# PC-Based Thermometer

KISTER GENESIS M. JIMENEZ
JESSIE A. BAUL
LOPE BEN C. PORQUIS


## Abstract

*This paper addresses the problem of developing a PC-based thermometer that utilizes a Negative Temperature Coefficient (NTC) thermistor as a temperature sensor interfaced to the computer for data logging. The design was originally developed by (Osterman, 2000) and the application was written in Delphi. The researchers fabricated the original design, however they ported the application to Visual Basic. The advantage of Visual Basic is on simplicity and basic program coding. The device is intended to be used as a demo application computer interfacing and data acquisition. It was observed that on a different computer specification, a specific interpolation set of data is required to derive the relationship between temperature and charging time of the capacitor. The charging time is calculated by the software and it is important that no other applications or processes consuming significant amounts of resources is active during the calculation process.*


Keywords: PC-Based Thermometer, data acquisition, computer interfacing, RS232, NTC temperature sensor, Visual Basic.

KISTER GENESIS M. JIMENEZ, Undergraduate student, of the Department of Electronics & Communications Engineering, College of Engineering (COE) MSU-IIT. His interest is computer programming and he is from Iligan City. JESSIE A. BAUL, is an Undergraduate student of the Department of Electronics and Communications Engineering, COE, MSU-IIT, His interest is hardware development and is also from Iligan City. LOPE BEN C. PORQUIS, Assistant Professor I, Department of Electronics and Communications Engineering, COE, MSU-IIT, His interest are hardware development and robotics is from Iligan City

## Introduction

Temperature is an important factor in most of the things in the universe. It is measured by an instrument called thermometer. A thermometer is made from a material that is significantly sensitive (a sensor) to temperature and a casing to protect or handle the temperature dependent material. A scale is used to compare the temperature and the reaction of the material.

This project is aiming to fabricate a PC-based thermometer that utilizes a Negative Temperature Coefficient (NTC) thermistor as a sensor and is interfaced to the computer for data logging. This is a process of data acquisition where the change of resistance with the temperature of the thermistor will be sent, analyzed and shown on the PC.

The device will connect to the computer via the Transmit Data pin (TxD), Data Terminal Ready pin (DTR) and Data Send Ready pin (DSR) of the serial port. DTR will be enabled during temperature measurement. The thermistor and a capacitor are in loop and DTR and TxD serves as the source. The capacitor is then charged and when it reaches enough voltage to be treated as logic high for the control pins with respect to the TxD, the DSR pin will be enabled. Visual Basic handles the serial communication and count the time of charging of the capacitor. The temperature is then calculated from an equation that relates the temperature to the time of charging of the capacitor.

Since it is PC-based, the precision of the device is greatly affected by other processes in the computer that utilizes most of the CPU resources. However, if the computer's process is average, the precision could increase. The temperature measured from this device can have a better precision if the equation that relates the temperature and time of charging is formulated with as many as possible sets of ordered pair from the temperature vs. time graph in the calibration.

## Objective

To develop a PC-based thermometer using a Negative Temperature Coefficient

Thermistor on a Visual Basic Platform.

## Materials and Equipment

**Materials:**

10 μF Electrolytic Capacitor
1N4148 Diode
Male DB9 (serial connector)
2 m wire
Soldering Lead
Masking Tape
15 kΩ NTC Thermistor
Plastic Cellophane
Ice Cubes
Hot water

**Equipment:**

Household Thermometer
Personal Computer
Basin
Electric Soldering Iron
Screw Driver
Cutt

