AN720

Measuring Temperature Using the Watch Dog Timer (WDT)

Author: *Ian Lao*
Microchip Technology Inc.
Chandler, AZ

INTRODUCTION

This application note shows how Microchip Technology's Watch Dog Timer (WDT) can be used to acquire rough temperature measurements.

Recent advances in sensor technology have allowed for the development of many different sensors to measure temperature. However, almost all of these are implemented as dedicated function sensors.

Microchip has now developed a method of combining both rough temperature sensing and microcontroller functionality on the same device without the need for external components.

Preliminary analysis of the on-board WDT shows a piecewise linear correlation between temperature and the timeout period of the WDT. The WDT timeout period appears to increase for a fixed V_{DD} as temperature increases. Tests indicate that this property may be used for cost effective rough temperature sensing.

The WDT module is similar across many families of microcontrollers from Microchip. This allows for a wide range of different applications to be developed using the same technique.

Though actual application results may differ, an accuracy of up to $\pm 1°C$ may be seen. The linearity of the WDT is not guaranteed but has been observed.

Note:	It is up to the user to test the device in the system to determine accuracy/usability.

THEORY

The WDT is an 8-bit timer with an 8-bit pre-scaler option driven from a free running on-chip RC oscillator. This oscillator is completely independent of pins OSC1/CLKIN, OSC2/CLKOUT, and the INTRC oscillator. As with any RC oscillator, variances in temperature will affect the frequency of the circuit. Cumulative effects will therefore show up as a change in the timeout period of the WDT.

By utilizing another timer as a reference, a sample may be established whereby changes in the WDT timeout period can be measured. Calibrated temperature can then be derived via Equation 1.

Equation 1:

CC = COUNT*Scalar - Offset

CC => calibrated count value

C => COUNT; number of times TMR0 has rolled over

Offset => calibration offset due to voltage variance or self-heating (determined by testing against a known fixed temperature)

Scalar => calibration scalar due to process or application design ("slope" determined by testing 2 known temperatures)

Process variations across lots, part families, and different cores are expected. Since the WDT is clocked by an RC oscillator, these differences are expected to influence the "slope" of the piecewise linear WDT response.

HARDWARE REQUIRED

1. Voltage/temperature regulated power supply
2. Temperature-compensated oscillator or crystal clock source

Note:	If the INTRC is used for the reference timer, no external clock components are required to implement this design. For greater accuracy, an external temperature-compensated oscillator may be used.

IMPLEMENTATION

Resources Used

This design uses two timers and a 16-bit count register to count the number of times TMR0 has rolled over since the last WDT timeout. Two calibration constants are used to negate the effects of self-heating and process variation/application design.

1. Reference Timer (TMR0);
   The reference timer may be implemented using the INTRC or an external temperature-compensated clock source to drive TMR0.
2. Measurement Timer (WDT);
   The WDT is utilized as the measurement timer. It is configured to use the on-board prescaler that is set to a ratio of 1:8 in this example. A ratio of 1:8 was chosen to allow the 16-bit count register to capture usable TMR0 roll overs without overflowing. This ratio also allows for a granularity in the count register small enough to detect changes in temperature.

Note: Users should test their code to determine the appropriate prescaler ratio to use in their application.

Firmware

Once TMR0 and WDT are configured, both are released to begin incrementing. A 16-bit register is used to count the number of times TMR0 rolls over (COUNT). TMR0 is allowed to continue incrementing and rolling over until the WDT times out. This COUNT is then used as the input to Equation 1 to give a resultant calibrated count.

Use caution when interrupts other than TMR0 (for devices that have interrupts), are active during rough temperature measurements to ensure capturing all TMR0 roll over events. WDT timeouts are asynchronous events. Missing a TMR0 rollover will add to the error of the reading.

