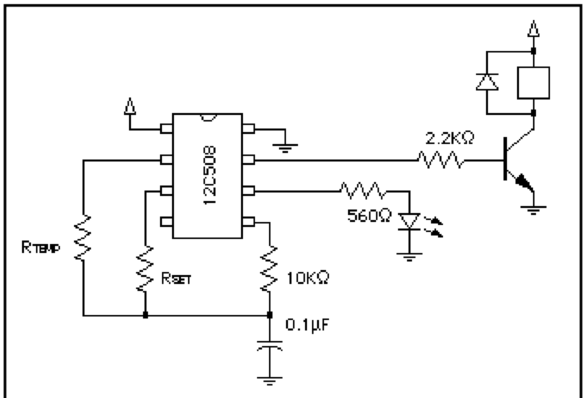
Solid State Thermostat Using PIC12C508

Author: Jim Nagy
London, Ontario
Canada
email: nagy@wwdc.com

OVERVIEW

A solid state thermostat, using a PIC12C508 as the measuring and control device. This circuit demonstrates the PIC12C508's ability to measure an unknown resistance, and either communicate it's value to another device through a serial connection, or to provide direct control outputs, as has been done in this case. The resistance could be from a light dependent resistor, a temperature dependent resistor, or any other variable resistance.

BLOCK DIAGRAM



APPLICATION OPERATION

This circuit takes advantage of the schmitt trigger input of GP2 to monitor the voltage on a capacitor while it is discharged - first through a known reference resistor, then through an unknown resistance. The time taken to discharge the capacitor in each case is recorded, and used to determine a course of action. For this application, the two times obtained are simply compared to one another (with some hysteresis added to avoid chatter), but they could also be used to determine the value of the unknown resistance, by noting that the ratio of the two resistances is equal to the ratio of the two discharge times for each.

The thermostat has been designed as a backup thermostat to the main one in our house. It operates a relay below a fixed setpoint of 10°C or 50°F, should the primary heating control fail. Since the output is from a relay, though, it could be put to a variety of uses. By reversing the logic (or the contact), the output could indicate cooling failure, whether from air conditioning, or from a food freezer (that should remain below 0°C or 32°F).

The circuit is constructed as shown in the block diagram. The output relay is powered by 12V, while the PIC12C508 is powered from 5V through a 78L05 regulator. The transistor is a standard 2N3904, and the diode is a 1N4001. The main components of interest, however, are the Rtemp, Rset and 10K resistors, as well as the 0.1µF capacitor. These four components make up the measuring circuit.

A measurement is conducted by charging the capacitor to +5V through the 10K resistor, then switching GP2 to an input, and discharging the capacitor through the setpoint resistor (Rset - 18K), then repeating the process using the unknown resistor (Rtemp - a 10K at 25°C thermistor such as the 271-110A from Radio Shack). If the time taken to discharge the capacitor through Rtemp is greater than the time for Rset, for three consecutive counts, then GP0 is set high, energizing the output relay. The relay will remain energized until a temperature of 12°C or 54°F is measured. The 18K (Rset) value was chosen as it represents a temperature of 10°C or 50°F. A value of 30K could be chosen for a freezer alarm, or 8.2K for an A/C alarm.

Sensor Interface

This circuit uses the watchdog timer to wake up at maximum (2.3 sec) intervals, then make a measurement. During each measurement, the LED is turned on, providing some visual feedback to show there is activity. Should the temperature be found to be out of limits, however, (likely due to a bad sensor) the error condition is shown by a reduction in the watchdog period, effectively causing a faster (warning) flash rate.

One problem that occurred during the design of this circuit is that the WDT seemed to reset the IO port state to all inputs upon wakeup from sleep, so I couldn't just check the present state with a BTFS instruction. Rather than rely on file storage to determine the required state of the output I believe I took a novel approach by using the timing capacitor as a memory storage element. Upon wakeup, the level at GP2 is checked, and if it's high, then the output to the relay is driven high as well. The small blip in coil voltage isn't even noticed by the relay.

