00.00.66 Beta               6-16-1994  23:59:52                PAGE  1


LOC  OBJECT CODE     LINE SOURCE TEXT

                     0001 ;Title DISPLAY.ASM
                     0002 ;This file displays a binary value found in the display
                     0003 ;register “DisplayRegister”. The binary value is converted
                     0004 ;to BCD and then displayed.
                     0005         list p=16c71,f=inhx8m
                     0006         include “16cxx.h”
                     0179
                     0180
                     0181
                     0006
                     0007 ;
0001                 0008 TRUE    equ     1
0000                 0009 FALSE   equ     0
0000                 0010 FUZZY   equ     FALSE
                     0011         if FUZZY
                     0012         #define _ledEn  _portb,3
                     0013         #define _ledData _portb,1
                     0014         #define _ledClk _portb,2
                     0015         else
0046                 0016         #define _ledEn  _portb,3
0047                 0017         #define _ledData _portb,2
0048                 0018         #define _ledClk _portb,1
                     0019         endif
0011                 0020 HighFreq        equ     0x11
0012                 0021 LowFreq         equ     0x12
001A                 0022 acca    equ     1a
001B                 0023 accb    equ     1b
001C                 0024 accc    equ     1c
001D                 0025 accd    equ     1d
001E                 0026 acce    equ     1e
001F                 0027 accf    equ     1f
0009                 0028 time    equ     09
0010                 0029 temp    equ     10
                     0030 ;
                     0031 ;
                     0032         org     0
                     0033 start
0000 3004            0034         movlw   .4              ;initialize time
0001 0089            0035         movwf   time            ;       /
0002 019B            0036         clrf    accb            ;
0003 0186            0037         clrf    _portb          ;init ports
0004 0185            0038         clrf    _porta          ;       /
0005 1586            0039         bsf     _ledEn          ;disallow writes to display
0006 3017            0040         movlw   B’00010111'     ;RA3 as output, rest as inputs
0007 3070            0042         movlw   B’01110000'     ;RB4-6 as inputs rest outputs
0008 3087            0044         movlw   B’10000111'     ;ps with RTCC for Tcyl/256
0009 0062            0045         option                  ;       /
000A 0181            0046         clrf    _rtcc           ;start timer
                     0047 wait
000B 202D            0048         call    Display         ;display on leds
000C 280B            0049         goto    wait
                     0050 ;
                     0051
                          ;*****************************************************************
                     0052 ;       This subroutine converts a 8 bit binary word
                     0053 ;       into a 3 digit BCD
                     0054 ;       The input is in accb
                     0055 ;       output is in accc and accd with  lsd in ACCD.
                     0056 ;       The basic idea is that a 8 bit binary # has a value
                     0057 ;       between 0 and 255. First we check if the # is > 99
                     0058 ;       then if it is > 199. After each check we inc the MSD
                     0059 ;       Lastly we convert the LSD which will have a value
                     0060 ;       between 0 and 99.
```

```
                    0061 ;
                    0062 ;*****************************************************************
                    0063 ;
                    0064 Bin8toBcd3
000D 3002           0065         movlw   2
000E 009C           0066         movwf   accc
000F 0190           0067         clrf    temp
0010 30C7           0068         movlw   .199       ;check if # is > 199
0011 021B           0069         subwf   accb,w     ;       /
0012 1903           0070         btfsc   _z         ;= 199?
0013 2816           0071         goto    Bcd99B
0014 1803           0072         btfsc   _c         ;       /
0015 2821           0073         goto    Bcd199     ;yes then do >200 #
                    0074 Bcd99B
0016 039C           0075         decf    accc       ;else inc Msd of BCD
0017 3063           0076         movlw   .99        ;and see > 99
0018 021B           0077         subwf   accb,w     ;       /
0019 1903           0078         btfsc   _z         ; == 99?
001A 281D           0079         goto    Bcd99A     ;yes then skip over
001B 1803           0080         btfsc   _c         ;       /
001C 2821           0081         goto    Bcd199     ;no then do 99
                    0082 Bcd99A
001D 039C           0083         decf    accc
                    0084 Bcd99
001E 081B           0085         movf    accb,w
001F 009D           0086         movwf   accd
0020 2823           0087         goto    get10th
                    0088 Bcd199
0021 009D           0089         movwf   accd       ;get result in ACCD
0022 039D           0090         decf    accd       ;dec to get correct value
                    0091 get10th
0023 300A           0092         movlw   .10
0024 021D           0093         subwf   accd,w     ;reduce by 10
0025 1C03           0094         btfss   _c         ;see if done
0026 282A           0095         goto    BcdOver    ;yes then end
0027 009D           0096         movwf   accd       ;get new value in ACCD
0028 0A90           0097         incf    temp       ;inc 10s count
0029 2823           0098         goto    get10th    ;do next
                    0099 BcdOver
002A 0E10           0100         swapf   temp,w     ;get in w
002B 049D           0101         iorwf   accd       ;or with 1s
002C 0008           0102         return
                    0103 ;
                    0104 ;*********************************************************
                    0105 ;       This routine displays 3 digits on a LT8522 display.
                    0106 ;       Three wires are required to drive the display
                    0107 ;       Enable –> active low when writing to display
                    0108 ;       Clock  –> 1 start followed by 35 more (36 total)
                    0109 ;                 36 clock required for load to occur.
                    0110 ;                 Rising edge of Clock is active.
                    0111 ;       Data   –> start data bit = high;
                    0112 ;                 1st data bit –> segment A of MSD
                    0113 ;                 2nd data bit –> segment B of MSD
                    0114 ;                 so on...
                    0115 ;                 8th data bit –> d.p. of MSD
                    0116 ;                 9th data bit –> segment A of 2nd digit
                    0117 ;                 10th data bit –> segment B of 2nd digit
                    0118 ;                 so on...
                    0119 ;                 16th data bit –> d.p. of 2nd digit
                    0120 ;                 17th data bit –> segment A of LSD
                    0121 ;                 18th data bit –> segment B of LSD
                    0122 ;                 so on ...
                    0123 ;                 24th data bit –> d.p. of LSD
                    0124 ;                 25th data bit –> appears on pin 4 of display
                    0125 ;                 26th data bit –> appears on pin 5 of display
                    0126 ;                 so on ...
                    0127 ;                 34th data bit –> appears on pin 13 of display.
                    0128 ;                 to dirve segment set data = high.
```