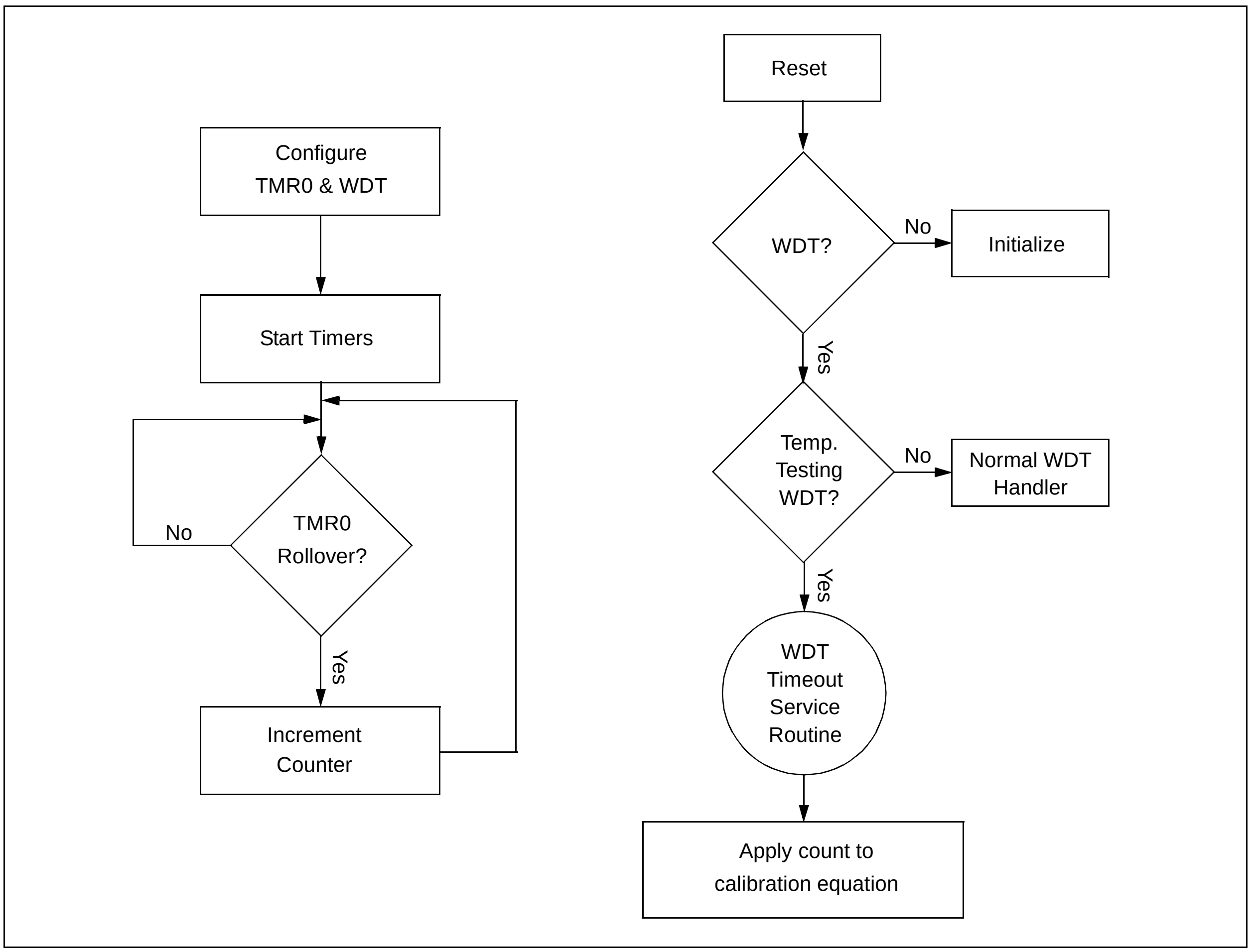
A look-up table or algorithm may be used to convert the calibrated count to Fahrenheit or Celsius for display.

Figure 1 illustrates the flow diagram for this program.

Appendix A is the source code listing.

Note: The part must not be put into sleep mode during temperature measurements as sleep mode disables TMR0.

FIGURE 1: FIRMWARE FLOW DIAGRAM



CALIBRATION

In using the WDT to measure temperature, calibration of the microcontroller against system errors is required. Since the WDT is piece-wise linear with temperature, we know that the two major components of error are the Scalar (Slope) of the line and the "offset" of the line. Process variations in the RC oscillator, which clocks the WDT and the application design itself, will determine Scalar. Variations in operating voltage and self-heating cause "offset".

In order to calibrate a part to measure temperature, both of these co-efficients must be determined and stored in memory for future use. Two dedicated memory locations (normally near the end of memory) are used to store them. Users should write their application program to include a calibration mode that uses the WDT temperature measurement mechanism, but outputs the uncalibrated count values onto the port pins. This program is then run against two known calibration temperatures. The difference in count values divided by the difference in known temperatures is the Scalar. By assigning a calibrated COUNT value to one of the two known calibration temperatures and solving Equation 1, the "offset" can be determined. In-Circuit Serial Programming™ (ICSP) mode or Serial EEPROM can then be used to store the two calibration values.

All of the sources of error mentioned in Section should also be taken into consideration when calibrating.

EXAMPLE 1:

Calibration example assuming:

1. Fixed temperature-compensated VDD
2. Fixed temperature-compensated reference oscillator
3. Area of temperature interest: +25°C - +75°C
4. Measured uncalibrated COUNTS @ +25°C Calibration Point 1: COUNT = 475 decimal
5. Measured uncalibrated COUNTS @ +75°C Calibration Point 2: COUNT = 595 decimal

To calculate the Scalar (Slope), the formula is:

$$\text{Scalar} = \frac{\text{Cal Point 2 - Cal Point 1}}{\text{Temp Cal Point 2 - Temp Cal Point 1}}$$

$$\text{Scalar} = \frac{595 - 475}{+75°\text{C} - +25°\text{C}} = 2.4 \text{ COUNT/°C}$$

Scaler = 2.4 COUNT/°C

To calculate the offset, the formula is:

Assigned Cal. COUNT Value = COUNT x Scalar - Offset

Assume Assigned Value = 0

0 = COUNT x Scalar - Offset

Offset = COUNT x Scalar

@ +25°C Offset = Uncal. COUNT x Scalar

1140.0 = 475 x 2.4

Now Scalar = 2.4 and Offset = 1140.0

EXAMPLE 2:

To make a calibrated COUNT calculation @ 55°C:

CC = COUNT x Scalar - Offset

@ +55°C 192 = 555.0 x 2.4 - 1140.0

SOURCES OF ERROR

When taking temperature measurements, errors may be introduced into the calculations. The most common sources of errors are:

1. Insufficient soak time;
   A certain amount of time is required for any system to stabilize. The varying materials used typically require time to reach thermal equilibrium.
2. Insufficient acquisition time;
   Total acquisition time is typically represented by the equation:

   $T_{Aq} = T_{Soak} + T_{Sample}$

   T_{Aq} => acquisition time. Total time to make a calibrated measurement.