BILL OF MATERIALS

Part#	Manufacture
PIC12C508	Microchip
10K@25°C Thermistor	One source is Radio Shack 271-110A

APPENDIX A: SOFTWARE LISTING

```
;
; PICostat
 ========
;  by Jim Nagy, July 1997
;
;  A solid state thermostat, using a PIC12C508 as the measuring and control
; device. This circuit serves to demonstrate the '508s ability to measure an
; analog quantity, and make decisions based on it.
;
; The PICostat has been designed as a backup heating thermostat. It operates
; at a fixed temperature, when the measured resistance of a thermistor is
; equal to a reference resistance. Hysteresis is added on the shutoff to
; avoid hunting.
;
; Circuit connections are as follows:
; - Output is active high from GP0 (pin 7)
; - An 'activity' LED output (active high) is from GP1 (pin 6)
; - the voltage on a grounded0.1uF capacitor is monitored at GP2 through
; a 10K resistor
; - A thermistor (10K@25C) is connected from GP5 (pin 2) to the capacitor
; - A reference resistor is connected from GP4 (pin 3) to the capacitor
; - GP3 (pin 4) is configured as an active low MCLR, with internal pullup
; - +5V is connected to pin 1, gnd to pin 8
;
;  The reference resistor is 18K (providing operation at 10C or 50F)
;
; ***************************************************************************

; Program equates
  = 0026  Hyst    EQU H'26' ; Hysteresis for turnoff (about 2deg C at a 10C setpoint)
  = 0003  Cycles  EQU 3     ; #times that temp must be stable for, before output change

                            ; Standard Equates
  = 0000  W             EQU 0
  = 0001  F             EQU 1

  = 0007  GPWUF         EQU 7
  = 0005  PA0           EQU 5
  = 0004  TO            EQU 4
  = 0003  PD            EQU 3
  = 0002  Z             EQU 2
  = 0002  Zero          EQU 2
  = 0001  DC            EQU 1
  = 0000  C             EQU 0
  = 0000  Carry         EQU 0

  = 0000  MCLRDisabled  EQU 0
  = 0010  MCLREnabled   EQU H'10'
  = 0000  CodeProtect   EQU 0
  = 0008  NoCodeProtect EQU H'08'
  = 0000  WDTDisabled   EQU 0
  = 0004  WDTEnabled    EQU H'04'
  = 0002  IntRCOsc      EQU H'02'
  = 0003  ExtRCOsc      EQU H'03'
  = 0001  XTOsc         EQU H'01'
  = 0000  LPOsc         EQU 0

                            ; '508 Registers
  = 0000 INDF           EQU H'00'
  = 0001 TMR0           EQU H'01'
  = 0002 PCL            EQU H'02'
  = 0003 STATUS         EQU H'03'
  = 0004 FSR            EQU H'04'
  = 0005 OSCCAL         EQU H'05'
```

```
  = 0006 GPIO           EQU H'06'

                          ; program variables
  = 0007 TRefLo         EQU H'07'              ; Lo byte of 7uS counter - ref resistor
  = 0008 TRefHi         EQU H'08'              ; Hi byte of "  "

  = 0009 TMeasLo        EQU H'09'              ; Lo byte of measurement counter
  = 000A TMeasHi        EQU H'0A'              ; Hi byte of "  "

  = 000B OnCount        EQU H'0B'              ; delay for output turn-on
  = 000C OffCount       EQU H'0C'              ; delay for output turn-off


 ;      *********************************************
 ;       Setting the ID words...

                        ORG H'0200'
0200   0000    ID0             Data.W  H'0000'
0201   0000    ID1             Data.W  H'0000'
0202   0000    ID2             Data.W  H'0000'
0203   0005    ID3             Data.W  H'0005'

 ;      *********************************************
 ;       and the Fuses...

                       ORG H'0FFF'
0FFF   001E    CONFIG  Data.W  MCLREnabled + NoCodeProtect + WDTEnabled + IntRCOsc


 ;      *********************************************
 ;       PIC starts here on power up...
 ;      *********************************************

                        ORG H'00'

