Introduction

This manual describes the parts, programming and use of the photovoltaic performance monitoring system (PVPM). PVPM is an automated system for collecting data from a PV system. This system has been developed for, and is in use at the Solar Performance Testing Unit of the PV Research Lab of the Department of Physics, Moi University. In this manual, the various parts comprising the system and how they function are described.

Any questions, comments or corrections may be addressed to the author at marymueni@angelfire.com.

Parts

The PVPM system comprises three main components; the CPC/PV system, the interfacing component and the software component. The CPC/PV system comprises a PV panel equipped with low-concentration parabolic reflectors. The interfacing component includes a constant voltage electronic bridge, an IO card and a PC. The software component is a program that collects and interprets data from the CPC/PV system via the interface.
The bridge for which this program was designed is a constant voltage bridge. This means that it keeps the voltage reading across the panel at a constant value so that only the current varies. Moreover, the bridge converts the current reading into a voltage reading and steps down the voltages so that they can be fed into the computer. Conversion factors are provided with the bridge to enable the user to convert the readings back to the actual voltage and current values. These conversion values are:

Current: Multiply by 14.8
Voltage: Multiply by 8.5

The IO card for this program is a Computer Boards card. The details concerning this card are comprehensively covered in the technical manuals supplied with the card.
The output from the card is in **counts**, which the PVPM program then converts to voltages before the bridge conversion factors are applied.
The card comes with its own installation software, which is used for configuration purposes. Libraries for different programming languages are also provided in the Universal Library diskette. For this case, the software for the data access has been written in Visual Basic 3.0. The program has been left in its raw form (not compiled as stand alone) so as to enable future modifications when necessary. The setback with this, however, is that the program can only be run in a PC that has the Visual Basic 3.0 program installed and all the necessary files must be manually loaded. Details of this are given in the Technical manual that accompanies this user manual.

The user manual gives a screen-by-screen description of how to use the PVPM software while the technical manual gives the source code for the software.

PVPM

USER MANUAL

The Solar Energy Research Group, Department of Physics, Moi University
P.O. Box 1125, Eldoret, Kenya

Introduction

Three of the card's channels are in use for this program, one for irradiance, the other voltage and the third current. The PVPM program can be run in two modes, the single channel mode and the multi-channel mode. The single channel mode allows the user to read only one channel at a time while the multi channel mode reads all three channels and processes the data. In both modes the data is saved to a data file specified by the user.

When the program begins, a screen as shown in fig. 1.1 is displayed. This is the main starting screen from which the two modes of the program are started.

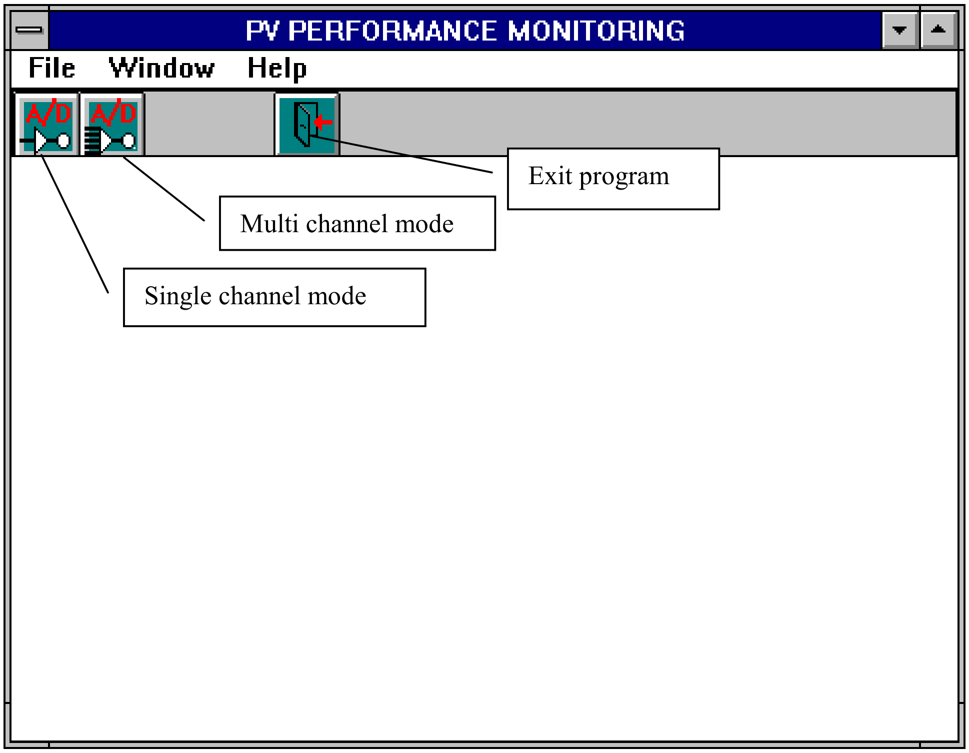
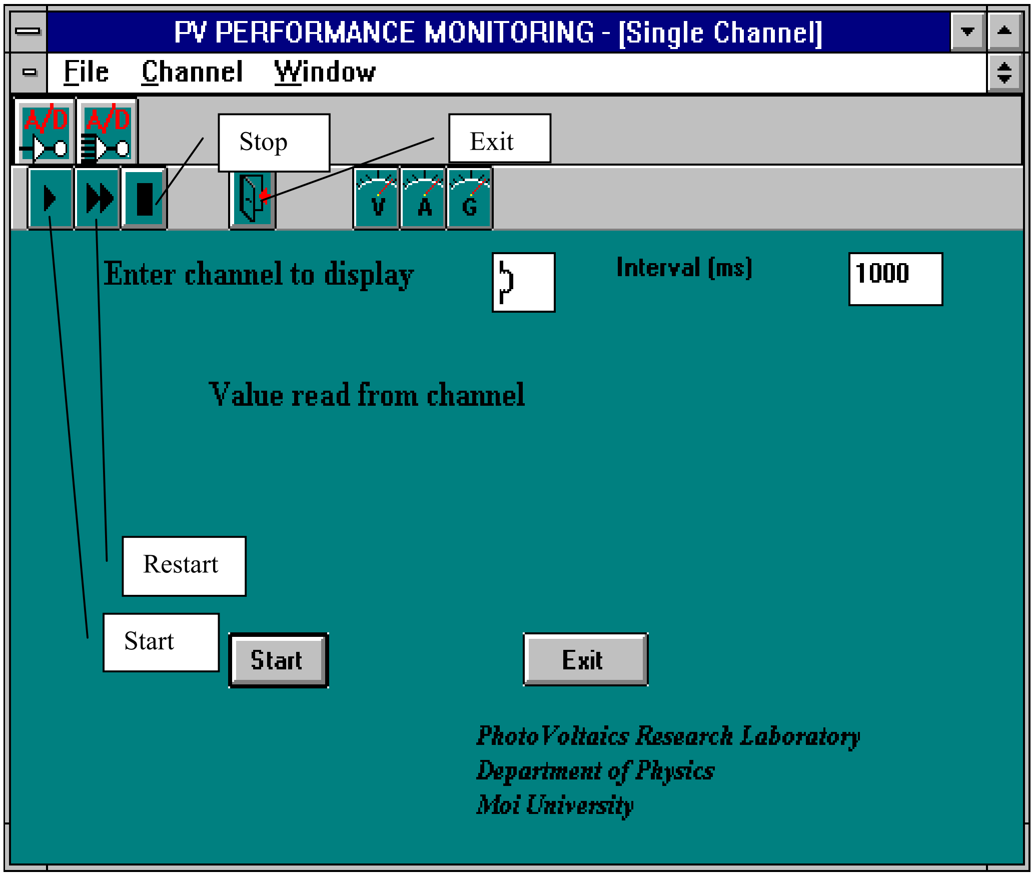


Fig. 1.1 The opening screen

By pressing the icon indicated, one can start the program in single channel or multi channel mode. Alternatively, the File menu can be used. Similarly, one can quit the program from the icon shown or from the File menu.

1.1 Single Channel Mode

When the user opts for the single channel mode, the screen shown in figure 1.1.1 appears. The user then runs the program by: pressing the "start" button, pressing the "start" icon on the toolbar or using the Start item on the File menu.



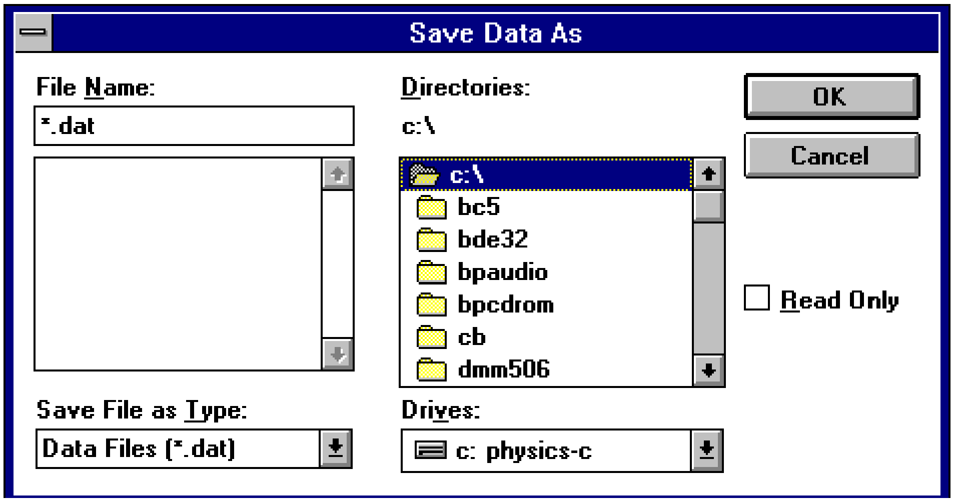
Enter channel to display

Value read from channel

Interval (ms) 1000

Fig. 1.1.1 The Single channel mode

Immediately the program begins, the user will be prompted to give a name for the file in which the data collected will be saved, as shown in figure 1.1.2.



File Name: *.dat

Directories: c:\

□ Read Only

Save File as Type: Data Files (*.dat)

Drives: c: physics-c

Fig. 1.1.2 Prompt to save data

The data file will have the extension .DAT and can then be accessed and imported into any desired software for analysis. The data is saved in comma-separated format (CSV).