```
                  0129 ;
                  0130 ;       The routine does a leading zero blanking.
                  0131 ;       The 3 BCD nibbles should be available in accc and accd,
                  0132 ;       with the MSD in the low nibble of accc.
                  0133 ;*****************************************************************
                  0134 Display
002D 205E         0135         call    StartDisplay
002E 0E11         0136         swapf   HighFreq,w
002F 390F         0137         andlw   0x0f
0030 2040         0138         call    LedValue
0031 2052         0139         call    DisplayW
0032 0811         0140         movf    HighFreq,w
0033 390F         0141         andlw   0x0f
0034 2040         0142         call    LedValue
0035 2052         0143         call    DisplayW
0036 0E12         0144         swapf   LowFreq,w
0037 390F         0145         andlw   0x0f
0038 2040         0146         call    LedValue
0039 2052         0147         call    DisplayW
003A 0812         0148         movf    LowFreq,w
003B 390F         0149         andlw   0x0f
003C 2040         0150         call    LedValue
003D 2052         0151         call    DisplayW
003E 2064         0152         call    EndDisplay
003F 3400         0153         retlw   0
                  0154 ;
                  0155 ;
                  0156 ;
                  0157 LedValueAddress
                  0158         if LedValueAddress < 0x100
                  0159 LedValue
0040 018A         0160         clrf    _pclath
0041 0782         0161         addwf   _pcl
0042 34FC         0162         retlw   0xfc    ;code for 0
0043 3460         0163         retlw   0x60    ;code for 1
0044 34DA         0164         retlw   0xda    ;code for 2
0045 34F2         0165         retlw   0xf2    ;code for 3
0046 3466         0166         retlw   0x66    ;code for 4
0047 34B6         0167         retlw   0xb6    ;code for 5
0048 34BE         0168         retlw   0xbe    ;code for 6
0049 34E0         0169         retlw   0xe0    ;code for 7
004A 34FE         0170         retlw   0xfe    ;code for 8
004B 34E6         0171         retlw   0xe6    ;code for 9
004C 34EE         0172         retlw   0xee    ;code for A
004D 343E         0173         retlw   0x3e    ;code for b
004E 349C         0174         retlw   0x9c    ;code for C
004F 347A         0175         retlw   0x7a    ;code for d
0050 349E         0176         retlw   0x9e    ;code for E
0051 348E         0177         retlw   0x8e    ;code for F
                  0178         endif
                  0179 ;
                  0180 ;
                  0181 DisplayW
0052 0090         0182         movwf   temp
0053 3008         0183         movlw   .8
0054 009A         0184         movwf   acca
                  0185 DisplayLoop
0055 0D90         0186         rlf     temp
0056 1803         0187         btfsc   _c
0057 1506         0188         bsf     _ledData
0058 1486         0189         bsf     _ledClk
0059 1086         0190         bcf     _ledClk
005A 1106         0191         bcf     _ledData
005B 0B9A         0192         decfsz  acca
005C 2855         0193         goto    DisplayLoop
005D 3400         0194         retlw   0
                  0195 ;
                  0196 ;
```