## IV. Methodology

### A. Construction of Thermometer

1. Solder the cathode of 1N4148 to pin 6 (DSR pin) of the DB9 and the anode to pin 4 (DTR pin).

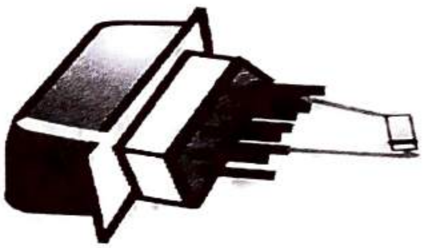

**Figure 1.** Signal Diode Assembly

2. Solder the negative leg of the capacitor to pin 3 (TXD pin) and the other leg to pin6 (DSR pin) of the DB9.

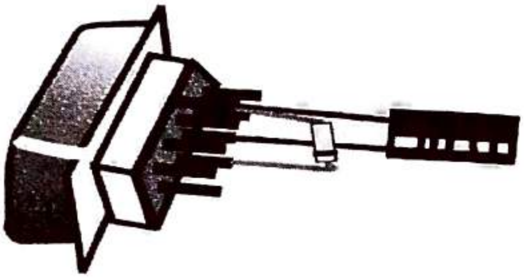

**Figure 2.** Mounting of the capacitor.

3. Cut the two-meter wire by two. Solder the end of one wire to pin 6 and one end of the other wire to pin 4 of the DB9.

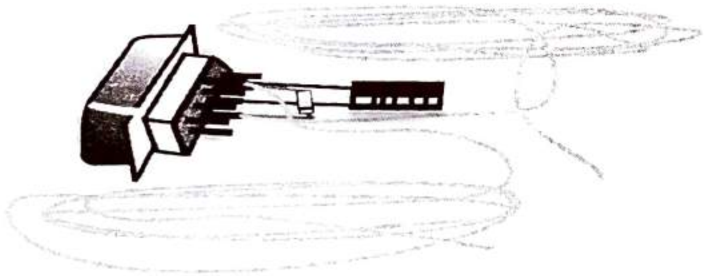

**Figure 3**. Connecting the wires.

4. Tape both wires so they stay together. Then, at the ends of both wires, connect the thermistor. You can use some improvised socket so you can remove and reattach the thermistor whenever you want.

**Figure 4.** Thermistor connections.

5. Assemble the casing of the DB9.

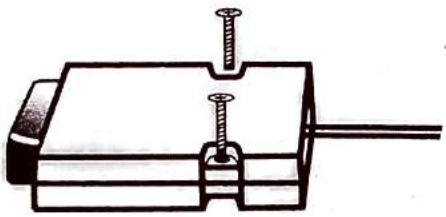

**Figure 5.** DB9 casing assembly.

## B. Software (Using Visual Basic IDE)

### B1. VB Components and Properties

Command buttons:

| NAME | CAPTION |
|---|---|
| Cmdset | "set" |
| cmdCalOk | "Ok" |
| cmdManchk | "Manual Check" |

Frames:

| NAME | CAPTION |
|---|---|
| Frame1 | "Approx. time of charging" |
| Frame2 | "Test Interval" |
| Frame3 | "Display" |
| Frame4 | "Calibrate" |

Labels:

| NAME | CAPTION |
|---|---|
| Label1 | "Seconds" |
| Label2 | "(range 20-3600)" |
| Label4 | "ms" |
| Label6 | "Temp=A*ln(1/log(chargetime(ms))) + B" |
| Label7 | "A and B are constants that may differ from computer to computer. Use a Spreadsheet to fit temp vs. 1/log(chargetime(ms)) in a logarithmic curve." |
| lbldisp | "" |

Radio Buttons:

| **NAME** | **CAPTION** |
|---|---|
| optce | "In Celsius" |
| optke | "In Kelvin" |
| optfa | "In Fahrenheit" |

Textboxes:

| **NAME** | **TEXT** |
|---|---|
| txtA | "107.6" |
| txtB | "99.12" |
| txtinterval | "20" |
| txttime | "" |

Timer:

Name = Timer1
Interval = 20000

MSComm:

Name = MSComm1
CommPort = 1

B2. User Interface

Place the component on the form like the figure below.