Once the user has specified the name and location of the file, the program then begins running. The user should also specify the channel he desires to read by entering the channel number in the "Enter Channel to Display" box. The channels available presently are:

0	**Voltage**
1	**Current**
2	**Irradiance**

The default channel is 0. If the box is left empty, channel 0 is assumed. If a number between 2 and 8 is entered, a self-explanatory error message appears warning the user that these channels are not available. If a number greater than 8 is entered, a Computer Boards error is generated and the program aborts. This is because the card only has eight channels and only these are supported.

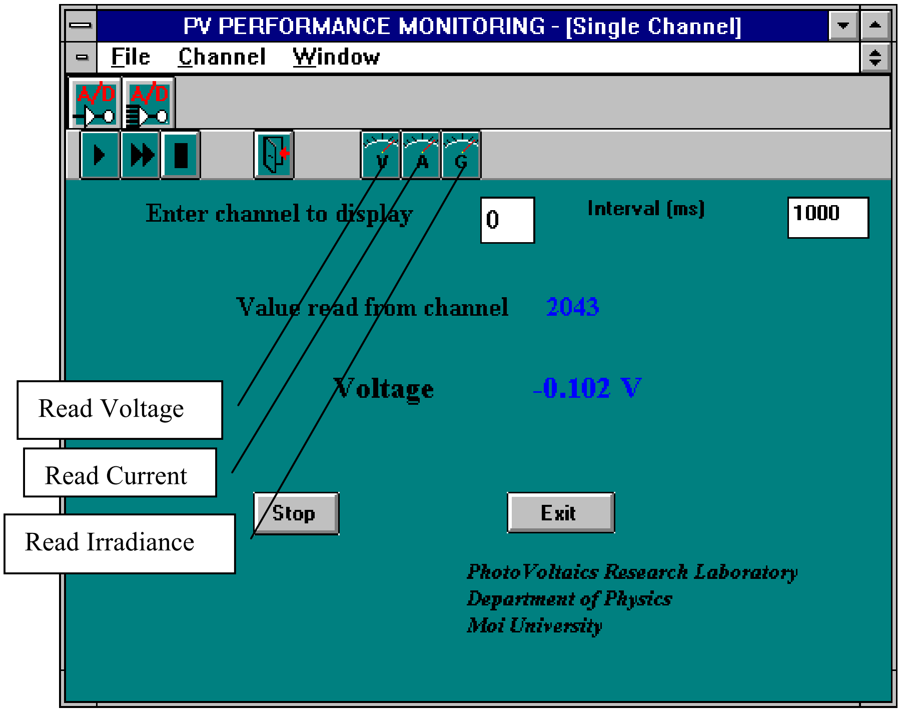
If the user presses OK then the program resumes and waits for user input. If the user presses HELP a help screen appears showing the user the available channels. To avoid confusion, the user is advised to use the icons on the toolbar to select the channel to be read. These are indicated in figure 1.1.3.

Once an acceptable channel is selected, the program runs showing both the name of the channel being monitored and the value being read and the units. Voltage will be in Volts, current in Amperes and irradiance in Watts per square meter. The rate of data collection, i.e., the interval between consecutive data points is set by default at 1000ms but this can be adjusted by changing the value in the "Interval" box.

Figure 1.1.3 shows a sample screen of the PVPM program running in Single channel mode.

To stop the program, the user can press the Stop button, press the Stop icon ▮ or use the File menu. A Restart button appears and can be used to restart the data access process.

Similarly, the Restart icon ⏩ on the toolbar or the Restart item on the File menu can be used. Restarting the program does not interfere with the data that is already saved on file. The new data is simply added at the end of the previous data and so no data is lost. The saved data has a column indicating time and so the user will be able to tell when the program was stopped and when it was restarted.



PV PERFORMANCE MONITORING - [Single Channel]

Enter channel to display 0

Interval (ms) 1000

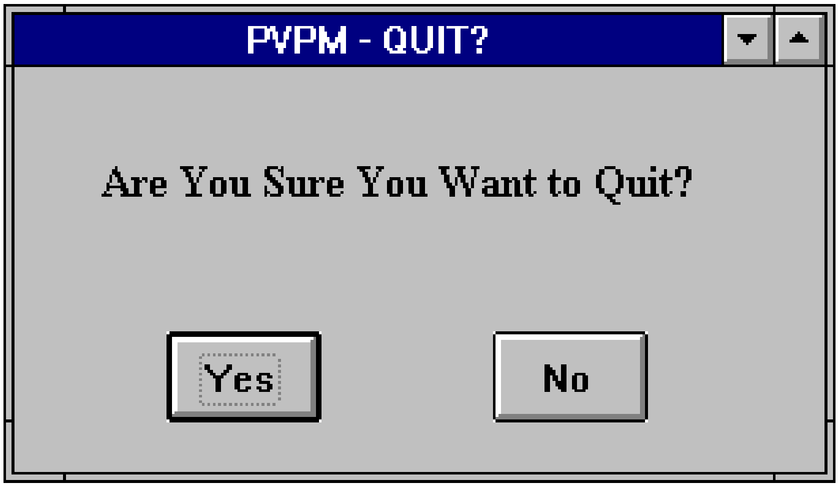
Value read from channel 2043

Voltage -0.102 V

PhotoVoltaics Research Laboratory
Department of Physics
Moi University

Figure 1.1.3 PVPM program running in Single channel mode.

To exit the program, it is advisable to first stop the program and then press the exit button or press the exit icon or use the File menu. The program will prompt the user to confirm that he wishes to Quit. This is illustrated in figure 1.1.4.



PVPM - QUIT?

Are You Sure You Want to Quit?

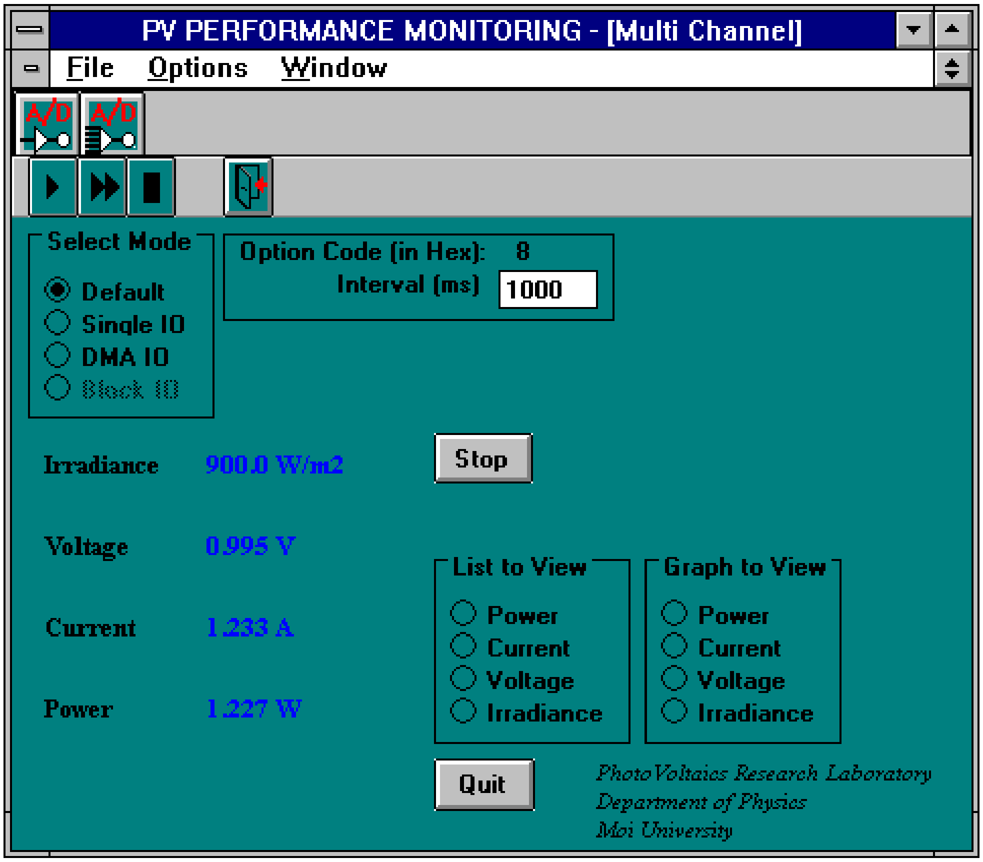
Figure 1.1.4 Confirm that you wish to Quit.

If the user presses yes, the Single channel mode exits and the user is returned to the main PVPM program (figure 1.1). If No is chosen, the user is taken back to the Single Channel mode.

1.2. Multi Channel Mode

When the user opts for the multi channel mode, the screen shown in figure 1.2.1 appears. The user then runs the program by: pressing the "start" button, pressing the "start" icon on the toolbar or using the Start item on the File menu. The mode of data access should be left at the Default. Details on the other modes can be obtained from the manuals supplied with the card. As in the case of the Single channel mode, the interval between consecutive readings can be varied using the interval box.

The multi channel program works in a similar way to the single channel mode in the case of starting, restarting, stopping, quitting and saving data. The buttons, icons or File menu options may be used for these purposes.


PV PERFORMANCE MONITORING - [Multi Channel]
File
Options
Window
A/D
A/D
Stop
Quit
PhotoVoltaics Research Laboratory
Department of Physics
Moi University

Select Mode
● Default
○ Single IO
○ DMA IO
○ Block IO

Option Code (in Hex): 8
Interval (ms) 1000

Irradiance 900.0 W/m2
Voltage 0.995 V
Current 1.233 A
Power 1.227 W

List to View
○ Power
○ Current
○ Voltage
○ Irradiance

Graph to View
○ Power
○ Current
○ Voltage
○ Irradiance

Figure 1.2.1 The PVPM program in multi channel mode

The three channels are read simultaneously and the values of voltage and current multiplied to give the power output in Watts. This data is again saved in the file specified by the user (figure 1.1.2).

The user can choose to view a list or plot of the data as it is being collected. Only one or list or plot can be viewed at a time by selecting the desired option from the List/Graph to View options.

PVPM

TECHNICAL

MANUAL

Introduction

The PVPM program has been written in Visual Basic 3.0. The program has been left in an uncompiled ("raw") state. This is for two main reasons:

1. It makes it easier to modify the code for any adjustments made to the set up.
2. The card comes with its own software, which would need to be incorporated into the resulting stand-alone program.

The main setback to leaving the program in this manner is that it can only be run in a computer that has Visual Basic 3.0 installed.

In case there is a need to transfer the program to another computer, the following steps must be followed:

1. Ensure that Visual Basic 3.0 is loaded onto the computer you are transferring the program to.
2. Create a folder to hold the program and all the files that are required to run it, as listed below.
3. Copy the project file together with all the files into the folder.
4. Open the project and ensure that it is running. You can then create an executable file (File|Make exe file) and use this to create a shortcut to the program.

