TB004

Automatic Calibration of the WDT Time-out Period

Author: Stan D'Souza
Advanced Microcontroller
Technology Division

INTRODUCTION

The WDT timer is a simple RC timer with a typical time-out period of about 18 ms. This time-out period is dependent on Voltage, Temperature and Silicon process variations. Hence the tolerance on the time-out period is very wide: Min. of 9 ms to a Max. of 33 ms (please refer to appropriate datasheet for device dependent value). There are applications where an additional timer would be useful as an approximate time keeper, hence getting a more precise value of the WDT time-out is useful. This Tech Brief implements an automatic calibration of the WDT time-out period on start-up.

IMPLEMENTATION

The hardware used for this brief is the PICDEM1 board. It is assumed that the main processor oscillator is an accurate crystal or ceramic resonator running at 4 MHz. The program flowchart is depicted in Figure 1. A check is made to see if a certain **Codebyte** exists, if it does not then a power-up is assumed and the calibration is executed. Note that instead of using the 8-bit wide **CodeByte**, the $\overline{PD}$ bit in the **STATUS** register could also be used for this very same purpose. The calibration uses an internal timed 1 ms interval. To get a better resolution, the WDT is postscaled by 4 in order to measure a longer time period . Every 1 ms interval is kept track by incrementing the **WDTValue**. The WDT will eventually time-out and cause a reset. The value in **WDTValue** in then divided by 4 to get the exact WDT time-out period and displayed as a binary number on PORTB. In order to check for repeatability, just press the reset button on the PICDEM1 board and the whole calibration process will be repeated. A power-down, followed by a power-up will also have the same effect.

CONCLUSION

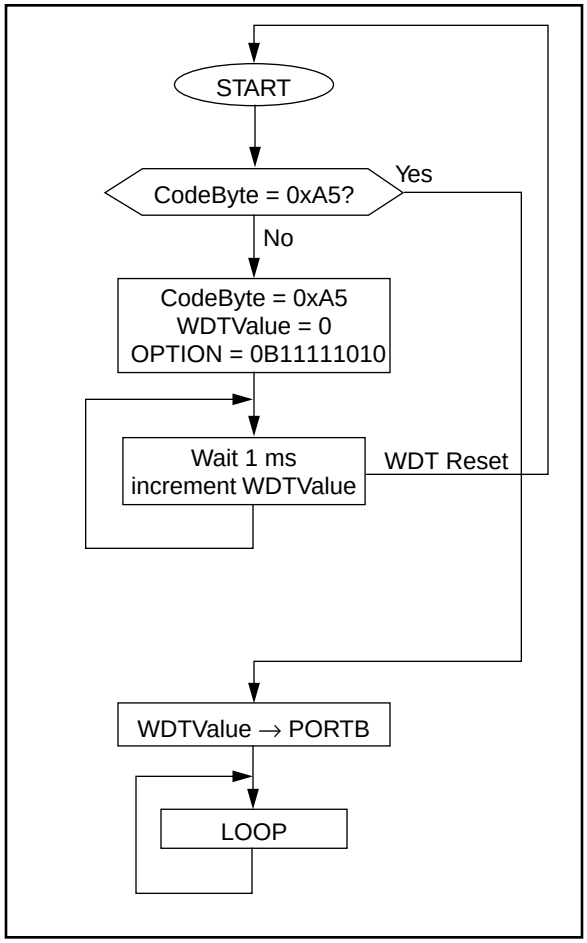
The calibration of the WDT is simple and takes very little overhead in a program. The code in Appendix A is written for a PIC16C84 device, but can be translated to work on any PIC16CXXX product. The code in Appendix B is written for the PIC16C5X family.

RAM Used: 2 Bytes

ROM Used: 50 Words

Execution time: Up to a max. of 132 ms from start-up.