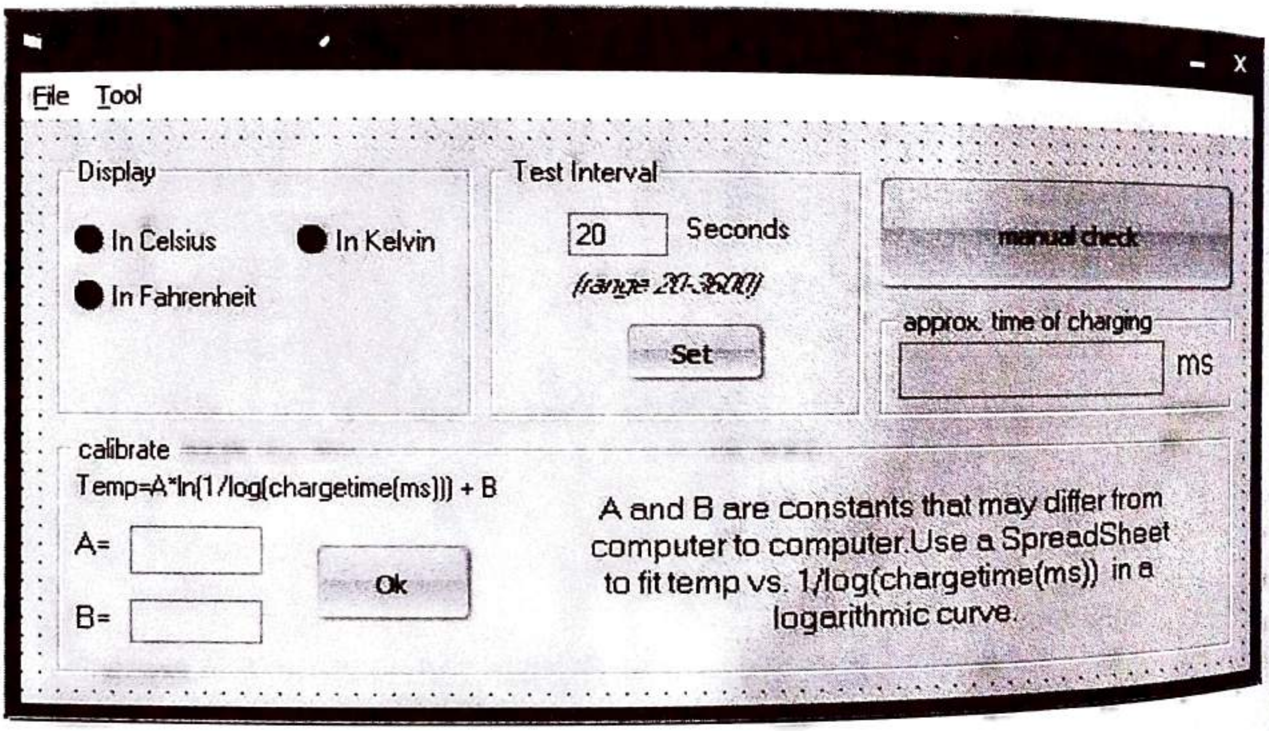


**Figure 6.** Application Screen Shot.

B3. Part of the Source Code for Checking the Temperature

```
Private Declare Sub ResetTime Lib "TimeHiRes.dll" ()
    Private Declare Function ReadTime Lib "TimeHiRes.dll" () As Double

    Call testtemp
    Call dispTemp

    Private Function Log10(Number As Double) As Double
```

```
    Log10 = Log(Number) / 2.30258509299405
End Function
Private Sub testtemp()
    MSComm1.DTREnable = True
    Call ResetTime
    While (MSComm1.DSRHolding = False)
    Wend
    ChargeTime = Str(ReadTime())
    MSComm1.DTREnable = False
    temp = A * Log(1 / Log10(ChargeTime * 1000)) + B
    txttime.Text = (ChargeTime * 1000)
End Sub

Private Sub dispTemp()
If optce.Value = True Then
    ftemp = Format(temp, "#0.00 °C")
ElseIf  optfa.Value = True Then
    ftemp = Format((temp * 9 / 5) + 32, "#0.00 °F")
Else
    ftemp = Format(temp + 237.16, "#0.00 K")
End If
lbldisp.Caption = ftemp
Form1.Caption = ftemp
End Sub
```

## C. Calibrating the Device

1. Plug the device on Serial Port at the back of the computer.
2. Place the thermistor part of the device and the household thermometer inside the plastic cellophane.
3. Put the Cellophane on the basin and cover it with Ice cubes just enough so that the markings are still visible.
4. Open the Software controller for the PC-based thermometer.
5. Wait until the reading of the household thermometer is stable.
6. Click the Manual Check, record the time of charging and the temperature (in Celsius) reading on the household thermometer. Repeat this step until sufficient data is gathered for plot the equation of temp vs. charge time.
7. Repeat steps five and six but put hot water on the basin.

8. Fit the gathered data on an exponential curve in MS Excel. Create a graph for temp vs. 1/log(chargetime) to have a natural logarithmic curve which is much simpler and accurate than the first curve. The equation should be of the form:

$$Temperature = A * \ln\left(\frac{1}{\log(ChargingTime\,(ms))}\right) + B\,(^{o}C) \quad (1)$$