The following files (List 2.1) are n ecessary for the program to run. These are the files that were created specifically for this program.

List 2.1 Files for the PVPM program

Actual name of file in Computer	Visual basic name
ABPERF.FRM	frmperAbout
CHANERR.FRM	frmError
CONFIRM1.FRM	frmConfPerf1
CONFPERF.FRM	frmConfPerf
DATAIRR.FRM	frmDataListI
DATALC.FRM	frmDataListC
DATALIST.FRM	frmDataListP
DATAV.FRM	frmDataListV
GRAPHC.FRM	frmGraphC
GRAPHI.FRM	FrmGraphI
GRAPHV.FRM	FrmGraphV
GRAPHP.FRM	FrmGraphP
HELPPERF.FRM	frmDD1Help
LOADBMP.FRM	FrmLoadBmp
MDIBASE.FRM	MdifrmBase
PERF1.FRM	frmDataDisplay1
PERF2.FRM	FrmDataDisplay
PERF4.FRM	FrmProgInfo
CBW.BAS	

The last file, CBW.BAS, comes with the Universal Library for Visual Basic and MUST be loaded into any project that runs with th e card. List 2.2 gives the Visual Basic controls that are necessary for the program to run. These are mostly located in the Windows/System folder in the computer.

List 2.2 Visual basic controls for the PVPM program

ANIBUTON.VBX
CMDIALOG.VBX
CRYSTAL.VBX
GAUGE.VBX
GRAPH.VBX
GRID.VBX
KEYSTAT.VBX
MSCOMM.VBX
MSMASKED.VBX
MSOLE2.VBX
MSOUTLIN.VBX
PICCLIP.VBX
SPIN.VBX
THREED.VBX

Some of these load automatically whenever a new project is started in VB. Some have to be added to the project manually using the File | Add File menu.

The Code

This section gives the code for the entire PVPM program. Each interface is shown followed by the code for that interface.
Note that

- Each interface has various objects (text boxes, labels, timers, etc.) and each object has it's own section of code. Some objects may have more than one section of code depending on how many of the methods belonging to that object have been utilized.
- The beginning and end of each section of code (**Sub** and **End Sub**) are in bold while comments are in italics.
- Comments always have an apostrophe (´) preceding them and this instructs the compiler to ignore them.
- Each comment explains the line(s) of code immediately following it, but in some cases the comment is written immediately after the code, on the same line.
- Some sections of code are blank.
- Note that an underscore (_) indicates that the line of code continues onto the next line. The visual basic editor supports long sentences that are wrapped in Word.
- Procedures: In Visual basic,the word **Sub** precedes any procedure and the words **End Sub** mark the end of that procedure. The word immediately following the word Sub is the name of the procedure. For example

```
Sub Objectname_Event
        …code here…
End Sub
```

where ***Objectname*** is the name of the object, e.g a command button, a text box, etc. and ***Event*** is the name of the event that will trigger the execution of the code, for example Click, Change, etc.

- Within the code, properties of objects are set using the VB format as below:

  Objectname.Property =Value

  where ***Objectname*** is the name of the object, ***Property*** is the property of that object that you wish to set, e.g Visible, Hide, etc., and ***Value*** is the value that the object will take, for example False, True etc. The exception to this rule is in the case where obj ects being referred to are in a different child form, in which case the name of the form is first given followed by the name of the object and then the property. (The mdifrmbase form is the parent form and all the rest are child forms). For example

  frmdatadisplay1.lblshowparameter.Visible = 0

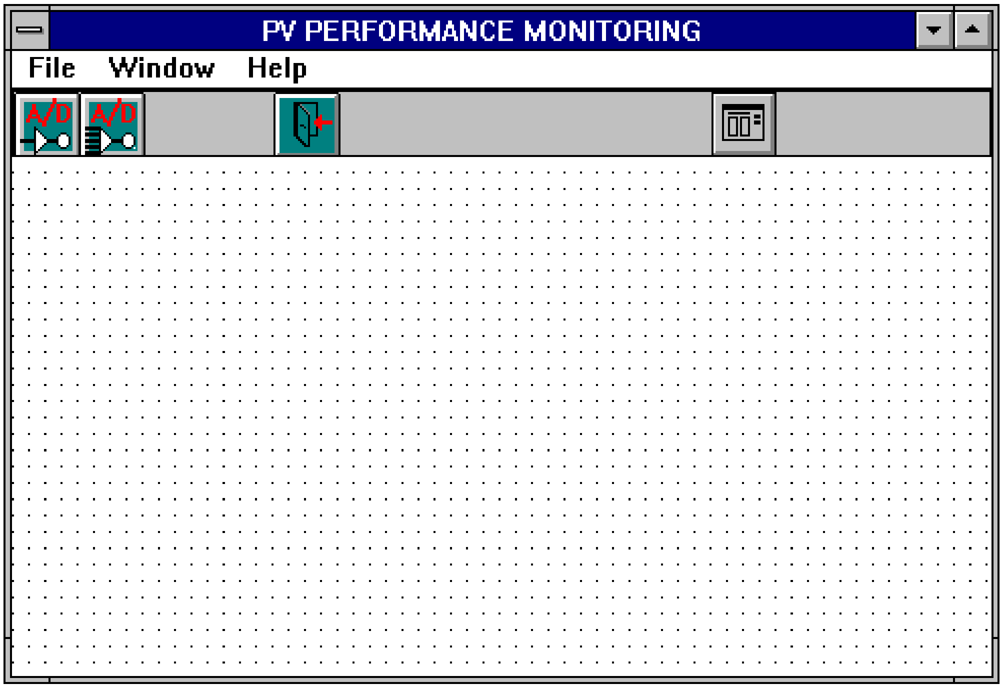
  where frmdatadisplay1 is the name of the form, lblshowparameter is the name of an object on that form and Visible is the name of the property being set.
- A naming convention has been used for ease of identifying object s. The names of the objects used begin with a prefix that identifies the type of object. The most commonly used are summarized below.

Prefix	Object
cmd	Command button
cmd3D	3D command button
frm	Form
lbl	Label
mnu	Menu item
tmr	Timer
txt	Text box

The most important user interfaces for the PVPM program are the main startup interface (fig.1.1.1) called MdifrmBase, the single channel interface (fig. 1.2.1) called frmDataDisplay1 and the multi channel interface (fig. 1.3.1) called frmDataDisplay. All other files are called from these three. We therefore begin by giving the code for these three and then give the code for the rest of the interfaces in alphabetical order.

(All files are referred to by their actual names with their visual basic object na mes in brackets).

Code 2.1 MDIBASE.FRM (MdifrmBase)



Declarations:

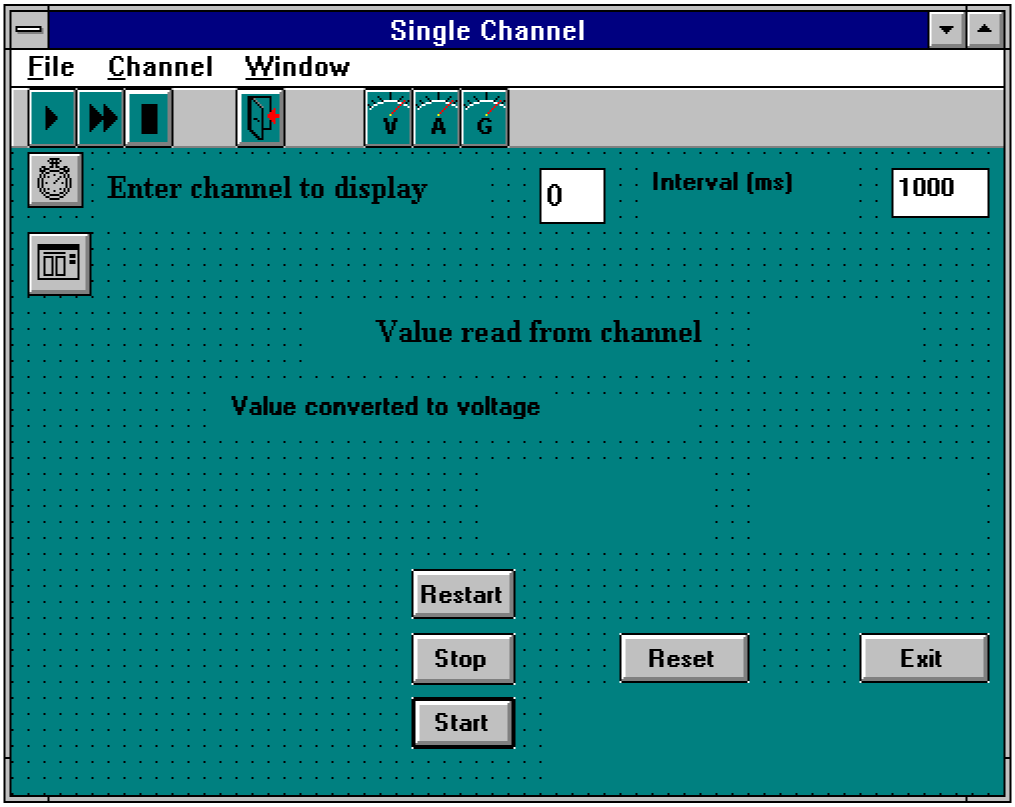
```
' File:                    PVPM.MAK

' Library Call Used:       cbAIn%(), cbAInScan%() sampling mode options.

' Purpose:                 i. Reads a user specified A/D Input Channel.
'                         ii.Scans a range of A/D Input Channels and
'                         stores the sample data in an array using
'                         a user specified sampling mode.
' Current Output:         i. Displays the analog input on a user-specified
'                            channel.
'                         ii.Displays the analog input on 3 channels.

' Other Library Calls:    cbErrHandling%()

' Special Requirements:   Board 0 must have an A/D converter.
'                         Board must support the mode selected for (ii).

' (c) Copyright 2000, PVLab.
' All rights reserved.
'=====================================================

Sub MDIForm_Load ()

End Sub
```

```
Sub cmdExit_Click ()
   ' Exit the program
   End
End Sub

Sub cmdMulti_Click ()
   ' Open the Multi Channel monitoring form
   frmDataDisplay.Show

   ' Hide the main exit icon to prevent accidentally closing the
   ' program while it is still executing
   cmdExit.Visible = False
End Sub

Sub cmdSingle_Click ()
   ' Open the Single Channel monitoring form
   frmDataDisplay1.Show
   ' Hide the main exit icon to prevent accidentally closing the
   ' program while it is still executing
   cmdExit.Visible = False
End Sub

Sub mnuFile_Click ()

End Sub

Sub mnuFileDash_Click ()

End Sub

Sub mnuFileExit_Click ()
   ' Exit the program
   End
End Sub

Sub mnuFileMulti_Click ()
   ' Open the Multi Channel monitoring form
   frmDataDisplay.Show

   ' Hide the main menu exit option to prevent accidentally closing the
   ' program while it is still executing
   cmdExit.Visible = False
End Sub

Sub mnuFileSingle_Click ()
   'Open the Single Channel monitoring form
   frmDataDisplay1.Show

   ' Hide the main menu exit option to prevent accidentally closing the
```

```
  '  program while it is still executing
  cmdExit.Visible = False
End Sub

Sub mnuHelp_Click ()

End Sub

Sub mnuHelpAbout_Click ()

  '  Display the About information by sending the following text to the label lblAbout in the
 '  form frmPerfAbout and then display the form.
  CaptionText1 = "This program has been developed for the Photovoltaic Research Group by"
  CaptionText2 = "        M Mueni and M Mw amburi"
  CaptionText3 = "        The PhotoVoltaics Research Group"
  CaptionText4 = "        Department of Physics"
  CaptionText5 = "        Moi University"
  CaptionText6 = "        P.O.Box 1125"
  CaptionText7 = "        Eldoret"
  CaptionText8 = "        Kenya."
  frmPerfAbout.lblAbout.Caption = CaptionText1 & Chr$(10) & Chr$(10) & CaptionText2 & _
    Chr$(10) & CaptionText3 & Chr$(10) & CaptionText4 & Chr$(10) & CaptionText5 & _
    Chr$(10) & CaptionText6 & Chr$(10) &  CaptionText7 & Chr$(10) & CaptionText8
  frmPerfAbout.Show
End Sub

Sub mnuHelpInfo_Click ()
'    Display information about the program by sending the following text to the label lblProgInfo
'    in the form frmProgInfo and then display the form.

  CaptionText1 = "This program allows you to do performance monitoring of a PV system._
     Three channels are available; these channels are Irradiance, Voltage, and Current."
  CaptionText2 = "Two monitoring options are available.  The 'Si ngle Channel' Option _
   allows you to monitor one channel at a time while with the 'Multi Channel' option you _
   can monitor all three channels simultaneously.  The second option also shows the _
   calculated total power output at each instan t and can display a list as well as a plot of any_
   chosen parameter."
  CaptionText3 = "  In both cases, the data collected is saved to a data file specified by the _
   user.  This allows one to afterwards analyze and manipulate the data as d esired."
  CaptionText4 = "The user is allowed to specify the time interval between consecutive data_
   collection times.  This is provided for in the box designated 'Interval'.  Note that the figure_
   you specify is in milliseconds.  The defaul t is 1000 milliseconds, i.e. 1 second.  "
  frmProgInfo.lblProgInfo.Caption = CaptionText1 & Chr$(10) & Chr$(10) & CaptionText2_
    & Chr$(10) & Chr$(10) & CaptionText3 & Chr$(10) & Chr$(10) & CaptionText4 &_
    Chr$(10) & Chr$(10)
  frmProgInfo.Show
End Sub

Sub MnuWindow_Click ()
```

```
End Sub

Sub mnuWindowCascade_Click ()
   ' The index 0 represents the Cascade arrangement
   mdifrmBase.Arrange 0
End Sub

Sub mnuWindowTileH_Click ()
   '  The index 1 represents the Tile Horizontal arrangement
   mdifrmBase.Arrange 1
End Sub

Sub mnuWindowTileV_Click ()
   '  The index 2 represents the Tile Horizontal arrangement
   mdifrmBase.Arrange 2
End Sub

Sub Picture1_Click ()

End Sub
```

Code 2.2 PERF1.FRM (frmDataDisplay1)



Enter channel to display 0

Interval (ms) 1000

Declarations:

```
Const BoardNum% = 0            ' Board number
```

Sub Form_Load ()

```
  ' define the overwrite prompt
  ' declare revision level of Universal Library
  Const OFN_OVERWRITEPROMPT = &H2&
  ULStat% = cbDeclareRevision(CURRENTREVNUM)

  ' Initiate error handling
  ' activating error handling will trap errors like
  ' bad channel numbers and non-configured conditions.
  ' Parameters:
  '  PRINTALL      :all warnings and errors encountered will be printed
  '  DONTSTOP     :if an error is encountered, the program will not stop,
  '                errors must be handled locally

   ' Change working directory to the directory
   ' where the application was executed.
   ChDir app.Path
   ChDrive app.Path

  ULStat% = cbErrHandling%(PRINTALL, DONTSTOP)
  If ULStat% <> 0 Then Stop

  '  If cbErrHandling% is set for STOPALL or STOPFATAL during the program