FIGURE 1: PROGRAM FLOWCHART



APPENDIX A:

```
MPC "C" COMPILER BC.193 22-Aug-1995                                  PAGE  1


                                /* In this example, a PIC16C84s WDT is calibrated on startup
                                   for better accuracy.  Accuracy is exact to +/- 0.25mS.
                                   The code written in C works on a PICDEM1 board.  The WDT
                                   value is written to PORTB at the end of the calibration.
                                   The value is in binary. */

                                #pragma option v
                                #include <c:\mpc\16c84.h>
                                #pragma option +1;
03FB                            #define MAXROM 1019
0005 03FB                       #pragma memory ROM [MAXROM]     @ 0x05;
000C 0024                       #pragma memory RAM [36]         @ 0x0C;
                                #pragma option +1;

                                #include <c:\mpc\delay14.h>

                                #pragma option +l;

000D                            char WdtValue;
000E                            chr CodeByte;

                                void InitPorts(void)
                                {
0005 1283    BCF    STATUS,RP0      PORTB = 0; //configure PORTB as output
0006 0186    CLRF   PORTB
0007 1683    BSF    STATUS,RP0      TRISB = 0;
0008 0186    CLRF   PORTB
0009 0008    RETURN               }

                                void main(void)
                                {
000A 2005    CALL   0005h            InitPorts();
000B 30A5    MOVLW  A5h              if(CodeByte != 0xA5)
000C 1283    BCF    STATUS,RP0
000D 0212    SUBWF  TMR1L,W
000E 1903    BTFSC  STATUS,Z
000F 281E    GOTO   001Eh
0010                                   {
0010 1283    BCF    STATUS,RP0         WdtValue = 0;
0011 0191    CLRF   PIR2
0012 30A5    MOVLW  A5h                CodeByte = 0xA5;
0013 0092    MOVWF  TMR1L
0014 30FA    MOVLW  FAh                OPTION = 0B11111010;  // set WDT post scaler = 4
0015 1683    BSF    STATUS,RP0
0016 0081    MOVWF  TMR0
0017 0064    CLRWDT                    CLRWDT();
                                       while(1)
                                         {
0018 3001    MOVLW  01h
0019 202A    CALL   002Ah
001A                                     Delay_Ms_4MHz(1);    // delay for 1mS

001A 1283    BCF    STATUS,RP0         WdtValue++;       // count mS
001B 0A91    INCF   PIR2
001C 2818    GOTO   0018h             }
                                      }
001D 2827    GOTO   0027h           else
                                      {
001E 1283    BCF    STATUS,RP0      CodeByte = 0;
001F 0192    CLRF   TMR1L
```

```
0020 0811    MOVF   PIR2,W                PORTB = WdtValue >> 2;    // divide by 4 for
                                                                  ;    // exact timeout value
0021 008D    MOVWF  PIR1
0022 0C8D    RRF    PIR1
0023 0C8D    RRF    PIR1
0024 303F    MOVLW  3Fh
0025 050D    ANDWF  PIR1,W
0026 0086    MOVWF  PORTB
                                    }
                                   while(1)
0027 0064    CLRWDT                       CLRWDT();          // loop forever
0028 2827    GOTO   0027h
0029 0008    RETURN                }

                                   void Delay_Ms_4MHz(registerw delay)
0000                               {
                                   #asm
002A 1283                        BCF STATUS, RP0     ;1
002B 008D                        MOVWF __WImage      ;1

                         DLMS4M1
                                 RADIX DEC  ;Use decimal values
002C 30F9                        MOVLW 249           ;1
002D 0084                        MOVWF FSR           ;1

                         DLMS4M2                     ; 4 cycles
002E 0000                        NOP                 ;1
002F 0B84                        DECFSZ  FSR         ;1
0030 282E                        GOTO    DLMS4M2     ;2

0031 0B8D                        DECFSZ __WImage     ;1
0032 282C                        GOTO DLMS4M1        ;2
                                       #endasm
0033 0008    RETURN                    }

0000 3000    MOVLW  00h
0001 008A    MOVWF  PCLATH
0002 280A    GOTO   000Ah


ROM USAGE MAP

    0000 to 0002    0005 to 0033
    Total ROM used 0032


Errors             :    0
Warnings           :    0
```