   T_{Soak} => soak time to reach thermal equilibrium

   T_{Sample} => time required to capture a number of uncalibrated COUNTS and average the result of the raw data through a "debounce" algorithm
3. Calibration errors;
   Errors may be introduced by incorrectly determining the Scalar or Offset values. Both of these equation terms are based on controlled known temperatures.
4. Sample error;
   Since temperature does not change quickly (i.e., in the milliseconds), typical applications will apply an algorithm similar to "debounce" that will filter out momentary spikes and steps in temperature readings.
5. Power supply;
   Variances in power supply voltage will effect the INTRC, external oscillator and WDT RC oscillator. These affects may be self-canceling in your application.
6. Reference oscillator;
   Variances in the reference oscillator due to process, voltage or temperature will affect TMR0.

COMMON USES

Many designs typically use rough temperature data as trip points to indicate over-heating or operation below recommended minimum temperature specifications. Other uses may include but are not limited to:

1. Rough calibration of other hardware/systems/ processes
2. Temperature hysteresis measurements

EXPERIMENTAL DATA

The data in Figure 2 was collected using a sample of 8 typical production PIC12C509A parts from the same manufacturing lot. A test board containing all eight parts was then given a soak time of thirty minutes at each tested temperature. Five hundred uncalibrated raw data COUNTS were then recorded and averaged for each tested temperature to produce Figure 2.

- Voltage was supplied and measured via a Topward 3303D DC power supply and Fluke model 87 DMM, respectively.
- A Hart Scientific High Precision Bath Model 7025 with Hart Scientific Black Stack Temperature Probe model 2560 provided the various different temperatures.
- Data was captured using Hyperterminal running on a Windows 95 configured PC.

FIGURE 2: UNCALIBRATED COUNT DATA (V_{DD} = 5.0V)

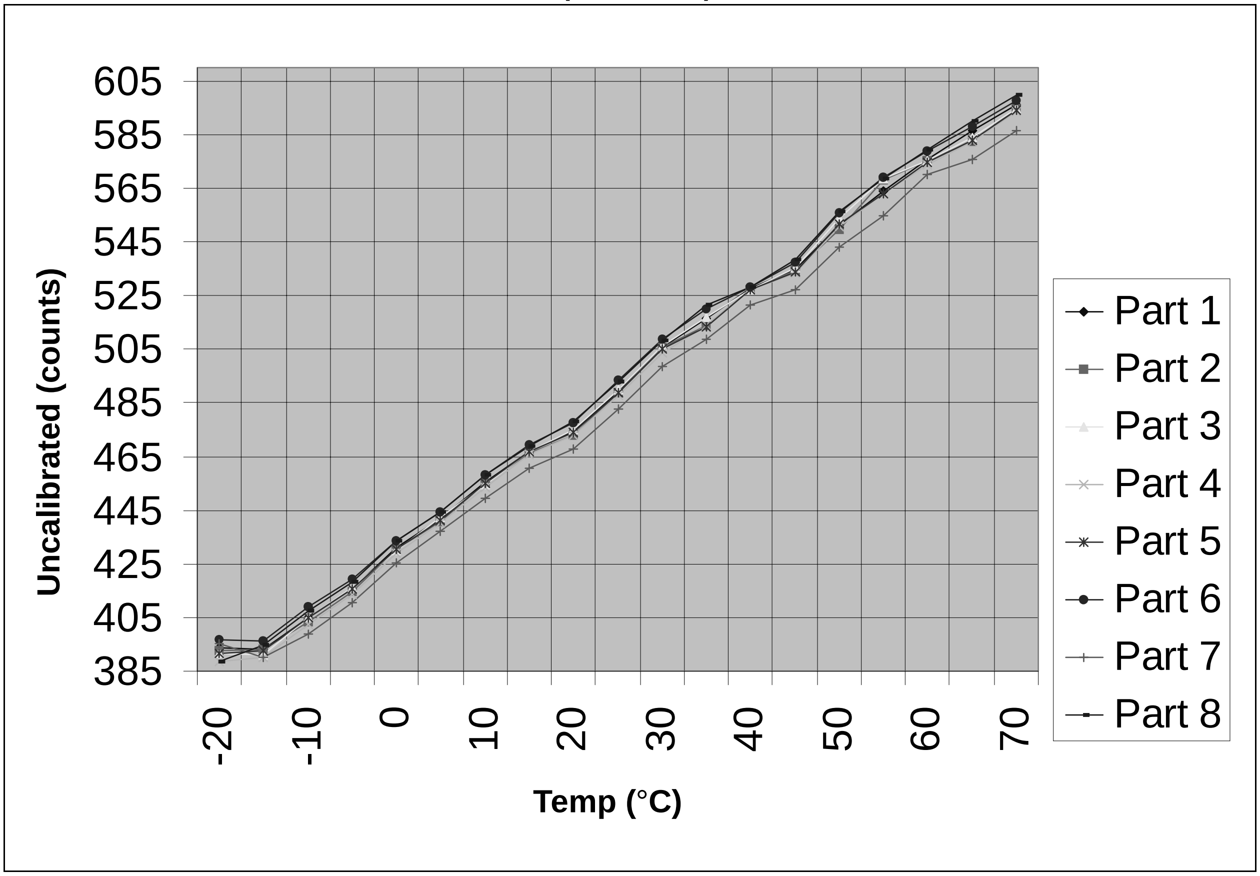


Figure 3 illustrates the standard deviation of the averages listed in Figure 2 across all eight parts under test at each temperature.