  '  design stage, Visual Basic will be unloaded when an error is encountered.
  ' We suggest trapping errors locally until the program is ready for compiling
  ' to avoid losing unsaved data during program design.  This can be done by
  ' setting cbErrHandling options as above and checking the value of ULStat%
  ' after a call to the library. If it is not equal to 0, an error has occurred.

  ' Center the form on the screen when it loads
  Top = Screen.Height / 2 - Height / 2
  Left = Screen.Width / 2 - Width / 2
  ' Enter the caption into the label in the form immediately it loads
  lblFoot1.Caption = "PhotoVoltaics Research Laboratory" & Chr$(10) & "Department of_
   Physics" & Chr$(10) & "Moi University"
End Sub

Sub cmd3DCurrent_Click ()
   ' Display channel number 1
   txtNumchan(i%).Text = 1
End Sub

Sub cmd3DExit_Click ()
   ' Show the Confirm Quit dialog
   frmConfPerf1.Show
End Sub

Sub cmd3DIrrad_Click ()
    ' Display channel number 2
   txtNumchan(i%).Text = 2
End Sub

Sub cmd3DRestart_Click ()
   cmdStopConvert.Visible = 1  ' Show the Stop command button
   cmd3DStop.Enabled = True ' Enable the Stop icon on the toolbar
   cmdStopConvert.Default = 1  ' Set the Stop command button as the default
   cmdRestart.Visible = 0 'Hide the ReStart command button
   cmd3DRestart.Enabled = False  ' Disable the Restart icon on the toolbar
   cmdReset.Visible = 0   ' Hide the Reset command button
   mnuFileStop.Enabled = True  ' Enable the Stop option in the File menu
   mnuFileReset.Enabled = False  ' Disable the Reset option in the File menu

   ' Show the parameter and value boxes
   lblShowParameter.Visible = 1
   lblParameter.Visible = 1

   tmrConvert.Enabled = -1   ' Enable the timer
   tmrConvert.Interval = Val(txtInterval.Text)  'Set the timer interval at the user-specified value
End Sub
```

```
Sub cmd3DStart_Click ()
    '  CancelError is True
      On Error GoTo errhandling
      ' Set filters
      CMDialog1.Filter = "All Files (*.*)|*.*|Data Files (*.dat)|*.dat|Text Files (*.txt)|*.txt"
      ' Specify default filter
      CMDialog1.FilterIndex = 2
      ' display the File Open dialog
      CMDialog1.Action = 2
      CMDialog1.Flags = OFN_OVERWRITEPROMPT
      CMDialog1.DialogTitle = "Save Data As"
      Filename = CMDialog1.Filename

   ' Open the file and write the column names and then close the file
   Fnum = 2
   Open Filename For Output As Fnum
   Write #Fnum, "Date",  "Time", "Channel", "Value"
   Close

  cmdStartConvert.Visible = 0 ' Hide the Start command button
  cmd3DStart.Enabled = False  ' Disable the Start icon on the toolbar
  cmdStopConvert.Visible = 1  ' Show the Stop command button
  cmd3DStop.Enabled = True    ' Enable the Stop icon on the toolbar
  cmdStopConvert.Default = 1  ' Set the Stop button as default
  cmdReset.Visible = 0  ' Hide the Reset command button
  cmdRestart.Visible = 0 ' Hide the Restart command button
  mnuFileStart.Enabled = False ' Disable the Start option in the File menu
  mnuFileStop.Enabled = True ' Enable the Stop option in the File menu
  mnuFileReset.Enabled = False ' Disable the Reset option in the File menu
  mnuFileRestart.Enabled = False ' Disable the ReStart option in the File menu

  tmrConvert.Enabled = -1 ' Enable the timer
  tmrConvert.Interval = Val(txtInterval.Text) ' Set the timer interval at user-specified value
  lblParameter.Visible = 1 ' Show the parameter caption
  lblShowParameter.Visible = 1 ' Show the value
errhandling:
'  user pressed cancel button
   Exit Sub
End Sub

Sub cmd3DStop_Click ()
  tmrConvert.Enabled = 0 ' Disable the timer
  cmdRestart.Visible = 1 ' Show the Restart command button
  cmd3DRestart.Enabled = True ' Enable the Restart icon on the toolbar
  cmdReset.Visible = 1 ' Show the Reset command button
  mnuFileRestart.Enabled = True ' Enable the Restart option in the File menu
  mnuFileStop.Enabled = False ' Disable the Stop option in the File menu
  mnuFileReset.Enabled = True ' Enable the Reset option in the File menu
End Sub
```

```
Sub cmd3DVoltage_Click ()
  ' Display channel number 0
  txtNumchan(i%).Text = 0
End Sub

Sub cmdExit_Click ()
  ' Show the Confirm Quit dialog
  frmConfPerf1.Show
End Sub

Sub cmdReset_Click ()
  ' Reset all values to zero
  txtNumchan(i%).Text = 0
  lblShowData(i%).Caption = 0
  ' Hide the parameter and value boxes
  lblShowParameter.Visible = 0
  lblParameter.Visible = 0
  cmdStartConvert.Visible = 0 ' Hide the Start command button
  cmdRestart.Visible = 1 ' Show the Restart command button
  mnuFileStart.Enabled = False 'Disable the Start option in the File menu
  mnuFileRestart.Enabled = True ' Enable the Restart option in the File menu
End Sub

Sub cmdRestart_Click ()
  cmdStopConvert.Visible = 1 'Show the Stop command button
  cmd3DStop.Enabled = True 'Enable the Stop icon on the toolbar
  cmdStopConvert.Default = 1 'Set the Stop command button as the default
  cmdRestart.Visible = 0 ' Hide the ReStart command button
  cmd3DRestart.Enabled = False 'Disable the Restart icon on the toolbar
  cmdReset.Visible = 0   'Hide the Reset command button

  mnuFileStop.Enabled = True 'Enable the Stop option in the File menu
  mnuFileReset.Enabled = False ' Disable the Reset option in the File menu

  'Show the parameter and value boxes
  lblShowParameter.Visible = 1
  lblParameter.Visible = 1
  tmrConvert.Enabled = -1 'Enable the timer
  tmrConvert.Interval = Val(txtInterval.Text) 'Set the timer interval at the user-specified value
End Sub

Sub cmdStartConvert_Click ()
  ' CancelError is True
    On Error GoTo errH
    ' Set filters
    CMDialog1.Filter = "All Files (*.*)|*.*|Data Files (*.dat)|*.dat|Text Files (*.txt)|*.txt"
    ' Specify default filter
    CMDialog1.FilterIndex = 2
    ' display the File Open dialog
    CMDialog1.Action = 2
```