9. Click tools > calibrate to input the constants on textboxes A and B and click Ok.

## Results and Discussion

The TxD pin serves as a source since when signal is sent from the PC, this pin has -12 volts. The DTR pin which initially has +12 volts (disabled) also serves as a source. However, when DTR is 12 volts, DSR which is the input to the computer and measures the voltage of the capacitor becomes 0 volts because DTR and TxD cancel each other. This is the reason why we use TxD over the System Ground pin of the serial port. When the DTR is enabled, the capacitor is charged. The time of charge varies if the resistance in series of the capacitor is not constant. The equation relating the Time of charging with the resistance is

$$RC\frac{dv}{dt} + v = v_s \quad (2)$$

when solved becomes

$$v = v_s - v_s e^{-t/RC}. \quad (3)$$

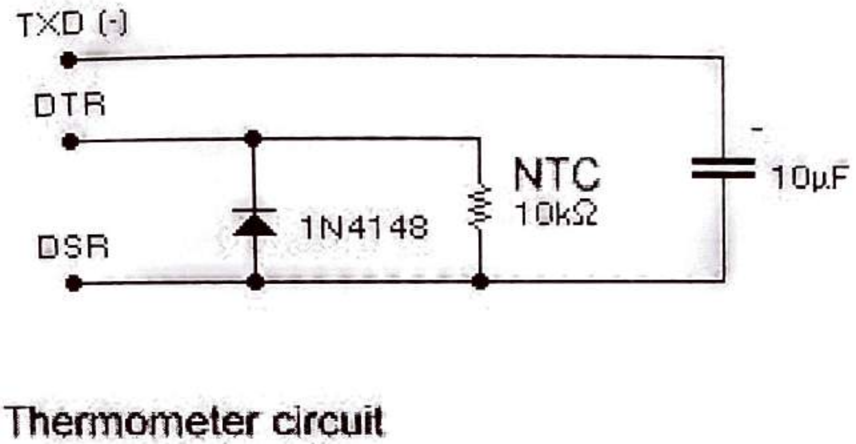


**Figure 7.** Sensor Schematic Diagram.

This equation is the basis of our method of converting the analog temperature to digital temperature. But, due to the different specification of computers, interpolation of sets of data is needed to arrive directly to a relationship between temperature and charging time of the capacitor. This process is called curve fitting.

The charging time is calculated by the software. It is in milliseconds and it is necessary that there are no other applications that consume greater resources when charging time or temperature is calculated. Because of this, connection checking of the device in the computer is omitted so that it will not interfere in the process.

**Table1.** Temperature Reading from Mercury and PC-Based Thermometer.

| Temp. of Mercury Thermometer | Temp. of PC-based Thermometer Using Chart A | Temp. of PC-based Thermometer Using Chart B |
|---|---|---|
| 31 °C | 34.0 °C | 32 °C |
| 55 °C | 51.8 °C | 55.6 °C |
| 12 °C | 9.5°C | 11.7 °C |
| 24 °C | 23.0°C | 23.8 °C |

Based on the table above, it shows that the difference of the readings of both thermometers is approximately ±3°C and ± 1°C. It shows that the equation on Chart B has less percentage of error than equation on Chart A. Each row of data was recorded in five minutes interval on different environment to ensure that the mercury thermometer is at equilibrium with the surrounding. The reading in the PC-based thermometer is based on the calibration done on a 1.5 GHz computer with Visual Basic 6 IDE, MS Excel 2007, an instance of the program itself and other processes that are necessary for the computer to operate are running.

## VI. Calibration Data

Chart A:

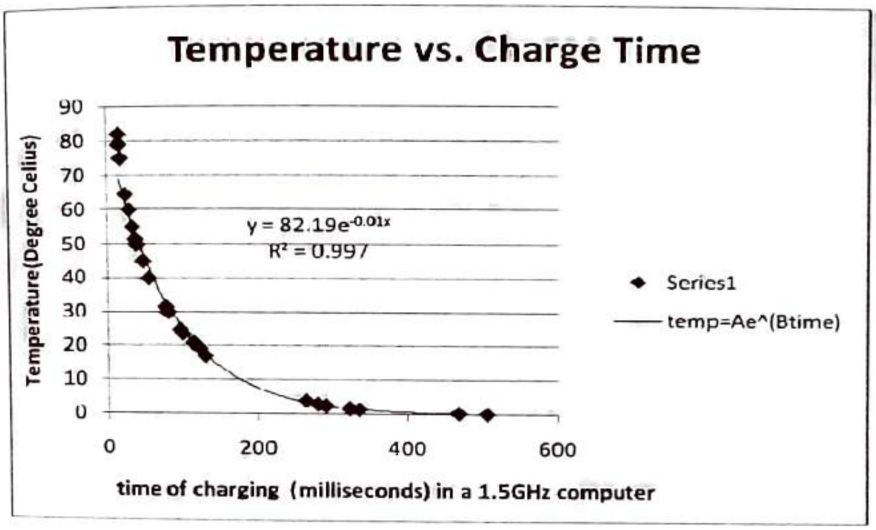


**Figure 8.** Temperature vs. Charge Time.

Chart B:

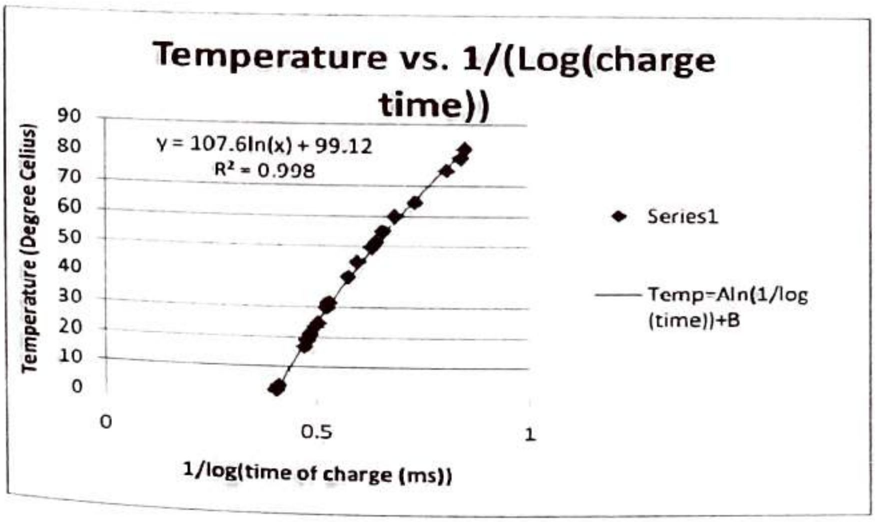


**Figure 9.** Temp. vs. 1/log(ChargeTime)

**Table2.** Calibration Data

| Time of Charge(ms) | Temperature(°C) | 1/log(charge time) |
| --- | --- | --- |
| 505.0377 | 0.2 | 0.369915 |
| 467.0139 | 0.4 | 0.374626 |
| 334.5422 | 1.5 | 0.396126 |
| 321.9106 | 1.8 | 0.398766 |
| 290.2992 | 2.5 | 0.406034 |
| 279.769 | 3 | 0.408697 |
| 263.8823 | 4.1 | 0.412982 |
| 130.0486 | 17 | 0.473013 |
| 122.621 | 19.2 | 0.478798 |
| 114.0902 | 21 | 0.486086 |
| 100.3747 | 23.7 | 0.499594 |
| 96.35797 | 24.8 | 0.504061 |
| 80.39074 | 30 | 0.524878 |
| 77.48408 | 31.5 | 0.529321 |
| 54.58107 | 40 | 0.575691 |
| 47.20685 | 45 | 0.59737 |
| 38.30194 | 50 | 0.631624 |
| 33.26961 | 55 | 0.65701 |
| 28.87211 | 60 | 0.684707 |
| 23.23655 | 64.5 | 0.731972 |
| 17.35794 | 75 | 0.806778 |
| 15.50259 | 79 | 0.840051 |
| 15.0283 | 82 | 0.849683 |
| 36.50384 | 51.5 | 0.640066 |
| 38.59248 | 49.8 | 0.630317 |

## Conclusion

A PC-based temperature measuring device has been developed in this project. The device was originally designed by (Oysterman, 2000) was the basis for this project. Its application interface based on Delphi was ported to Visual Basic. The accuracy of the system depends on the Personal Computer being used and the applications that are currently running. The coefficients resulting from the calibration process will also depend on the machine platform. The design of the sensing device was very simple and techniques in differential equations were used to calculate the temperature. Such simplicity makes this device suitable for showing how computer interfacing is carried out.

## References

Hitmil. (2007). *Visual Basic 6.0 Programming*. Available: http://www.hitmill.com/programming/vb.htm. Last Accessed: October 2007.

Oysterman, N.(2000). *Simple PC Thermometer*. Available: http://www.electronics-lab.com/projects/pc/013. Last Accessed: September 2007.

Peacock , C. (1998).*Interfacing the Serial / RS232 Port V5.0*. Available: http://www.senet.com.au/~cpeacock. Last Accessed: September 2007.

Rowe, M. (2002). *Temperature Measurement for Everyone*. Available: http://www.tmworld.com/article/CA187382.html. Last Accessed: October 2007.

Vishay Dale. (2002). *Selecting NTC Thermistors*. Available: http://www.physics.ucdavis.edu/Classes/Physics116/P116C_lab/seltherm.pdf. Last Accessed : October 2007.