FIGURE 3: ACROSS PARTS (V_{DD} = 5.0V)

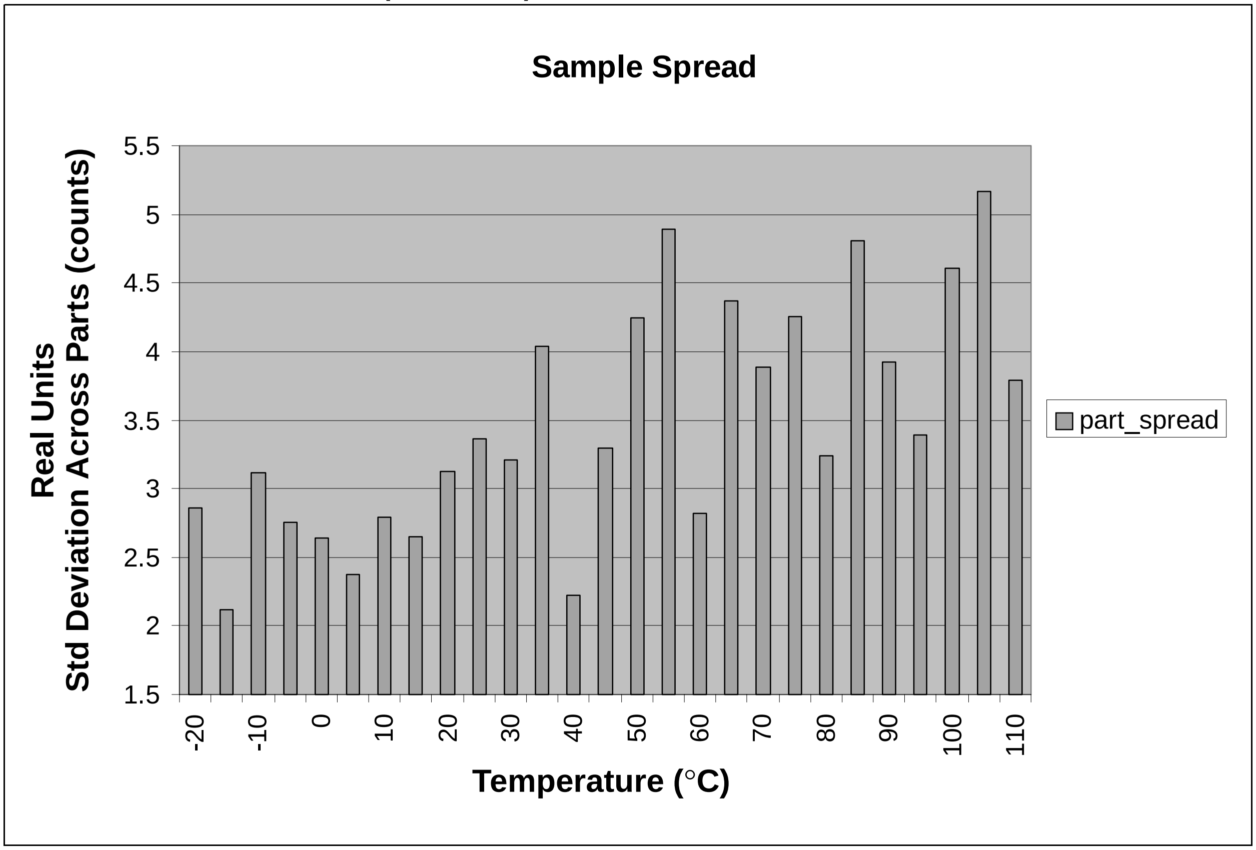


Figure 4 illustrates the standard deviation of the five hundred uncalibrated count data points collected to generate the uncalibrated count averages listed in Figure 2. The three parts with the greatest deviation are listed.

FIGURE 4: ACROSS RAW DATA POINTS (V_{DD} = 5.0V)