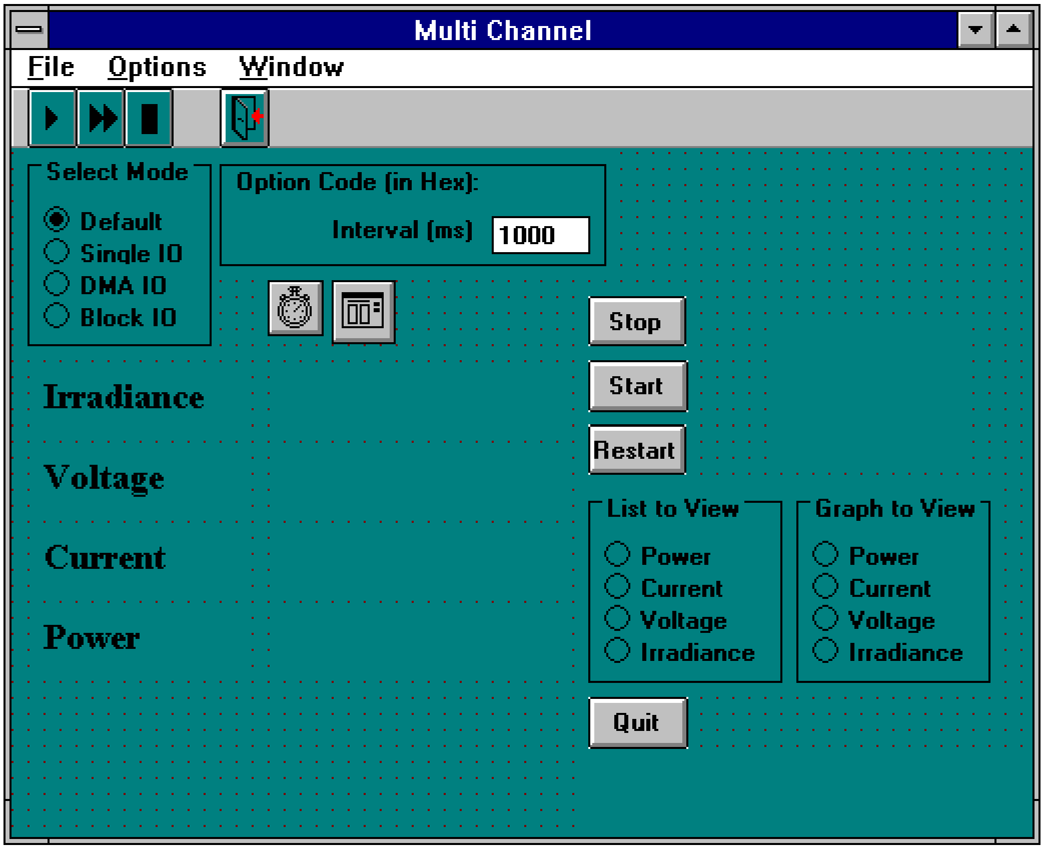
```
        CMDialog1.Flags = OFN_OVERWRITEPROMPT
        CMDialog1.DialogTitle = "Save Data As"
        Filename = CMDialog1.Filename

    '  Open the file and write the column names and then close the file
    Fnum = 2
    Open Filename For Output As Fnum
    Write #Fnum, "Date", "Time", "Channel", "Value"
    Close

   cmdStartConvert.Visible = 0  '  Hide the Start command button
   cmd3DStart.Enabled = False   ' Disable the Start icon on the toolbar
   cmdStopConvert.Visible = 1   ' Show the Stop command button
   cmd3DStop.Enabled = True     ' Enable the Stop icon on the toolbar
   cmdStopConvert.Default = 1   ' Set the Stop button as default
   cmdReset.Visible = 0 ' Hide the Reset command button
   cmdRestart.Visible = 0 ' Hide the Restart command button

   mnuFileStart.Enabled = False '  Disable the Start option in the File menu
   mnuFileStop.Enabled = True ' Enable the Stop option in the File menu
   mnuFileReset.Enabled = False ' Disable the Reset option in the File menu
   mnuFileRestart.Enabled = False ' Disable the ReStart option in the File menu

   tmrConvert.Enabled = -1 ' Enable the timer
   tmrConvert.Interval = Val(txtInterval.Text)  ' Set the timer interval at user-specified value
   lblParameter.Visible = 1  ' Show the parameter caption
   lblShowParameter.Visible = 1  ' Show the value

errH:
'  user pressed cancel button
    Exit Sub
End Sub

Sub cmdStopConvert_Click ()
   tmrConvert.Enabled = 0  ' Disable the timer
   cmdRestart.Visible = 1  ' Show the Restart command button
   cmd3DRestart.Enabled = True  ' Enable the Restart icon on the toolbar
   cmdReset.Visible = 1  ' Show the Reset command button

   mnuFileRestart.Enabled = True  ' Enable the Restart option in the File menu
   mnuFileStop.Enabled = False  ' Disable the Stop option in the File menu
   mnuFileReset.Enabled = True  ' Enable the Reset option in the File menu
End Sub

Sub lblChanPrompt_Click ()

End Sub

Sub lblfoot1_Click ()
```

```
End Sub

Sub lblInterval_Click ()

End Sub

Sub lblParameter_Click ()

End Sub

Sub lblShowData_Click (Index As Integer)

End Sub

Sub lblShowParameter_Click ()

End Sub

Sub lblShowVolts_Click (Index As Integer)

End Sub

Sub lblValueRead_Click ()

End Sub

Sub lblVoltsRead_Click ()

End Sub

Sub mnuChanCurrent_Click ()
   ' Display channel number 1
   txtNumchan(i%).Text = 1
End Sub

Sub mnuChanIrrad_Click ()
    ' Display channel number 2
   txtNumchan(i%).Text = 2
End Sub

Sub mnuChannel_Click ()

End Sub

Sub mnuChanVoltage_Click ()
   ' Display channel number 0
   txtNumchan(i%).Text = 0
End Sub

Sub mnuFile_Click ()
```

```
End Sub

Sub mnuFileExit_Click ()
   ' Show the Confirm Quit dialog
   frmConfPerf1.Show
End Sub

Sub mnuFileReset_Click ()
   ' Reset all values to zero
   txtNumchan(i%).Text = 0
   lblShowData(i%).Caption = 0
   ' Hide the parameter and value boxes
   lblShowParameter.Visible = 0
   lblParameter.Visible = 0
   cmdStartConvert.Visible = 0 ' Hide the Start command button
   cmdRestart.Visible = 1 ' Show the Restart command button
   mnuFileStart.Enabled = False ' Disable the Start option in the File menu
   mnuFileRestart.Enabled = True  ' Enable the Restart option in the File menu
End Sub

Sub mnuFileRestart_Click ()
   cmdStopConvert.Visible = 1 ' Show the Stop command button
   cmd3DStop.Enabled = True ' Enable the Stop icon on the toolbar
   cmdStopConvert.Default = 1 ' Set the Stop command button as the default
   cmdRestart.Visible = 0 ' Hide the ReStart command button
   cmd3DRestart.Enabled = False ' Disable the Restart icon on the toolbar
   cmdReset.Visible = 0   ' Hide the Reset command button

   mnuFileStop.Enabled = True ' Enable the Stop option in the File menu
   mnuFileReset.Enabled = False ' Disable the Reset option in the File menu

   ' Show the parameter and value boxes
   lblShowParameter.Visible = 1
   lblParameter.Visible = 1
   tmrConvert.Enabled = -1  ' Enable the timer
   tmrConvert.Interval = Val(txtInterval.Text) ' Set the timer interval at the user-specified value
End Sub

Sub mnuFileStart_Click ()
   ' CancelError is True
      On Error GoTo errhandle
      ' Set filters
      CMDialog1.Filter = "All Files (*.*)|*.*|Data Files (*.dat)|*.dat|Text Files (*.txt)|*.txt"
      ' Specify default filter
      CMDialog1.FilterIndex = 2
      ' display the File Open dialog
      CMDialog1.Action = 2
      CMDialog1.Flags = OFN_OVERWRITEPROMPT
      CMDialog1.DialogTitle = "Save Data As"
      Filename = CMDialog1.Filename

    ' Open the file and write the column names and then close the file
    Fnum = 2
    Open Filename For Output As Fnum
    Write #Fnum, "Date",  "Time", "Channel", "Value"
    Close

   cmdStartConvert.Visible = 0 '  Hide the Start command button
   cmd3DStart.Enabled = False  ' Disable the Start icon on the toolbar
   cmdStopConvert.Visible = 1  ' Show the Stop command button
   cmd3DStop.Enabled = True    ' Enable the Stop icon on the toolbar
   cmdStopConvert.Default = 1  ' Set the Stop button as default
   cmdReset.Visible = 0 ' Hide the Reset command button
   cmdRestart.Visible = 0 ' Hide the Restart command button

   mnuFileStart.Enabled = False ' Disable the Start option in the File menu
   mnuFileStop.Enabled = True ' Enable the Stop option in the File menu
   mnuFileReset.Enabled = False ' Disable the Reset option in the File menu
   mnuFileRestart.Enabled = False ' Disable the ReStart option in the File menu

   tmrConvert.Enabled = -1 ' Enable the timer
   tmrConvert.Interval = Val(txtInterval.Text) ' Set the timer interval at user-specified value
   lblParameter.Visible = 1 ' Show the parameter caption
   lblShowParameter.Visible = 1 ' Show the value

errhandle:
'  user pressed cancel button
    Exit Sub
End Sub


Sub mnuFileStop_Click ()
   tmrConvert.Enabled = 0 '  Disable the timer
   cmdRestart.Visible = 1 ' Show the Restart command button
   cmd3DRestart.Enabled = True ' Enable the Restart icon on the toolbar
   cmdReset.Visible = 1 ' Show the Reset command button
   mnuFileRestart.Enabled = True ' Enable the Restart option in the File menu
   mnuFileStop.Enabled = False ' Disable the Stop option in the File menu
   mnuFileReset.Enabled = True ' Enable the Reset option in the File menu
End Sub

Sub mnusep_Click ()

End Sub

Sub mnuWindow_Click ()

End Sub
```

```
Sub mnuWindowCascade_Click ()
    ' Cascade all open windows
    mdifrmbase.Arrange 0
End Sub

Sub mnuWindowTileH_Click ()
    ' Tile all open windows horizontally
    mdifrmbase.Arrange 1
End Sub

Sub mnuWindowTileV_Click ()
    ' Tile all open windows vertically
    mdifrmbase.Arrange 2
End Sub

Sub Panel3D1_DragDrop (Source As Control, X As Single, Y As Single)

End Sub

Sub tmrConvert_Timer ()
    '  Collect the data with cbAIn%()
   '  Parameters:
   '    BoardNum%    :the number used by CB.CFG to describe this board
   '    Chan%          :the input channel number
   '    Gain%          :the gain for the board.
   '    DataValue%  :the name for the value collected

   Gain% = BIP5VOLTS ' set the gain
   chan% = Val(txtNumchan(i%).Text)   '  set input channel to that specified by the user
   tmrConvert.Interval = Val(txtInterval.Text)  ' Set the timer interval to the value specified by the
´ user
   mdifrmbase.cmdExit.Visible = False  ' the Exit icon button in the main form is hidden

   ULStat% = cbAIn%(BoardNum%, chan%, Gain%, DataValue%)  ' Read the specified channel
   If ULStat% = 30 Then MsgBox "Change the Gain% argument to one supported by this_
                          board.",   0, "Unsupported Gain"
   If ULStat% <> 0 Then Stop  ' Error message

   ' Convert the counts to volts using the cbToEngUnits function
   ULStat% = cbToEngUnits(BoardNum%, Gain%, DataValue%, EngUnits!)
   If ULStat% <> 0 Then Stop

   lblShowData(i%).Caption = Format$(DataValue%, "0")     '  print the counts
   lblShowVolts(i%).Caption = Format$(EngUnits!, "0.000 ") + " V"   ' print the voltage

   lblShowData(0).Visible = True  '  Show the value read in counts

   ' Allocate the channel name and value to be displayed depending on the user's selection
   If chan% = 0 Then lblParameter.Caption = "Voltage"
   If chan% = 0 Then lblShowParameter.Caption = Format$(Val(lblShowVolts(i%).Caption) *_
        8.5, "0.000") + " V"

  If chan% = 1 Then lblParameter.Caption = "Current"
  If chan% = 1 Then lblShowParameter.Caption = Format$(Val(lblShowVolts(i%).Captio n) *_
        14.5, "0.000") + " A"

  If chan% = 2 Then lblParameter.Caption = "Irradiance"
  If chan% = 2 Then lblShowParameter.Caption = Format$(Val(lblShowVolts(i%).Caption) *_
        10000, "0.0") + " W/m2 "

  '  Generates an error message if the user enters an invalid channel number
  Do While chan% > 2
  Close Filename ' Closes the data file
  Beep
  tmrConvert.Enabled = 0 ' Stops the timer
  CaptionText1 = "           Invalid Channel Number!"
  CaptionText2 = "    Please enter  a number between 0 and 2."
  frmError.lblError.Caption = CaptionText1 & Chr(10) & Chr(10) & CaptionText2
  frmError.Show
  Exit Do
  Loop

  ' Display the channel name and value
  Channel = lblParameter.Caption
  value = lblShowParameter.Caption

   ' Write the data to a file specified by the user.
   ' The Common Dialog control is used here.
   Fnum = 2

   Filename = CMDialog1.Filename
   Open Filename For Append As Fnum
   Write #Fnum, Date$, Time$, Channel, value
   Close
End Sub

Sub txtInterval_Change ()

End Sub

Sub txtNumChan_Change (Index As Integer)

End Sub
```

