```
'---------------Title--------------

' File......DS1620_3_heater.pbp
' Started....5/27/08
' Microcontroller used:  Microchip Technology 16F88
'                        microchip.com
' PicBasic Pro Code, micro-Engineering Labs, Inc.
'                    melabs.com

'--------Program Desciption--------

' The program uses the Dallas DS1620 digital
' temperature device as a thermostat to:
'   * Control a PIC16F88 which in turn controls a 47 ohm
'     resistor acting as a heater.
' The PIC16F88:
'   * Turns on the resistor heater at the low temperature
'     limit (TL) input from the DS1620 and turns off the
'     resistor heater at the high temperature limit (TH)
'     input from the DS1620.
'   * Reads the DS1620 device and displays results on an LCD.
'   * Displays only positive °C temperatures.
' The DS1620 measures temperatures from -55°C to +125°C
' in 0.5°C increments

'-------------Comments-------------

' Before running this program, establish the high and low
' temperature limit settings, TH and TL, using DS1620_2.pbp.
' See: http://cornerstonerobotics.org/code/DS1620_2.pbp
' Also see
'
' http://cornerstonerobotics.
org/curriculum/lessons_year2/erii_ds1620_thermometer.pdf
'
' for a more detailed description of the application of this program.
' That web page includes photos of the heater and the DS1620
' theromstat.

'-------------Includes-------------

    INCLUDE "Modedefs.bas"      ' The Mode names for SHIFTIN and
                                ' SHIFTOUT are defined in the file
                                ' Modedefs.bas.

'----------PIC Connections---------

'       16F88 Pin               Wiring
'       ---------           ----------
'       RA0                 LCD pin 11(DB4)
'       RA1                 LCD pin 12(DB5)
'       RA2                 LCD pin 13(DB6)
'       RA3                 LCD pin 14(DB7)
'       RA4                 LCD Register Select(RS)
'       RB0                 DS1620 RST (Pin 3)
```

```
'       RB1                 DS1620 DQ (Pin 1)
'       RB2                 DS1620 CLK (Pin 2)
'       RB3                 LCD Enable(E)
'       RB4                 DS1620 TH (Pin 7)
'       RB5                 DS1620 TL (Pin 6)
'       RB6                 To a NPN transistor switch that controls
'                           the 47 ohm resistor heater
'       Vdd                 +5 V
'       Vss                 Ground
'       MCLR                4.7K Resistor to +5 V

'--------DS1620 Connections--------

'       DQ (Pin 1)          PIC RB1
'       CLK (Pin 2)         PIC RB2
'       RST (Pin 3)         PIC RB0
'       GND (Pin 4)         Ground
'       TCOM (Pin 5)        No Connection
'       TLOW (Pin 6)        PIC RB5
'       THIGH (Pin 7)       PIC RB4
'       Vdd (Pin 8)         +5 V

'------------Variables-------------

    temp    VAR     WORD        ' WORD to store temperature variable,
                                ' temp

'--------DS1620 Control Pins-------

    DSRST   VAR     PORTB.0     ' Name PORTB.0 as DSRST (DS1620 Reset)
    DSDQ    VAR     PORTB.1     ' Name PORTB.1 as DSDQ (DS1620 Data)
    DSCLK   VAR     PORTB.2     ' Name PORTB.2 as DSCLK (DS1620 Clock)
    DSTH    VAR     PORTB.4     ' Name PORTB.4 as DSTH,the high temperature
                                ' limit input, TH, from the DS1620.
    DSTL    VAR     PORTB.5     ' Name PORTB.5 as DSTL,the low temperature
                                ' limit input, TL, from the DS1620.
    heater  VAR     PORTB.6     ' Name PORTB.6 as heater

'---------Initialization--------

    TRISB = %00110000   ' Set pins RB4 and RB5 of PORTB as inputs,
                        ' the remaining PORTB pins are outputs.

    ANSEL = 0           ' Configure all pins to digital
                        ' operation since not using ADC
                        ' (Analog to Digital Converter)

    OSCCON = $60        ' Sets the internal oscillator in the
                        ' 16F88 to 4 MHz

'--------Main Code--------

    PAUSE 1000          ' Pause 1 second to allow LCD to setup

    LOW DSRST           ' Reset the DS1620
```

```
' Main loop to control the heater, read temperature from the DS1620,
' and then display it on the LCD.

loop:

' Control 47 ohm resistor heater element

    IF DSTL = 1 THEN    ' If the low temperature limit (TL) input from the
                        ' DS1620 is reached, then turn on the heater
                        ' element.

    GOTO heat           ' Go to "heat" label to turn on the heater

    ELSE                ' If the low temperature limit is not
                        ' reached, then do not turn on the heater.

    GOTO noheat         ' Go to the "noheat" label to turn the heater
                        ' off and display the temperature on the LCD

    ENDIF               ' End of IF..THEN statement

heat:                   ' "heat" label

    heater = 1          ' Set PORTB.6 (heater) HIGH to turn on heater

    PAUSE 2000          ' Wait 2000 ms or 2 seconds

    IF DSTH = 1 THEN GOTO noheat
                        ' If the high temperature limit (TH) input from the
                        ' DS1620 is reached, then turn off the heater.

    GOSUB displaytemp   ' If DSTH does not = 1 then, go to the display
                        ' temperature subroutine, "displaytemp".

    GOTO heat           ' Loop back to the "heat" label

noheat:                 ' "noheat" label

    heater = 0          ' Set PORTB.6, heater, LOW to turn off heater

    GOSUB displaytemp   ' Go to the display temperature subroutine,
                        ' "displaytemp".

    GOTO loop           ' Jump to "loop" label and do it forever

END

' Display temperature on LCD

displaytemp:            ' Display temperature subroutine

' Convert temperature from DS1620

    DSRST = 1           ' Enable DS1620
```

```
    SHIFTOUT DSDQ, DSCLK, LSBFIRST, [$ee]
                          ' Send initiate temperature conversion
                          ' command, $ee, on data pin DSDQ,
                          ' synchronized by clock pin DSCLK, shift
                          ' data out lowest bit first, LSBPRE

    DSRST = 0             ' Reset the DS1620 to enable conversion

    PAUSE 1000            ' Pause 1 second to complete conversion

' Read temperature from DS1620

    DSRST = 1             ' Enable DS1620

    SHIFTOUT DSDQ, DSCLK, LSBFIRST, [$aa]
                          ' Send read command, $aa

    SHIFTIN DSDQ, DSCLK, LSBPRE, [temp\9]
                          ' Read 9-bit temperature.
                          ' Shifts in 9 bits of variable temp,
                          ' [temp\9], on data pin DSDQ,
                          ' synchronized by clock pin DSCLK,
                          ' shift data in lowest bit first,
                          ' LSBPRE

    DSRST = 0             ' Reset the DS1620

' Display temperature as a decimal

    LCDOUT $fe, 1, DEC (temp >> 1), ".", DEC (temp.0*5), " Degrees C"

                          ' Shift temp to right one position,(temp >> 1),
                          ' to display the integer portion of temp then
                          ' multiply bit 0 of temp by 5 (temp.0*5) to
                          ' display decimal portion of temp.
                          ' The bit temp.0 is either a 0 or 1,
                          ' so (temp.0*5) is either 0 or 5 proceeded
                          ' by a decimal from  the entry "."

   RETURN
```