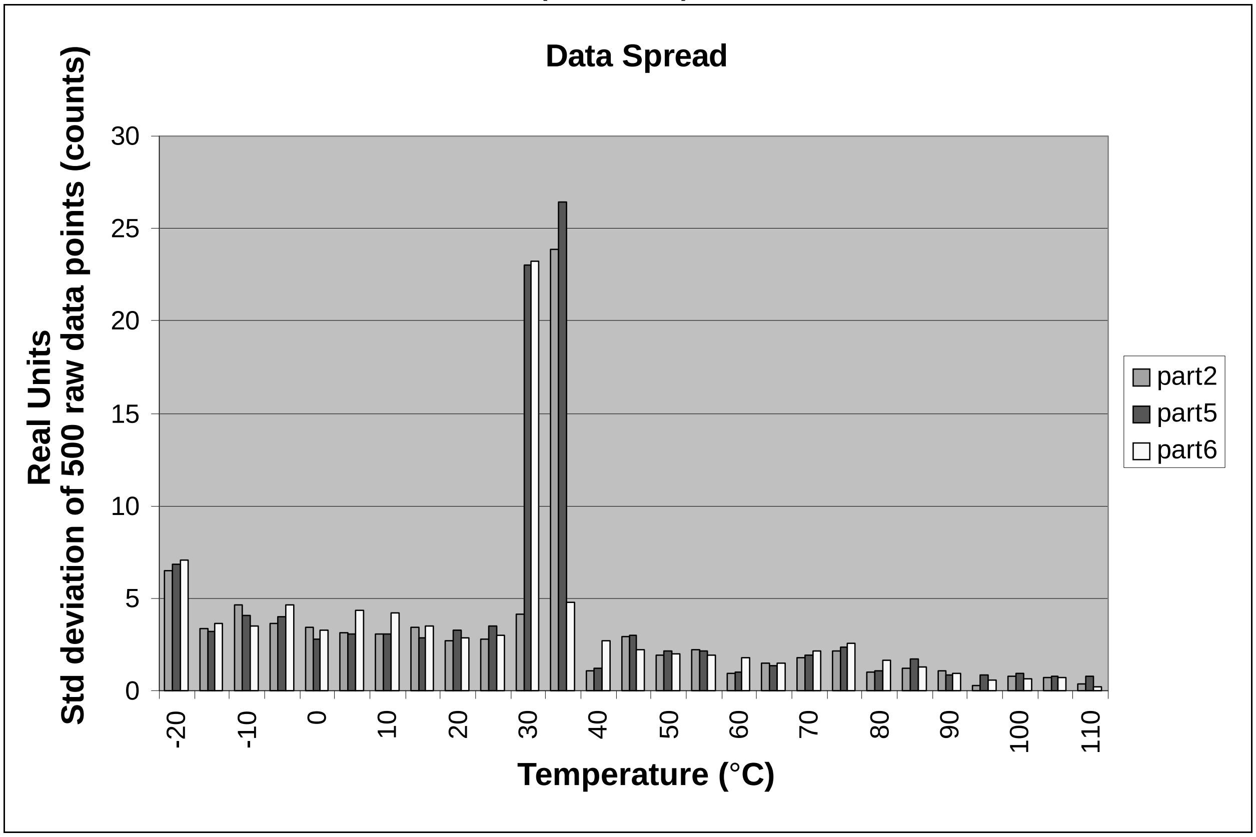
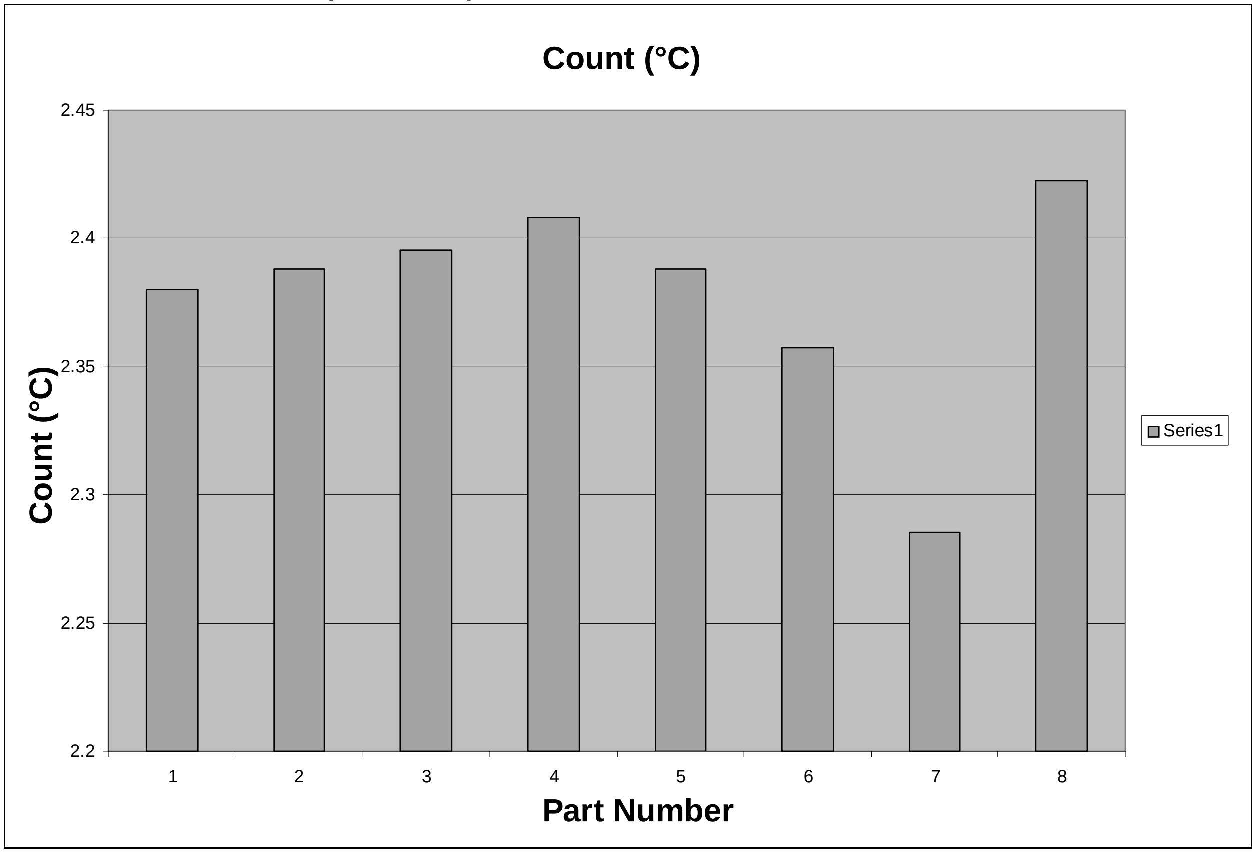


Figure 5 illustrates the calculated uncalibrated COUNTS per degree C for each of the eight tested parts.