Code 2.3 PERF2.FRM (frmDataDisplay)



Select Mode
● Default
○ Single IO
○ DMA IO
○ Block IO

Option Code (in Hex):

Interval (ms) 1000

List to View
○ Power
○ Current
○ Voltage
○ Irradiance

Graph to View
○ Power
○ Current
○ Voltage
○ Irradiance

Declarations:

```
Const BoardNum = 0          ' Board number
Const NumPoints& = 2048      ' Number of data points to collect
Const FirstPoint& = 0       ' set first element in buffer to transfer to array

Dim ADData%(NumPoints&)  ' dimension an array to hold the input values
Dim ShowVolts%(NumPoints&)
Dim MemHandle%              ' define a variable to contain the handle for
                    ' memory allocated by Windows through cbWinBufAlloc%()
Dim Options%              ' define cbAInScan() Options as Module level variable
```

```
Sub Form_Load ()
' Define the overwrite prompt
Const OFN_OVERWRITEPROMPT = &H2&

  '  declare revision level of Universal Library
  ULStat% = cbDeclareRevision(CURRENTREVNUM)

  ' Initiate error handling
  '  activating error handling will trap errors like
  '  bad channel numbers and non-configured conditions.
  ' Parameters:
  '   PRINTALL    :all warnings and errors encountered will be printed
  '   DONTSTOP    :if an error is encountered, the program will not stop,
  '               errors must be handled locally
  ULStat% = cbErrHandling%(PRINTALL, DONTSTOP)
  If ULStat% <> 0 Then Stop

  MemHandle% = cbWinBufAlloc%(NumPoints&)  ' set aside memory to hold data
  If MemHandle% = 0 Then Stop

  Options% = CONVERTDATA
   ' Change working directory to the directory
   '  where the application was executed.
   ChDir app.Path
   ChDrive app.Path

   ' Set the option buttons for viewing data lists and graphs at false
   OptPowerList.Value = False
   OptCurrentList.Value = False
   OptVoltList.Value = False
   OptIrradList.Value = False
   OptPowerGraph.Value = False
   OptCurrentGraph.Value = False
   OptVoltGraph.Value = False
   OptIrradGraph.Value = False

   ' Set the foot caption of the form
   lblFoot.Caption = "PhotoVoltaics Research Laboratory" & Chr$(10) & "Department of_
            Physics" & Chr$(1 0) & "Moi University"
End Sub

Sub Form_Click ()
    ' Set the option buttons for viewing data lists and graphs at false
    OptPowerList.Value = False
    OptCurrentList.Value = False
    OptVoltList.Value = False
    OptIrradList.Value = False
    OptPowerGraph.Value = False
    OptCurrentGraph.Value = False
    OptVoltGraph.Value = False
    OptIrradGraph.Value = False
End Sub

Sub cmd3DExit_Click ()
  If ULStat% <> 0 Then Stop
  tmrConvert.Enabled = 0  ' Disable the timer
  frmConfPerf.Show  ' Show the Confirm Quit dialogue
End Sub

Sub cmd3DRestart_Click ()
```

```
   cmdStop.Visible = 1  ' Show the Stop command button
   cmdStop.Default = 1  ' Set the Stop command as default
   cmd3DStop.Enabled = True  ' Enable the Stop icon on the toolbar
   cmdRestart.Visible = 0  ' Hide the Restart command button
   cmd3DRestart.Enabled = False  ' Disable the Restart icon on the toolbar
   mnuFileStop.Enabled = True  ' Enable the Stop option in the File menu
   mnuFileRestart.Enabled = False  ' Disable the Restart option in the File menu

   tmrConvert.Enabled = -1 ' Enable the timer
   tmrConvert.Interval = Val(txtInterval.Text)  ' Set the timer interval at the value specified by the
  ′  user
End Sub

Sub cmd3DStart_Click ()
   '  CancelError is True
      On Error GoTo errhandling
      ' Set filters
      CMDialog1.Filter = "All Files (*.*)|*.*|Data Files (*.dat)|*.dat|Text Files (*.txt)|*.txt"
      ' Specify default filter
      CMDialog1.FilterIndex = 2
      ' display the File Open dialog
      CMDialog1.Action = 2
      ' Set the flags property to the overwrite prompt
      CMDialog1.Flags = OFN_OVERWRITEPROMPT
      ' Set the title for the Save As dialog
      CMDialog1.DialogTitle = "Save Data As"
      ' Set the file name to the name specified by the user
      Filename = CMDialog1.Filename

   ' Open the data file and write the column titles
   Fnum = 2
   Open Filename For Output As Fnum
   Write #Fnum, "Date", "Time", "Irradiance", "Voltage", "Current", " Power"
   Close

  cmdStart.Visible = 0   ' Hide the Start command button
  cmd3DStart.Enabled = False  ' Disable the Start icon on the toolbar
  mnuFileStart.Enabled = False  ' Disable the Start option in the File menu
  cmdStop.Visible = -1 ' Show the Stop command button
  cmd3DStop.Enabled = True  ' Enable the Stop icon on the toolbar
  mnuFileStop.Enabled = True  ' Enable the Stop option in the File menu
  tmrConvert.Enabled = -1 ' Enable the timer
  tmrConvert.Interval = Val(txtInterval.Text)  ' Set the timer interval to the value specified by the
  ‘  user

errhandling:
' user pressed cancel button
  Exit Sub
End Sub
```

```
Sub cmd3DStop_Click ()

    tmrConvert.Enabled = 0  ' Disable the timer
    cmdStop.Visible = 0 ' Hide the Stop command button
    cmd3DStop.Enabled = False ' Disable the Stop icon on the toolbar
    mnuFileStop.Enabled = False ' Disable the Stop option in the File menu
    cmdRestart.Visible = 1 ' Show the Restart command button
    cmd3DRestart.Enabled = True ' Enable the Restart icon on the toolbar
    mnuFileRestart.Enabled = True ' Enable the Restart option in the File menu
End Sub

Sub cmdRestart_Click ()
    cmdStop.Visible = 1 ' Show the Stop command button
    cmdStop.Default = 1 ' Set the Stop command as default
    cmd3DStop.Enabled = True  ' Enable the Stop icon on the toolbar
    cmdRestart.Visible = 0 ' Hide the Restart command button
    cmd3DRestart.Enabled = False ' Disable the Restart icon on the toolbar
    mnuFileStop.Enabled = True ' Enable the Stop option in the File menu
    mnuFileRestart.Enabled = False ' Disable the Restart option in the File menu

    tmrConvert.Enabled = -1 ' Enable the timer
    tmrConvert.Interval = Val(txtInterval.Text) ' Set the timer interval at the value specified by the
    ‘ user
End Sub

Sub cmdStart_Click ()
    ' CancelError is True
        On Error GoTo errhandle
        ' Set filters
        CMDialog1.Filter = "All Files (*.*)|*.*|Data Files (*.dat)|*.dat|Text Files (*.txt)|*.txt"
       ' Specify default filter
        CMDialog1.FilterIndex = 2
        ' display the File Open dialog
        CMDialog1.Action = 2
        ' Set the flags property to the overwrite prompt
        CMDialog1.Flags = OFN_OVERWRITEPROMPT
        ' Set the title for the Save As dialog
        CMDialog1.DialogTitle = "Save Data As"
        ' Set the file name to the name specified by the user
        Filename = CMDialog1.Filename

    ' Open the data file and write the column titles
    Fnum = 2
    Open Filename For Output As Fnum
    Write #Fnum, "Date", "Time", "Irradiance", "Voltage", "Current", "Power"
    Close

   cmdStart.Visible = 0  ' Hide the Start command button
   cmd3DStart.Enabled = False ' Disable the Start icon on the toolbar
   mnuFileStart.Enabled = False ' Disable the Start option in the File menu
   cmdStop.Visible = -1 ' Show the Stop command button
   cmd3DStop.Enabled = True  ' Enable the Stop icon on the toolbar
   mnuFileStop.Enabled = True  ' Enable the Stop option in the File menu
   tmrConvert.Enabled = -1 ' Enable the timer
   tmrConvert.Interval = Val(txtInterval.Text)  ' Set the timer interval to the value specified by the
   ‘  user

errhandle:
' user pressed cancel button
    Exit Sub
End Sub

Sub cmdStop_Click ()
    tmrConvert.Enabled = 0   ' Disable the timer
    cmdStop.Visible = 0  ' Hide the Stop command button
    cmd3DStop.Enabled = False  ' Disable the Stop icon on the toolbar
    mnuFileStop.Enabled = False  ' Disable the Stop option in the File menu
    cmdRestart.Visible = 1   ' Show the Restart command button
    cmd3DRestart.Enabled = True  ' Enable the Restart icon on the toolbar
    mnuFileRestart.Enabled = True  ' Enable the Restart option in the File menu
End Sub

Sub cmdStopConvert_Click ()
   If ULStat% <> 0 Then Stop
   tmrConvert.Enabled = 0  ' Disable the timer
   frmConfPerf.Show  ' Show the Confirm Quit dialogue
End Sub

Sub framGraph_DragDrop (Source As Control, X As Single, Y As Single)

End Sub

Sub framList_DragDrop (Source As Control, X As Single, Y As Single)

End Sub

Sub fraMode_DragDrop (Source As Control, X As Single, Y As Single)

End Sub

Sub fraOptInt_DragDrop (Source As Control, X As Single, Y As Single)

End Sub

Sub Label2_Click ()

End Sub

Sub lblADData_Click (Index As Integer)
```

```
End Sub

Sub lblChan3_Click ()

End Sub

Sub lblChan4_Click ()

End Sub

Sub lblCurrent_Click ()

End Sub

Sub lblfoot_Click ()

End Sub

Sub lblInterval_Click ()

End Sub

Sub lblirrad_Click ()

End Sub

Sub lblModeNo_Click ()

End Sub

Sub lblOptCode_Click ()

End Sub

Sub lblShowCurrent_Click ()

End Sub

Sub lblShowIrrad_Click ()

End Sub

Sub lblShowPower_Click ()

End Sub

Sub lblShowVoltage_Click ()

End Sub
```

```
Sub lblShowVolts_Click (Index As Integer)

End Sub

Sub lblVoltage_Click ()

End Sub

Sub mnuFile_Click ()

End Sub

Sub mnuFileQuit_Click ()
    If ULStat% <> 0 Then Stop
   tmrConvert.Enabled = 0  ' Disable the timer
   frmConfPerf.Show  ' Show the Confirm Quit dialogue
End Sub

Sub mnuFileRestart_Click ()
    cmdStop.Visible = 1  ' Show the Stop command button
    cmdStop.Default = 1  ' Set the Stop command as default
    cmd3DStop.Enabled = True  ' Enable the Stop icon on the toolbar
    cmdRestart.Visible = 0  ' Hide the Restart command button
    cmd3DRestart.Enabled = False  ' Disable the Restart icon on the toolbar
    mnuFileStop.Enabled = True  ' Enable the Stop option in the File menu
    mnuFileRestart.Enabled = False  ' Disable the Restart option in the File menu
    tmrConvert.Enabled = -1 ' Enable the timer
    tmrConvert.Interval = Val(txtInterval.Text)  ' Set the timer interval at the value specified by the
   ‘  user
End Sub

Sub mnuFileStart_Click ()
    '  CancelError is True
        On Error GoTo errhand
        '  Set filters
        CMDialog1.Filter = "All Files (*.*)|*.*|Data Files (*.dat)|*.dat|Text Files (*.txt)|*.txt"
       ' Specify default filter
        CMDialog1.FilterIndex = 2
        ' display the File Open dialog
        CMDialog1.Action = 2
        ' Set the flags property to the overwrite prompt
        CMDialog1.Flags = OFN_OVERWRITEPROMPT
        ' Set the title for the Save As dialog
        CMDialog1.DialogTitle = "Save Data As"
        ' Set the file name to the name specified by the user
        Filename = CMDialog1.Filename

    ' Open the data file and write the column titles
    Fnum = 2
    Open Filename For Output As Fnum
    Write #Fnum, "Date", "Time", "Irradiance", "Voltage", "Current", "Power"
    Close

   cmdStart.Visible = 0  ' Hide the Start command button
   cmd3DStart.Enabled = False ' Disable the Start icon on the toolbar
   mnuFileStart.Enabled = False '  Disable the Start option in the File menu
   cmdStop.Visible = -1 ' Show the Stop command button
   cmd3DStop.Enabled = True ' Enable the Stop icon on the toolbar
   mnuFileStop.Enabled = True ' Enable the Stop option in the File menu
   tmrConvert.Enabled = -1 ' Enable the timer
   tmrConvert.Interval = Val(txtInterval.Text) ' Set the timer interval to the value specified by the
   ‘  user

errhand:
' user pressed cancel button
   Exit Sub
End Sub

Sub mnuFileStop_Click ()
    tmrConvert.Enabled = 0  ' Disable the timer
    cmdStop.Visible = 0 ' Hide the Stop command button
    cmd3DStop.Enabled = False ' Disable the Stop icon on the toolbar
    mnuFileStop.Enabled = False  '  Disable the Stop option in the File menu
    cmdRestart.Visible = 1 ' Show the Restart command button
    cmd3DRestart.Enabled = True  ' Enable the Restart icon on the toolbar
    mnuFileRestart.Enabled = True  ' Enable the Restart option in the File menu
End Sub

Sub mnuOptBlockIO_Click ()
  '  A/D conversions are initiated by a trigger.
  '  Transfers are handled by REP/INSW
  Options% = CONVERTDATA + BLOCKIO
  '  Select the option button for BLOCKIO
  OptModeBlock.Value = True
End Sub

Sub mnuOptDefault_Click ()
   ' The data will automatically be converted to 12-bit A/D values
  Options% = CONVERTDATA
  '  Select the option button for Default
  OptDefault.Value = True
End Sub

Sub mnuOptDMAIO_Click ()
   '  A/D conversions are initiated by a trigger.
   '  Transfers are initiated by a DMA request
   Options% = CONVERTDATA + DMAIO
    ' Select the option button for DMA
   OptModeDMA.Value = True
```

