

PROGRAMMING THE PALM TUNGSTEN: THIRD IN A SERIES

Edwin G. Wiggins
Webb Institute

Introduction

The first article in this series[1] describes the creation of a simple applet for the Palm Tungsten in the programming language PocketC. That applet is a Palm Database (pdb) file. It will run only within the PocketC environment. The second article[2] describes the creation of a more complicated applet as a stand-alone (prc) file.

This article describes an applet that solves the ideal gas law for a variety of gases. It handles three cases which correspond to the three possible unknowns: P, T, and v. This is a rather simple applet, but one that can be very useful to engineering students and even practicing engineers.

Mathematical Background

Readers are, no doubt, familiar with the ideal gas law, which is commonly written

$$Pv = RT,$$

where P is the absolute pressure, v is the specific volume, R is the specific gas constant for the gas being analyzed, and T is the absolute temperature. The specific gas constant is, of course, the universal gas constant divided by the molar mass of the gas.

The Ideal Gas Applet

As in the previous articles, the program code is presented in segments below. Each segment is discussed in turn. The code itself is heavily commented to make it almost self-explanatory. Many sections of this code are similar to the code discussed in the previous articles. These sections are discussed only briefly here. The discussion below focuses on the use of the

SWITCH command, which is used three times in this applet: to determine which input values should be requested, to select the appropriate molar mass, and to designate which algebraic form of the ideal gas law should be used.

The first segment contains the commands required to create a prc file.

```
// Ideal Gas Law
// Program date: 2/19/2005

@cid "idg!";
@ver "1.0";
@category "Engineering";
@name "Ideal Gas";
@dbname "Ideal Gas";
@licon1 "IDGBIG.bmp";
@licon8 "IDGBIG.bmp";
@sicon1 "idgsmall.bmp";
```

At the beginning of the program itself, all variables are declared.

```
main() {

// Declare variables

float P; // pressure
float v; // specific volume
float T; // temperature

float R; // universal gas constant

float M; // working value of molar mass
float M1; // molar mass of air
float M2; // molar mass of CO2
float M3; // molar mass of CO
float M4; // molar mass of He
float M5; // molar mass of H2
float M6; // molar mass of CH4
float M7; // molar mass of N2
float M8; // molar mass of O2

string gas; // name of gas
string unk; // symbol for unknown property
```

Next, the user is asked to input the symbol for the gas to be analyzed.

```

// Get input
clear();
title("Ideal Gas");

alert("The following gases are supported;
\n Air,CO2,CO,He \n H2,CH4,N2,O2");
gas = gets("Gas ?");

```

An error trap is used in case the user inputs the symbol of a gas that is not supported. The PocketC syntax for “not equal to” is “!=”. The syntax for “and” is “&&”. Thus, if the user inputs a symbol that is not recognized, the program displays an error message and exits.

```

// error trap in case user enters unsupported gas

if ( gas != "Air" && gas != "CO2" && gas !=
    "CO" && gas != "He" && gas != "H2"
    && gas != "CH4" && gas != "N2"
    && gas != "O2") {
    puts("\n\n That gas is NOT supported.");
    wait();
    exit();
} //end if

// end of error trap

```

The user is asked to indicate which factor in the ideal gas equation is unknown.

```

unk = gets("Unknown? (P,v,T)");
// which property is unknown

```

PocketC is case sensitive. That is V and v are different variables. To avoid confusion, the user’s input is always converted to upper case by the “strupr()” command.

```

unk =strupr(unk);

```

The user may make a mistake in entering the symbol for the unknown factor, so an error trap is inserted to detect this. If the unknown factor is anything but P, V, or T, the program displays an error message and exits.

```

// error trap in case user enters other than P or v or T

if ( unk != "P" && unk != "V" && unk != "T") {
    puts ( "Please enter P or v or T." );
    wait();
    exit();
}

```

```

} // end of error trap

```

There are three cases to be handled, because the unknown can be P, V, or T. The switch command is used to select the appropriate case. The syntax of this command is shown below. Choice of the case to be executed is based on the value of the variable “unk”.

```

switch (unk) {

```

If the user has indicated that the pressure is unknown, this portion of the code is executed.

```

case "P":
    v = (float) gets("v= ? (ft^3/lbm)");
    T = (float) gets("T= ? (R)");
    break;