FIGURE 5: COUNTS/°C (V_{DD} = 5.0V)



APPENDIX A: SOURCE CODE

```
MPASM 02.30 Released         TSTAT2~1.ASM   9-15-1999  13:06:10         PAGE  1


LOC  OBJECT CODE     LINE SOURCE TEXT
  VALUE

          00001
;***************************************************************************************
          00002 ;This program demonstrates how the WDT and TMR0(reference timer) may be used for
          00003 ;rough temperature measurements. No filtering/debounce or algorithm is applied on
          00004 ;the raw data. The raw un-calibrated COUNTS are output to a PIC16C54C for transmittal
          00005 ;to a PC. GP<1:0> are used for data communication and GP3 is used as an output
                ;enable.
          00006 ;In typical applications, users will need to add code to cover WDT time out when not
          00007 ;taking rough temperature measurements. WDT tracking register WDTSTAT bit 0 used to
          00008 ;indicate if WDT timeouts are being used for rough temp measurements or in the normal
          00009 ;application.
          00010 ;
          00011 ;
          00012 ;       Program:        TSTAT2~1.ASM
          00013 ;       Revision Date:  9/7/99 Compatibility with MPlab 4.11
          00014 ;
          00015 ;
          00016 ;
          00017
;***************************************************************************************
          00018
          00019
          00020         LIST P=PIC12C509A;, F=INHX8M
          00021         #include "P12C509A.INC"
          00001         LIST
          00002 ; P12C509A.INC  Standard Header File, Version 1.00    Microchip Technology, Inc.
          00108         LIST
          00022
0FFF 0FFE 00023         __CONFIG _MCLRE_OFF & _CP_OFF & _WDT_ON & _IntRC_OSC
          00024
          00025 ;;
          00026 ;       declare registers
          00027
          00028 ;Note *
          00029 ;       All core program variables in page 0
          00030 ;
          00031
          00032         cblock          0x07    ;bank 0
          00033
 00000007 00034 T_COUNT:2                      ;counter for # of times tmr0 rolls (lo/hi byte)
 00000009 00035 SCREEN                         ;screen register for tmr0 roll over
 0000000A 00036 DUMP                           ;holding register
 0000000B 00037 BIT_COUNT                      ;# of bits to be sent
 0000000C 00038 WDTSTAT                        ;status register of wdt being used in
          00039                                ;temperature or normal application mode
          00040
 0000000D 00041 TEMP6                          ;temp register used by routines
 0000000E 00042 TEMP7                          ;
 0000000F 00043 TEMP8                          ;
          00044
          00045         endc
          00046 ;
          00047 ;
          00048 ;;
          00049
0000      00050         org     0x00
```



```
MPASM 02.30 Released         TSTAT2~1.ASM   9-15-1999  13:06:10         PAGE  2


LOC  OBJECT CODE     LINE SOURCE TEXT
  VALUE

0000 0025 00051         movwf   OSCCAL           ;load osc calibration for IntRC
0001 0C01 00052         movlw   b'00000001'      ;clear bus driver latch
0002 0026 00053         movwf   GPIO             ;
0003 0CFF 00054         movlw   b'11111111'      ;disable bus drivers
0004 0006 00055         tris    GPIO             ;
0005 04A3 00056         bcf     STATUS,PA0       ;set bank pointers to page 0
0006 04A4 00057         bcf     FSR,5            ;set address map to page 0
0007 04C4 00058         bcf     FSR,6
0008 0A09 00059         goto    Resetvector
          00060
          00061 ;;
          00062 ;       main memory
          00063
          00064
          00065         ;reset vector
0009      00066 Resetvector                      ;
          00067
0009 0C8B 00068         movlw   b'10001011'      ;load option register word
000A 0002 00069         option                   ;
          00070
          00071         ;check for power on reset
000B 0783 00072         btfss   STATUS,NOT_TO    ;must test condition of TO=1
000C 0A1B 00073         goto    Wdtest           ;to tell if power on reset.
          00074                                  ;there is no sleep mode support.
          00075                                  ;if not a POR, must be a WDT reset.
          00076                                  ;jump to the POR or WDT routines.
          00077
          00078 ;;
          00079         ;power on reset handler
000D      00080 P_reset                          ;initializtion routine
          00081
000D 0C00 00082         movlw   0x00             ;clear counters for measurement
000E 0027 00083         movwf   T_COUNT          ;
000F 0028 00084         movwf   T_COUNT+1        ;
0010 002C 00085         movwf   WDTSTAT          ;clear wdt tracking register
          00086
          00087
0011 050C 00088         bsf     WDTSTAT,0        ;set tracking register bit 0 to
          00089                                  ;indicate that wdt timeouts are being
          00090                                  ;used for rough temp measurements.
          00091                                  ;This register is typically set elsewhere
          00092                                  ;in a real application but for the
          00093                                  ;purposes of this example, is set here.
          00094
          00095
          00096                 ;init timers
0012 0004 00097         clrwdt                   ;initialize wdt
0013 0C00 00098         movlw   0x00             ;initialize timer0
0014 0021 00099         movwf   TMR0             ;and allow to free run
          00100
0015 0A16 00101         goto    $+1              ;delay to let tmr0 go past
0016 0A17 00102         goto    $+1              ;screen point
0017 0A18 00103         goto    $+1              ;
```

```
MPASM 02.30 Released         TSTAT2~1.ASM   9-15-1999  13:06:10         PAGE  3


LOC  OBJECT CODE     LINE SOURCE TEXT
  VALUE

0018 0A19 00104         goto    $+1             ;
0019 0A1A 00105         goto    $+1             ;
          00106
001A 0A57 00107         goto    Countimer       ;branch to counting routine
                                       00109 ;;
          00110         ;test what type of interupt
001B      00111 Wdtest
          00112                 ;test for wdt in temp measure or normal mode
001B 070C 00113         btfss   WDTSTAT,0       ;test wdt mode tracking bit.
          00114                                 ;if =1 then is in temperature mode.
          00115                                 ;if =0 then is in normal app mode.