0000   0025             MOVWF   OSCCAL        ; store the factory osc. calibration value
0001   0FFF             XORLW   H'FF'         ; leave room for a patch

 ;       subroutines must be in the low page, so jump to higher memory...
0002   0683             BTFSC   STATUS,TO     ; check if we're here from WDT timeout
0003   0A24             GOTO    Init          ; no, do a full reset
0004   0663             BTFSC   STATUS,PD     ; was a timeout, but were we in sleep
0005   0A24             GOTO    Init          ; no - code error
0006   0A27             GOTO    Main          ; yes, carry on



  ;      *********************************************
  ;               Charge
  ;       Charges up the capacitor, and waits 10 time constants
  ;       GP2 is left as an OUTPUT afterward, and GP4,5 are inputs

0007   0C38      Charge  MOVLW   B'00111000' ; turn GP4 and GP5 off (inputs), and GP2 on
0008   0006      TRIS    GPIO
0009   0546      BSF     GPIO,2              ; start charging, but wait ~10mS
000A   0C0D      MOVLW   H'0D'               ; outer loop counter
000B   002A      MOVWF   TMeasHi             ; (OK to trash these regs right now)
000C   0069  ch1 CLRF    TMeasLo
000D   02E9  ch2 DECFSZ  TMeasLo,F           ; wait 256*3uS (repeated 13*)
000E   0A0D      GOTO    ch2
000F   02EA      DECFSZ  TMeasHi,F
0010   0A0C      GOTO    ch1
0011   0800      RETLW   0


;      *********************************************
```

```
;               Measure
;       Simple counting loop that waits for GP2 to go low
;       (each count is approx. 7uS)
0012   006A  Measure CLRF    TMeasHi      ; clear the counters
0013   0069          CLRF    TMeasLo
0014   0746  m1      BTFSS   GPIO,2       ; check if the cap is discharged
0015   0800          RETLW   0            ; if so, we're done
0016   02A9          INCF    TMeasLo,F    ; else, count one pass
0017   0643          BTFSC   STATUS,Zero  ; check for overflow of lo byte
0018   03EA          INCFSZ  TMeasHi,F
0019   0A14          GOTO    m1
                    ;GOTO    OOLimits     ; somethings wrong with the sensor...
 ;      ********************************************
 ;                           OOLimits
 ;       Timing count is Out of Limits! Overflow of counter
 ;       occurs at approx -110C, so circuit must
be open.
001A   0C03 OOLimits MOVLW   Cycles
001B   002B          MOVWF   OnCount      ; reset the 'on' counter
001C   006C          CLRF    OffCount     ; and pretend that we've turned off properly
001D   0004          CLRWDT
001E   0CCB          MOVLW   B'11001011'  ; switch the prescaler to /8
001F   0002          OPTION               ; (for fast LED flashing)
0020   0066          CLRF    GPIO         ; all outputs off
0021   0C38          MOVLW   B'00111000'  ; GP0-2 are outputs
0022   0006          TRIS    GPIO
0023   0003          SLEEP                ; then bail
  ;      ********************************************
  ;       Power On jumps to here...either Init, or main
  ;      ********************************************
0024   0C03 Init    MOVLW   Cycles        ; reset the counters, as we
0025   002B         MOVWF   OnCount       ; haven't made any measurements yet
0026   002C         MOVWF   OffCount
0027   0C06 Main    MOVLW   B'00000110'   ; turn the LED on and start charging the cap
0028   0646         BTFSC   GPIO,2        ; but check if the cap is already charged,
0029   0C07         MOVLW   B'00000111'   ; and if so, also turn the relay on
002A   0026         MOVWF   GPIO
002B   0C38         MOVLW   B'00111000'   ; GP0, GP1, and GP2 are outputs,
002C   0006         TRIS    GPIO          ; GP3, GP4, and GP5  are inputs