```
                     0197 StartDisplay
005E 1186            0198         bcf     _ledEn
005F 1506            0199         bsf     _ledData
0060 1486            0200         bsf     _ledClk
0061 1086            0201         bcf     _ledClk
0062 1106            0202         bcf     _ledData
0063 3400            0203         retlw   0
                     0204 ;
                     0205 EndDisplay
0064 1486            0206         bsf     _ledClk
0065 1086            0207         bcf     _ledClk
0066 1486            0208         bsf     _ledClk
0067 1086            0209         bcf     _ledClk
0068 1486            0210         bsf     _ledClk
0069 1086            0211         bcf     _ledClk
006A 1586            0212         bsf     _ledEn
006B 3400            0213         retlw   0
                     0214
                     0215 ;
                     0216         org     0x1ff
01FF 2800            0217         goto    start
                     0218
                     0219 ;
                     0220         end
                     0221 ;
                     0222
                     0223


MEMORY USAGE MAP ('X' = Used,  '-' = Unused)

0000 : XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX
0040 : XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXX-- --------

0180 : -------- -------- -------- --------
01C0 : -------- -------- -------- -------X

All other memory blocks unused.


Errors   :    0
Warnings :    0
```


NOTES:

WORLDWIDE SALES & SERVICE

AMERICAS

Corporate Office
Microchip Technology Inc.
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 602 786-7200 Fax: 602 786-7277
Technical Support: 602 786-7627
Web: http://www.mchip.com/microhip

Atlanta
Microchip Technology Inc.
500 Sugar Mill Road, Suite 200B
Atlanta, GA 30350
Tel: 770 640-0034 Fax: 770 640-0307

Boston
Microchip Technology Inc.
5 Mount Royal Avenue
Marlborough, MA 01752
Tel: 508 480-9990 Fax: 508 480-8575

Chicago
Microchip Technology Inc.
333 Pierce Road, Suite 180
Itasca, IL 60143
Tel: 708 285-0071 Fax: 708 285-0075

Dallas
Microchip Technology Inc.
14651 Dallas Parkway, Suite 816
Dallas, TX 75240-8809
Tel: 214 991-7177 Fax: 214 991-8588

Dayton
Microchip Technology Inc.
35 Rockridge Road
Englewood, OH 45322
Tel: 513 832-2543 Fax: 513 832-2841

Los Angeles
Microchip Technology Inc.
18201 Von Karman, Suite 455
Irvine, CA 92715
Tel: 714 263-1888 Fax: 714 263-1338

New York
Microchip Technology Inc.
150 Motor Parkway, Suite 416
Hauppauge, NY 11788
Tel: 516 273-5305 Fax: 516 273-5335

AMERICAS (continued)

San Jose
Microchip Technology Inc.
2107 North First Street, Suite 590
San Jose, CA 95131
Tel: 408 436-7950 Fax: 408 436-7955

ASIA/PACIFIC

Hong Kong
Microchip Technology
Unit No. 3002-3004, Tower 1
Metroplaza
223 Hing Fong Road
Kwai Fong, N.T. Hong Kong
Tel: 852 2 401 1200 Fax: 852 2 401 3431

Korea
Microchip Technology
168-1, Youngbo Bldg. 3 Floor
Samsung-Dong, Kangnam-Ku,
Seoul, Korea
Tel: 82 2 554 7200 Fax: 82 2 558 5934

Singapore
Microchip Technology
200 Middle Road
#10-03 Prime Centre
Singapore 188980
Tel: 65 334 8870 Fax: 65 334 8850

Taiwan
Microchip Technology
10F-1C 207
Tung Hua North Road
Taipei, Taiwan, ROC
Tel: 886 2 717 7175 Fax: 886 2 545 0139

EUROPE

United Kingdom
Arizona Microchip Technology Ltd.
Unit 6, The Courtyard
Meadow Bank, Furlong Road
Bourne End, Buckinghamshire SL8 5AJ
Tel: 44 0 1628 851077 Fax: 44 0 1628 850259

France
Arizona Microchip Technology SARL
2 Rue du Buisson aux Fraises
91300 Massy - France
Tel: 33 1 69 53 63 20 Fax: 33 1 69 30 90 79

Germany
Arizona Microchip Technology GmbH
Gustav-Heinemann-Ring 125
D-81739 Muenchen, Germany
Tel: 49 89 627 144 0 Fax: 49 89 627 144 44

Italy
Arizona Microchip Technology SRL
Centro Direzionale Colleoni
Palazzo Pegaso Ingresso No. 2
Via Paracelso 23, 20041
Agrate Brianza (MI) Italy
Tel: 39 039 689 9939 Fax: 39 039 689 9883

JAPAN

Microchip Technology Intl. Inc.
Benex S-1 6F
3-18-20, Shin Yokohama
Kohoku-Ku, Yokohama
Kanagawa 222 Japan
Tel: 81 45 471 6166 Fax: 81 45 471 6122

9/22/95