001C 0A64 00116         goto    Nontempwdt      ;vector to normal app wdt handler here.
          00117                                 ;
          00118
          00119                 ;wdt temperature handler
001D      00120 Wdtvector
          00121                 ;print raw uncalibrated data
          00122
001D      00123 Raw
001D 0C00 00124         movlw   b'00000000'     ;zero communications bus and wait
001E 0026 00125         movwf   GPIO            ;to transfer data
001F 0CFF 00126         movlw   b'11111111'     ;while looking for output enables
0020 0006 00127         tris    GPIO            ;
          00128
          00129
0021      00130 OE                              ;test to see if output is enabled
          00131
0021 0004 00132         clrwdt
0022 0206 00133         movf    GPIO,W          ;sample portb
0023 0E08 00134         andlw   b'00001000'     ;mask unwanted bits
0024 002A 00135         movwf   DUMP            ;move to temporary register for test
0025 0C08 00136         movlw   b'00001000'     ;do test
0026 008A 00137         subwf   DUMP,W          ;
0027 0743 00138         btfss   STATUS,Z        ;test carry bit to see if OE.
0028 0A21 00139         goto    OE              ;cannot proceed to send data if no OE
          00140                 ;
          00141
0029      00142 Print                           ;setup for xfering data
          00143
0029 0C00 00144         movlw   b'00000000'     ;clear data latch
002A 0026 00145         movwf   GPIO            ;
002B 0CFD 00146         movlw   b'11111101'     ;set tris register
002C 0006 00147         tris    GPIO            ;
002D 0C11 00148         movlw   0x11            ;setup bit counter
002E 002B 00149         movwf   BIT_COUNT       ;to send 2 bytes of data
          00150                 ;
          00151
002F      00152 Clock_en                        ;once clock setup, check for
          00153                                 ;complete sending of all 2 bytes
          00154
002F 02EB 00155         decfsz  BIT_COUNT,F     ;test if 16 bits sent
0030 0A32 00156         goto    Senddata        ;
```

```
MPASM 02.30 Released         TSTAT2~1.ASM   9-15-1999  13:06:10         PAGE  4


LOC  OBJECT CODE     LINE SOURCE TEXT
  VALUE

0031 0A62 00157         goto    Softreset       ;reinit to take another measurement
          00158                 ;
          00159
          00160
0032      00161 Senddata                        ;must figure out whether sending upper or
          00162                                 ;lower byte
          00163
0032 0C09 00164         movlw   0x09            ;test if upper byte or lower byte
0033 008B 00165         subwf   BIT_COUNT,W     ;
0034 0603 00166         btfsc   STATUS,C        ;check to see iv value is zero
0035 0A37 00167         goto    Lower_8         ;jump to send lo byte
0036 0A47 00168         goto    Upper_8         ;jump to send hi byte
          00169                 ;
          00170
0037      00171 Lower_8
          00172
0037      00173 Test_lo                         ;check for clock strobe from receiving
          00174                                 ;unit. Clock must be lo. Then go hi.
          00175
0037 0004 00176         clrwdt
0038 0206 00177         movf    GPIO,W          ;test for clock lo to see if ready
0039 002A 00178         movwf   DUMP            ;put in temp register
003A 060A 00179         btfsc   DUMP,0          ;
003B 0A37 00180         goto    Test_lo         ;
          00181                 ;
          00182
003C      00183 Test_hi                         ;check for clock strobe. Send only on lo to
          00184                                 ;hi clock transition
          00185
003C 0004 00186         clrwdt
003D 0206 00187         movf    GPIO,W          ;test for clock hi to see if send
003E 002A 00188         movwf   DUMP            ;put in temp register
003F 070A 00189         btfss   DUMP,0          ;
0040 0A3C 00190         goto    Test_hi         ;
          00191                 ;
          00192
0041      00193 Lower_8_send                    ;xmit data 1 bit at a time by rotating thru
          00194                                 ;carry and checking it's value
          00195
0041 0426 00196         bcf     GPIO,1          ;reset data line
0042 0327 00197         rrf     T_COUNT,F       ;rotate into carry to test for 1 or 0
0043 0603 00198         btfsc   STATUS,C        ;test for 1 or 0
0044 0526 00199         bsf     GPIO,1          ;clear sending bit
0045 0000 00200         nop
          00201                 ;
          00202
          00203
0046 0A2F 00204         goto    Clock_en        ;return to send next data bit
          00205                 ;
          00206                 ;
          00207
          00208
0047      00209 Upper_8
```

```
MPASM 02.30 Released         TSTAT2~1.ASM   9-15-1999  13:06:10         PAGE  5


LOC  OBJECT CODE     LINE SOURCE TEXT
  VALUE

          00210
          00211
0047      00212 Test_lo_u                       ;check for clock strobe from receiving
          00213                                 ;unit. Clock must be lo. Then go hi.
          00214
0047 0004 00215         clrwdt
0048 0206 00216         movf    GPIO,W          ;test for clock lo to see if ready
0049 002A 00217         movwf   DUMP            ;put in temp register
004A 060A 00218         btfsc   DUMP,0          ;
004B 0A47 00219         goto    Test_lo_u       ;
          00220                 ;
          00221
004C      00222 Test_hi_u                       ;check for clock strobe. Send only on lo to
          00223                                 ;hi clock transition
          00224
004C 0004 00225         clrwdt
004D 0206 00226         movf    GPIO,W          ;test for clock hi to see if send
004E 002A 00227         movwf   DUMP            ;put in temp register
004F 070A 00228         btfss   DUMP,0          ;
0050 0A4C 00229         goto    Test_hi_u       ;
          00230                 ;
          00231
0051      00232 Upper_8_send                    ;xmit data 1 bit at a time by rotating thru
          00233                                 ;carry and checking it's value
          00234
0051 0426 00235         bcf     GPIO,1          ;reset data line
0052 0328 00236         rrf     T_COUNT+1,F     ;rotate into carry to test for 1 or 0
0053 0603 00237         btfsc   STATUS,C        ;test for 1 or 0
0054 0526 00238         bsf     GPIO,1          ;clear sending bit
0055 0000 00239         nop
          00240                 ;
          00241