002D   0004         CLRWDT                ; OOLimits may have changed things, so...
002E   0CCF         MOVLW   B'11001111'   ; int clock to TMR0, WDT uses /128 prescaler
002F   0002         OPTION                ; no pullups, and no wakeup on change
                                          ; first, measure the reference resistor
0030   0907         CALL    Charge        ; charge up the capacitor
0031   0586         BSF     GPIO,4        ; GP4 starts at +5V
0032   0C2C         MOVLW   B'00101100'   ; make Rref (GP4) an output
0033   0006         TRIS    GPIO          ; and GP2 an input
0034   0486         BCF     GPIO,4        ; drop GP4 to 0V and measure the time
0035   0912         CALL    Measure
0036   0209         MOVF    TMeasLo,W     ; tuck away the results
0037   0027         MOVWF   TRefLo
0038   020A         MOVF    TMeasHi,W
0039   0028         MOVWF   TRefHi
```

```
 ;       now, measure the unknown resistor
003A   0907         CALL    Charge        ; get the cap ready
003B   05A6         BSF     GPIO,5        ; will need GP5 at +5V
003C   0C1C         MOVLW   B'00011100'   ; make Rtemp (GP5) an output
003D   0006         TRIS    GPIO          ; and GP2 an input
003E   04A6         BCF     GPIO,5        ; drop GP5 to 0V and measure the time
003F   0912         CALL    Measure

;       add temp offset to reading (hysteresis)if the output is currently on
0040   0706         BTFSS   GPIO,0        ; is output on?
0041   0A46         GOTO    Compare       ; no
0042   0C26         MOVLW   Hyst          ; yes, add hysteresis
0043   01E9         ADDWF   TMeasLo,F
0044   0603         BTFSC   STATUS,Carry
0045   02AA         INCF    TMeasHi,F

;       compare Tmeas to Tref...
0046   020A Compare MOVF    TMeasHi,W     ; compare hi bytes...
0047   0088          SUBWF   TRefHi,W     ; W = TRefHi - TMeasHi
0048   0743          BTFSS   STATUS,Zero  ; zero will be set if they're equal
0049   0A4E          GOTO    cmp1

004A   0209          MOVF    TMeasLo,W    ; hi's are equal - check lo's
004B   0087          SUBWF   TRefLo,W     ; W = TRefLo - TMeasLo
004C   0643          BTFSC   STATUS,Zero  ; if ><, go check carry bit
004D   0A58          GOTO    TempLo       ; else, treat as if Tmeas>Tref

004E   0703 cmp1     BTFSS   STATUS,Carry ; check the status bit
004F   0A58          GOTO    TempLo       ; if clear, Tmeas>Tref

;       Tmeas<Tref, temp is higher than ref
0050   0C03 TempHi  MOVLW   Cycles        ; reset the turn-on counter
0051   002B         MOVWF   OnCount
0052   0706         BTFSS   GPIO,0        ; is output currently on?
0053   0A60         GOTO    Done          ; no it's off, just exit
0054   02EC         DECFSZ  OffCount,F    ; yes, check the turnoff delay
0055   0A60         GOTO    Done          ; not ready yet
0056   0406         BCF     GPIO,0        ; ready - turn the output off
0057   0A60         GOTO    Done

;       Tmeas>Tref, temp is lower than the ref
0058   0C03 TempLo  MOVLW   Cycles        ; reset the turn-off counter
0059   002C          MOVWF   OffCount
005A   0606          BTFSC   GPIO,0       ; is output currently off?
005B   0A5E          GOTO    tl1          ; no it's on, exit
005C   02EB          DECFSZ  OnCount,F    ; yes, check the turnon delay
005D   0A60          GOTO    Done         ; not ready yet
005E   0506 tl1      BSF     GPIO,0       ; ready - turn output on, and charge the cap!
005F   0907          CALL    Charge       ; On powerup, outputs will be inputs. So this is
;                    GOTO    Done         ; how I can tell if GP0 should be on

;       That's all...
0060   0426 Done    BCF      GPIO,1       ; turn the LED off
0061   0003         SLEEP                 ; and shut down
```