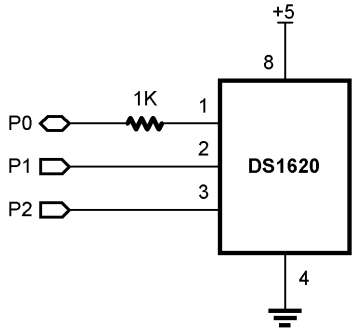
# Experiment #28: Temperature Measurement

This experiment demonstrates the use of a digital temperature sensor. Temperature measurement is a necessary component of environmental control applications (heating and air conditioning).

**Building The Circuit** (Note that schematic is NOT chip-centric)




```
' =============================================================================
'
'   File..... Ex28 - DS1620.BS2
'   Purpose... Temperature measurement
'   Author.... Parallax
'   E-mail.... stamptech@parallaxinc.com
'   Started...
'   Updated... 01 MAY 2002
'
'   {$STAMP BS2}
'
' =============================================================================


' -----------------------------------------------------------------------------
' Program Description
' -----------------------------------------------------------------------------

' This program measures temperature using the Dallas Semiconductor DS1620
' temperature sensor.
```

```
' -----------------------------------------------------------------------------
' I/O Definitions
' -----------------------------------------------------------------------------

DQ              CON     0                       ' DS1620.1 (data I/O)
Clock           CON     1                       ' DS1620.2
Reset           CON     2                       ' DS1620.3


' -----------------------------------------------------------------------------
' Constants
' -----------------------------------------------------------------------------

RdTmp           CON     $AA                     ' read temperature
WrHi            CON     $01                     ' write TH (high temp)
WrLo            CON     $02                     ' write TL (low temp)
RdHi            CON     $A1                     ' read TH
RdLo            CON     $A2                     ' read TL
StartC          CON     $EE                     ' start conversion
StopC           CON     $22                     ' stop conversion
WrCfg           CON     $0C                     ' write config register
RdCfg           CON     $AC                     ' read config register


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

tempIn          VAR     Word                    ' raw temperature
sign            VAR     tempIn.Bit8             ' 1 = negative temperature
tSign           VAR     Bit
tempC           VAR     Word                    ' Celsius
tempF           VAR     Word                    ' Fahrenheit


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

Initialize:
  HIGH Reset                                    ' alert the DS1620
  SHIFTOUT DQ, Clock, LSBFirst, [WrCfg, %10]    ' use with CPU; free-run
  LOW Reset
  PAUSE 10
  HIGH Reset
  SHIFTOUT DQ, Clock, LSBFirst, [StartC]              ' start conversions
  LOW Reset
```

```
' -----------------------------------------------------------------------------
' Program Code
' -----------------------------------------------------------------------------

Main:
  GOSUB Get_Temperature                          ' read the DS1620

  DEBUG Home
  DEBUG "DS1620", CR
  DEBUG "------", CR
  DEBUG SDEC tempC, " C    ", CR
  DEBUG SDEC tempF, " F    ", CR

  PAUSE 1000                                    ' pause between readings
  GOTO Main


' -----------------------------------------------------------------------------
' Subroutines
' -----------------------------------------------------------------------------

Get_Temperature:
  HIGH Reset                                    ' alert the DS1620
  SHIFTOUT DQ, Clock, LSBFIRST, [RdTmp]         ' give command to read temp
  SHIFTIN DQ, Clock, LSBPRE, [tempIn\9]         ' read it in
  LOW Reset                                     ' release the DS1620

  tSign = sign                                  ' save sign bit
  tempIn = tempIn / 2                           ' round to whole degrees
  IF (tSign = 0) THEN No_Neg1
  tempIn = tempIn | $FF00                       ' extend sign bits for negative

No_Neg1:
  tempC = tempIn                                ' save Celsius value
  tempIn = tempIn */ $01CC                      ' multiply by 1.8
  IF (tSign = 0) THEN No_Neg2                   ' if negative, extend sign bits
  tempIn = tempIn | $FF00

No_Neg2:
  tempIn = tempIn + 32                          ' finish C -> F conversion
  tempF = tempIn                                ' save Fahrenheit value
  RETURN
```

## Experiment #28: Temperature Measurement

### Behind The Scenes

The largest organ of the human body is the skin and it is most readily affected by temperature. Little wonder then that so much effort is put into environmental control systems (heating and air conditioning).

This experiment uses the Dallas Semiconductor DS1620 digital thermometer/thermostat chip. This chip measures temperature and makes it available to the BASIC Stamp through a synchronous serial interface. The DS1620 is an intelligent device and, once programmed, is capable of stand-alone operation using the T(com), T(hi) and T(lo) outputs.

The DS1620 requires initialization before use. In active applications like this, the DS1620 is configured for free running with a CPU. After the configuration data is sent to the DS1620, a delay of 10 milliseconds is required so that the configuration can be written to the DS1620's internal EEPROM. After the delay, the DS1620 is instructed to start continuous conversions. This will ensure a current temperature reading when the BASIC Stamp requests it.

To retrieve the current temperature, the Read Temperature ($AA) command byte is sent to the DS1620. Then the latest conversion value is read back. The data returned is nine bits wide. Bit8 indicates the sign of the temperature. If negative (sign bit is 1), the other eight bits hold the two's compliment value of the temperature. Whether negative or positive, each bit of the temperature is equal to 0.5 degrees Celsius.

The Celsius temperature is converted to whole degrees by dividing by two. If negative, the upper-byte bits are set to 1 so that the value will print properly with SDEC (signed numbers in the BASIC Stamp must be 16 bits in length). The temperature is converted to Fahrenheit using the standard formula:

$$F = (C * 1.8) + 32$$

### Challenge

Rewrite the program to write the temperature values to the StampWorks LCD module.