          00242
0056 0A2F 00243         goto    Clock_en        ;return to send next data
          00244                 ;
          00245                 ;
          00246
          00247
          00248
          00249 ;;
          00250         ;counting routine
0057      00251 Countimer
          00252
          00253         ;test to see if timer0 rolls over
0057      00254 Tmr0_byte                               ;count the number of tmr0's
          00255
0057 0201 00256         movf    TMR0,W                  ;copy tmr0 value to working register
0058 0029 00257         movwf   SCREEN                  ;
0059 0C0A 00258         movlw   0x0A                    ;load masking value
005A 0089 00259         subwf   SCREEN,W                ;subtraction to screen for FF -> 0
          00260                                         ;transition in tmr0
005B 0603 00261         btfsc   STATUS,C                ;test carry flag for
005C 0A57 00262         goto    Tmr0_byte               ;loop back and test for FF -> 0
```



```
MPASM 02.30 Released         TSTAT2~1.ASM   9-15-1999  13:06:10         PAGE  6


LOC  OBJECT CODE     LINE SOURCE TEXT
  VALUE

          00263
          00264         ;increment count lo byte
005D 02A7 00265         incf    T_COUNT,F               ;incr count (lo byte) once for every
          00266                                         ;tmr0 roll over
005E 0743 00267         btfss   STATUS,Z                ;test zero flag to see if need to
          00268                                         ;increment hi byte of count (16 bit
counter)
005F 0A57 00269         goto    Tmr0_byte               ;loop back and test until wdt reset
          00270
          00271         ;increment count hi byte
0060 02A8 00272         incf    T_COUNT+1,F             ;incr count (hi byte) once for every
          00273                                         ;T_COUNT roll over
0061 0A57 00274         goto    Tmr0_byte               ;loop back and test until wdt reset
          00275
          00276
          00277 ;;
          00278         ;soft reset routine
0062      00279 Softreset                               ;clear conditions and reset for another
          00280                                         ;rough temperature measurement
          00281
0062 0004 00282         clrwdt                          ;clear the wdt
0063 0A0D 00283         goto    P_reset                 ;return to reset checks
          00284
          00285
          00286 ;;
          00287         ;non-temp measurement mode wdt handler
0064      00288 Nontempwdt
0064 0A64 00289         goto $                          ;normal mode wdt timeout handler.
          00290                                       ;since only running in rough temp measure
          00291                                         ;mode, routine is just a place holder.
          00292
          00293
          00294 ;;
          00295         end
```

```
MPASM 02.30 Released         TSTAT2~1.ASM   9-15-1999  13:06:10         PAGE  7


SYMBOL TABLE
  LABEL                             VALUE

BIT_COUNT                         0000000B
C                                 00000000
Clock_en                          0000002F
Countimer                         00000057
DC                                00000001
DUMP                              0000000A
F                                 00000001
FSR                               00000004
GPIO                              00000006
GPWUF                             00000007
INDF                              00000000
Lower_8                           00000037
Lower_8_send                      00000041
NOT_GPPU                          00000006
NOT_GPWU                          00000007
NOT_PD                            00000003
NOT_TO                            00000004
Nontempwdt                        00000064
OE                                00000021
OSCCAL                            00000005
OSCFST                            00000003
OSCSLW                            00000002
PA0                               00000005
PCL                               00000002
PS0                               00000000
PS1                               00000001
PS2                               00000002
PSA                               00000003
P_reset                           0000000D
Print                             00000029
Raw                               0000001D
Resetvector                       00000009
SCREEN                            00000009
STATUS                            00000003
Senddata                          00000032
Softreset                         00000062
T0CS                              00000005
T0SE                              00000004
TEMP6                             0000000D
TEMP7                             0000000E
TEMP8                             0000000F
TMR0                              00000001
T_COUNT                           00000007
Test_hi                           0000003C
Test_hi_u                         0000004C
Test_lo                           00000037
Test_lo_u                         00000047
Tmr0_byte                         00000057
Upper_8                           00000047
Upper_8_send                      00000051
W                                 00000000
WDTSTAT                           0000000C
Wdtest                            0000001B
```

```
MPASM 02.30 Released         TSTAT2~1.ASM   9-15-1999  13:06:10         PAGE  8


SYMBOL TABLE
  LABEL                             VALUE

Wdtvector                         0000001D
Z                                 00000002
_CP_OFF                           00000FFF
_CP_ON                            00000FF7
_ExtRC_OSC                        00000FFF
_IntRC_OSC                        00000FFE
_LP_OSC                           00000FFC
_MCLRE_OFF                        00000FEF
_MCLRE_ON                         00000FFF
_WDT_OFF                          00000FFB
_WDT_ON                           00000FFF
_XT_OSC                           00000FFD
__12C509A                         00000001


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

0000 : XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX
0040 : XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXX----------- ----------------
0FC0 : ---------------- ---------------- ---------------- ---------------X

All other memory blocks unused.

Program Memory Words Used:   101
Program Memory Words Free:   923


Errors   :     0
Warnings :     0 reported,     0 suppressed
Messages :     0 reported,     0 suppressed
```