```

If the user has indicated that the volume is unknown, this portion of the code is executed. The user is instructed to enter pressure in psia, but this is converted immediately to pounds per square foot.

```

case "V":
    P = (float) gets("P= ? (psia)");
    P = P*144; // convert to psfa
    T = (float) gets("T= ? (R)");
    break;

```

If the user has indicated that the temperature is unknown, this portion of the code is executed. Again, pressure is entered in psia but converted to pounds per square foot.

```

case "T":
    v = (float) gets("v= ? (ft^3/lbm)");
    P = (float) gets("P= ? (psia)");

    P = P*144; // convert to psfa
    break;

```

```

} // end switch

```

```

// Perform calculations

```

Molar masses of the eight supported gases are specified, and the value of the universal gas constant is also specified.

```
// molar mass values
```

```
M1 = 28.97; // Air  
M2 = 44.04; // CO2  
M3 = 28.01; // CO  
M4 = 4.003; // He  
M5 = 2.016; // H2  
M6 = 16.04; // CH4  
M7 = 28.01; // N2  
M8 = 32.00; // O2
```

```
R=1545; // universal gas constant in ft*lb/lbmole*R
```

Based on user input, the appropriate molar mass value is selected from the list above.

```
// choose the appropriate molar mass
```

```
switch (gas) {  
    case "AIR" :  
        M=M1;  
        break;  
  
    case "CO2":  
        M=M2;  
        break;  
  
    case "CO":  
        M=M3;  
        break;  
  
    case "He":  
        M=M4;  
        break;  
  
    case "H2":  
        M=M5;  
        break;  
  
    case "CH4":  
        M=M6;  
        break;  
  
    case "N2":  
        M=M7;  
        break;  
  
    case "O2":  
        M=M8;  
        break;  
} //end switch
```

The SWITCH command is used to select the appropriate working equation.

```
// choose the appropriate working equation
```

```
switch (unk) {  
    case "P":  
        P = (R/M)*(T/v);  
        break;  
  
    case "V":  
        v = (R*T)/(P*M);  
        break;  
  
    case "T":  
        T = (P*v)/(R/M);  
        break;  
} // end switch
```

Pressure is converted back to psia, and values of all variables are displayed.

```
P=P/144; // convert back to psia
```

```
// Display output
```

```
clear();  
puts ( "\n\n  gas is " + gas + "\n\n" );  
  
puts ( "Molar mass= " + format(M,3) + "\n");  
puts ( "Gas constant= " + format(R/M,2) + " ft*lb/lbm*R \n\n");  
  
puts ( "Pressure= " + format(P,2) + " psia \n");  
puts ( "Volume= " + format(v,2) + " ft^3/lbm \n");  
puts ( "Temperature= " + format(T,1) + " R");  
  
wait();  
exit();  
} // End main
```

Conclusion

The applet presented here may be useful to students and practitioners of engineering. In addition it illustrates the use of the SWITCH command. The next article in this series will describe an electronic air table for the Palm Tungsten.

References

1. Wiggins, Edwin G., "Programming the Palm Tungsten," Computers in Education Journal, October – December 2005, Vol. XV, No. 4, pages 53-57.
2. Wiggins, Edwing G., "Programming the Palm Tungsten Second in a Series," Computers in Education Journal, January – March 2006, Vol. XVI, No. 1, pages 24-28.

Biographical Information

Edwin G. Wiggins holds BS, MS, and Ph.D. degrees in chemical, nuclear, and mechanical engineering respectively from Purdue University. He is the Mandell and Lester Rosenblatt Professor of Marine Engineering at Webb Institute in Glen Cove, NY. Ed is a past chairman of the New York Metropolitan Section of the Society of Naval Architects and Marine Engineers (SNAME) and a past regional vice president of SNAME. A Centennial Medallion and a Distinguished Service Award recognize his service to SNAME. As a representative of SNAME, Ed Wiggins serves on the Engineering Accreditation Commission Board of Directors of the Accreditation Board for Engineering and Technology ABET.

ASEE MEMBERS

How To Join Computers in Education Division (CoED)

- 1) Check ASEE annual dues statement for CoED Membership and add \$7.00 to ASEE dues payment.
- 2) Complete this form and send to American Society for Engineering Education, 1818 N. Street, N.W., Suite 600, Washington, DC 20036.

I wish to join CoED. Enclosed is my check for \$7.00 for annual membership (make check payable to ASEE).

PLEASE PRINT

NAME: _____

MAILING ADDRESS: _____

CITY: _____

STATE: _____

ZIP CODE: _____