APPENDIX B:

```
MPC "C" COMPILER BC.193 22-Aug-1995                                  PAGE  1


                                /* In this example, a PIC16C54s WDT is calibrated on startup
                                 for better accuracy. Accuracy is exact to +/- 0.25 mS.
                                 The code written in C works on a PICDEM1 board. The  WDT
                                 value is written to PORTB at the end of the calibration.
                                 The value is in binary. */

                                #pragma option v
                                #include <c:\mpc\16c54.h>
                                #pragma option +l;  // Enable header to show branch islands
0200                            #define MAXROM 512
0000 0200                       #pragma memory ROM [MAXROM] @ 0x00;
0008 0018                       #pragma memory RAM [24]     @ 0x08;  // Reserve byte for
                                                                    ;  //  __WImage
                                #pragma option +l;

                                #include <c:\mpc\delay12.h>

                                #pragma option +l;


0009                            char WdtValue;
000A                            char CodeByte;


                                void InitPorts(void)
                                {
0000 0066    CLRF   PORTB           PORTB = 0;     //set PORTB as output
0001 0C00    MOVLW  00h             __TRIS(0,PORTB);
0002 0006    TRIS   PORTB
0003 0800    RETLW  00h         }

                                void main(void)
                                {
0004 0900    CALL   0000h           InitPorts();
0005 0CA5    MOVLW  A5h             if (CodeByte != 0xA5)
0006 008A    SUBWF  0A,W
0007 0643    BTFSC  STATUS,Z
0008 0A14    GOTO   0014h
0009                                  {
0009 0069    CLRF   09                WdtValue = 0;
000A 0CA5    MOVLW  A5h               CodeByte = 0xA5;
000B 002A    MOVWF  0A
000C 0CFA    MOVLW  FAh               __OPTION(0B11111010); // set WDT post scaler = 4
000D 0002    OPTION
000E 0004    CLRWDT                   CLRWDT();
                                      while(1)
                                        {
000F 0C01    MOVLW  01h
0010 091F    CALL   001Fh
0011                                    Delay_Ms_4MHz(1); // delay for 1 mS
0011 02A9    INCF   09                  WdtValue++;       // count 1ms delays
0012 0A0F    GOTO   000Fh               }
                                      }
0013 0A1C    GOTO   001Ch           else

                                      {
0014 006A    CLRF   0A                CodeByte = 0;
0015 0209    MOVF   09,W              PORTB = WdtValue >> 2;  //divide by 4 for exact
                                                              // timeout value
0016 0027    MOVWF  PORTC
0017 0327    RRF    07
```

```
0018 0327    RRF    07
0019 0C3F    MOVLW  3Fh
001A 0147    ANDWF  07,W
001B 0026    MOVWF  PORTB
                                    }
                                 while(1)
001C 0004    CLRWDT                CLRWDT(); // loop forever
001D 0A1C    GOTO   001Ch
001E 0800    RETLW  00h          }


                                 void Delay_Ms_4MHz(registerw delay)
0000                             {
                                 #asm
001F 0000                NOP     ;1
0020 0027                MOVWF __WImage      ;1

                       DLMS4M1
                          RADIX DEC  ;Use decimal values
0021 0CF9                 MOVLW 249          ;1
0022 0028                 MOVWF __FSRImage          ;1

                       DLMS4M2               ;4 cycles
0023 0000                   NOP               ;1
0024 02E8                   DECFSZ __FSRImage        ;1
0025 0A23                   goto  DLMS4M2     ;2

0026 02E7                 DECFSZ __WImage     ;1
0027 0A21                 goto DLMS4M1        ;2
                                 #endasm
0028 0800    RETLW  00h          }

01FF 0A04    GOTO   0004h


ROM USAGE MAP

    0000 to 0028    01FF to 01FF
    Total ROM used 002A


Errors             :    0
Warnings           :    0
```

NOTES:

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.microchip.com

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: 972 991-7177 Fax: 972 991-8588

Dayton
Microchip Technology Inc.
Suite 150
Two Prestige Place
Miamisburg, OH 45342
Tel: 513 291-1654 Fax: 513 291-9175

Los Angeles
Microchip Technology Inc.
18201 Von Karman, Suite 1090
Irvine, CA 92612
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

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

Toronto
Microchip Technology Inc.
5925 Airport Road, Suite 200
Mississauga, Ontario L4V 1W1, Canada
Tel: 905 405-6279 Fax: 905 405-6253

ASIA/PACIFIC

Hong Kong
Microchip Asia Pacific
RM 3801B, Tower Two
Metroplaza
223 Hing Fong Road
Kwai Fong, N.T. Hong Kong
Tel: 852 2 401 1200 Fax: 852 2 401 3431

India
Microchip Technology India
No. 6, Legacy, Convent Road
Bangalore 560 025 India
Tel: 91 80 529 4846 Fax: 91 80 559 9840

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

Shanghai
Microchip Technology
Unit 406 of Shanghai Golden Bridge Bldg.
2077 Yan'an Road West, Hongiao District
Shanghai, Peoples Republic of China
Tel: 86 21 6275 5700
Fax: 011 86 21 6275 5060

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

Taiwan, R.O.C
Microchip Technology Taiwan
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 1628 851077 Fax: 44 1628 850259

France
Arizona Microchip Technology SARL
Zone Industrielle de la Bonde
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 Colleone
Palazzo Taurus 1 V. Le
Colleoni 1
20041 Agrate Brianza
Milan Italy
Tel: 39 39 6899939 Fax: 39 39 689 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

11/25/96

MICROCHIP

Printed on recycled paper.