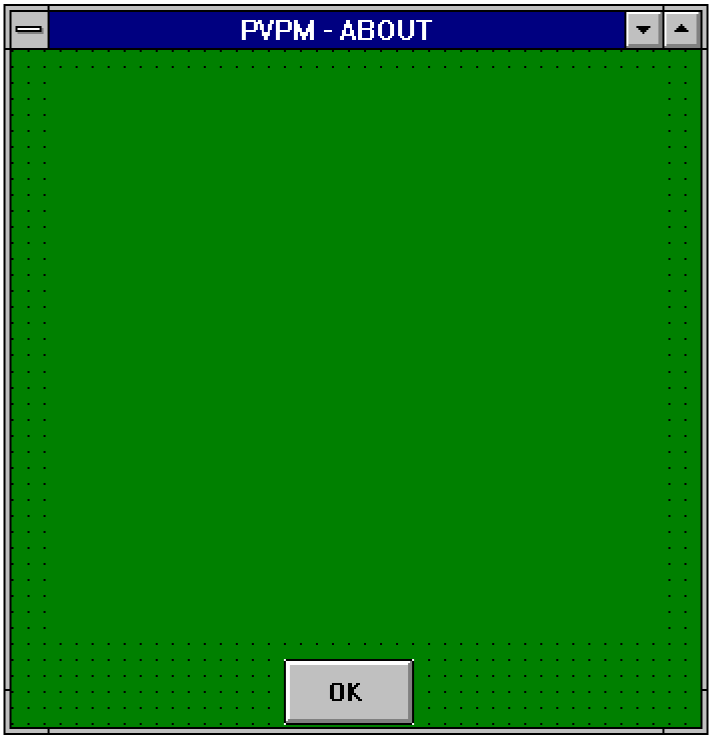
```
End Sub

Sub mnuOptions_Click ()

End Sub

Sub mnuOptSingleIO_Click ()
  '  A/D conversions and transfers to
  '  memory are initiated by an interrupt.
  Options% = CONVERTDATA + SINGLEIO
  '  Select the option button for SINGLEIO
  OptModeSing.Value = True
End Sub

Sub mnuWindow_Click ()

End Sub

Sub mnuWindowCascade_Click ()
   ' Cascade all open windows
   mdifrmbase.Arrange 0
End Sub

Sub mnuWindowTileH_Click ()
   '  Tile all open windows horizontally
   mdifrmbase.Arrange 1
End Sub

Sub mnuWindowTileV_Click ()
   '  Tile all open windows vertically
   mdifrmbase.Arrange 2
End Sub

Sub OptCurrentGraph_Click ()
   '  Show the graph for Current
   frmGraphC.Show
End Sub

Sub OptCurrentList_Click ()
   '  Show the data list for Current
   frmDataListC.Show
End Sub

Sub optDefault_Click ()
  ' The data will automatically be converted to 12-bit A/D values
  Options% = CONVERTDATA
End Sub

Sub OptIrradGraph_Click ()
   ' Show the graph for Irradiance
   frmGraphI.Show
End Sub

Sub OptIrradList_Click ()
   '  Show the data list for Irradiance
   frmDataListI.Show
End Sub

Sub optModeBlock_Click ()
  ' A/D conversions are initiated by a trigger.
  ' Transfers are handled by REP/INSW
  Options% = CONVERTDATA + BLOCKIO
End Sub

Sub optModeDMA_Click ()
  ' A/D conversions are initiated by a trigger.
  ' Transfers are initiated by a DMA request
  Options% = CONVERTDATA + DMAIO
End Sub

Sub optModeSing_Click ()
  ' A/D conversions and transfers to
  ' memory are initiated by an interrupt.
  Options% = CONVERTDATA + SINGLEIO
End Sub

Sub OptPowerGraph_Click ()
   '  Show the graph for Power
   frmGraphP.Show
End Sub

Sub OptPowerList_Click ()
    ' Show the data list for Power
    frmDataListp.Show
End Sub

Sub OptVoltGraph_Click ()
    '  Show the graph for Voltage
   frmGraphV.Show
End Sub

Sub OptVoltList_Click ()
   ' Show the data list for Voltage
   frmDataListV.Show
End Sub

Sub Panel3D1_DragDrop (Source AsControl, X As Single, Y As Single)

End Sub
```

```
Sub tmrConvert_Timer ()
  ' Display the data conversion mode
  lblModeNo.Caption = Hex$(Options%)
  mdifrmbase.cmdExit.Visible = False  ' the Exit icon button in the main form is hidden

  ' Collect the values with cbAInScan%()
  '  Parameters:
  '    BoardNum    :the number used by CB.CFG to describe this board
  '    LowChan%    :the first channel of the scan
  '    HighChan%   :the last channel of the scan
  '    CBCount&    :the total number of A/D samples to collect
  '    CBRate&     :sample rate
  '    Gain%       :the gain for the board
  '    ADData%     :the array for the collected data values
  '    Options%    :data collection options (determined by
  '                 Set Mode option buttons in this program)

  LowChan% = 0                ' first channel to acquire
  HighChan% = 2               '  last channel to acquire
  CBCount& = NumPoints&           ' total number of data points to collect
  CBRate& = 290               ' sampling rate (in Hz per channel)
  Gain% = BIP5VOLTS               ' set the gain

  If MemHandle% = 0 Then Stop      ' check that a handle to a memory buffer exists

  ULStat% = cbAInScan%(BoardNum%, LowChan%, HighChan%, CBCount&, CBRa te&,_
              Gain%, MemHandle%, Options%)

  ' Set the timer interval to the value specified by the user
  tmrConvert.Interval = Val(txtInterval.Text)

  Select Case ULStat%
    Case 0
    Case 84
      MsgBox "The CONVERT opt ion cannot be used with 16 bit convertors. Set Options% to_
          NOCONVERTDATA."
      Stop   ' Change Options% above to NOCONVERTDATA (Options% = 0)
    Case 91
      ULStat% = cbErrHandling%(DONTPRINT, DONTSTOP)
      '  Turn off library error handling for subsequent calls
    Case Else
      Stop
  End Select

' Transfer the data from the memory buffer set up by Windows to an array for use by Visual
‘ Basic
  ULStat% = cbWinBufToArray%(MemHandle%, ADData%(0), FirstPoint&, CBCo unt&)
  If ULStat% <> 0 Then Stop

  For I% = 0 To 2
```

```
    ' Display data for channels 0 to 2.  However these values are not visible at run time
    lblADData(I%).Caption = Format$(ADData%(I%), "0")
    ' Calculate the value from each channel in volts. These values are also not visible at run time
    lblShowVolts(I%).Caption = Format$(((10 / 4096) * Val(lblADData(I%).Caption))  - 5,_
          "0.000") + "V"
  Next I%

  ' Make the final output captions visible at run time
  lblShowIrrad.Visible = True
  lblShowVoltage.Visible = True
  lblShowCurrent.Visible = True
  lblShowPower.Visible = True

  '  Apply the conversion factors and display the final output values with the corresponding units
  lblShowIrrad.Caption = Format$(Val(lblShowVolts(2)) * 10000, "0.0") + " W/m2 "
  lblShowVoltage.Caption = Format$(Val(lblShowVolts(0)) * 8.5, "0.000") + " V "
  lblShowCurrent.Caption = Format$(Val(lblShowVolts(1)) * 14.5, "0.000") + " A "
  lblShowPower.Caption = Format$(Val(lblShowVoltage.Caption) *_
            Val(lblShowCurrent.Caption), "0.000") + " W "

  ' Define variables for data storage
  Irrad = Val(lblShowIrrad.Caption)
  Voltage = Val(lblShowVoltage.Caption)
  Current = Val(lblShowCurrent.Caption)
  Power = Val(lblShowPower.Caption)

  Fnum = 2 ' File number for data storage

  '  Open file for data storage and store the data
  Filename = CMDialog1.Filename
  Open Filename For Append As Fnum
  NL = Chr(10)
  Write #Fnum, Date$, Time$, Irrad, Voltage, Current, Power
  Close

  '  Allocate values to display in the graphs
  frmGraphI.GraphI.GraphData = Val(lblShowIrrad.Caption)
  frmGraphV.GraphV.GraphData = Val(lblShowVoltage.Caption)
  frmGraphC.GraphC.GraphData = Val(lblShowCurrent.Caption)
  frmGraphP.GraphP.GraphData = Val(lblShowPower.Caption)

  '  Allocate values to display in lists
  frmDataListI.GridI.Col = 0
  frmDataListI.GridI.Row = 0
  frmDataListI.GridI.Text = "Irradiance"
  frmDataListI.GridI.Col = 0
  frmDataListI.GridI.Row = 0
  frmDataListI.GridI.AddItem Format$(Val(lblShowVolts(2)) * 10000, "0.0")
  frmDataListV.GridV.Col = 0
  frmDataListV.GridV.Row = 0
```

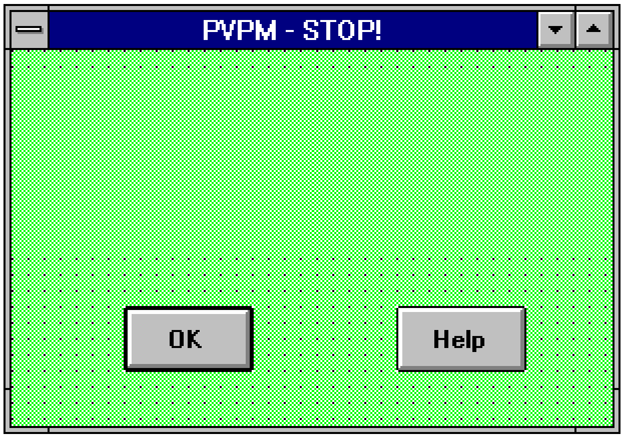
```
   frmDataListV.GridV.Text = "Voltage"
   frmDataListV.GridV.Col = 0
   frmDataListV.GridV.Row = 0
   frmDataListV.GridV.AddItem Format$(Val(lblShowVolts(0)) * 8.5, "0.000")
   frmDataListC.GridC.Col = 0
   frmDataListC.GridC.Row = 0
   frmDataListC.GridC.Text = "Current"
   frmDataListC.GridC.Col = 0
   frmDataListC.GridC.Row = 0
   frmDataListC.GridC.AddItem Format$(Val(lblShowVolts(1)) * 14.5, "0.000")
   frmDataListp.GridP.Col = 0
   frmDataListp.GridP.Row = 0
   frmDataListp.GridP.Text = "Power"
   frmDataListp.GridP.Col = 0
   frmDataListp.GridP.Row = 0
   frmDataListp.GridP.AddItem Format$(Val(lblShowVoltage.Caption) *_
           Val(lblShowCurrent.Caption), "0.000")

   ' Error handling
   On Error GoTo TopRowError
   frmDataListI.GridI.TopRow = frmDataListI.GridI.TopRow + 1
   frmDataListV.GridV.TopRow = frmDataListV.GridV.TopRow + 1
   frmDataListC.GridC.TopRow = frmDataListC.GridC.TopRow + 1
   frmDataListp.GridP.TopRow = frmDataListp.GridP.TopRow + 1
   On Error GoTo 0
   Exit Sub

TopRowError:
   frmDataListI.GridI.TopRow = 1
   frmDataListV.GridV.TopRow = 1
   frmDataListC.GridC.TopRow = 1
   frmDataListp.GridP.TopRow = 1
Resume Next
End Sub

Sub txtInterval_Change ()

End Sub
```

Code 2.4 ABPERF.FRM (frmperAbout)



```
Sub lblAbout_Click ()

End Sub

Sub Form_Load ()

End Sub

Sub cmdAboutOK_Click ()
    Unload frmProgInfo
    Unload frmPerfAbout
    mdifrmBase.Show
End Sub
```

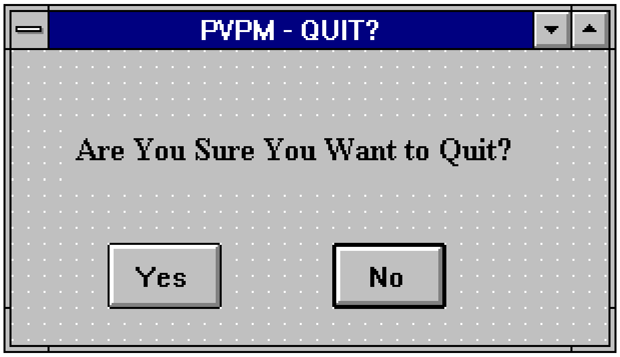
Code 2.5 CHANERR.FRM (frmError)



```
Sub Form_Load ()

End Sub

Sub lblError_Click ()

End Sub

Sub cmdErr_Click (Index As Integer)

Select Case Index

   Case 0

   ' Reset all the data to zero

   frmError.Hide
   frmDataDisplay1.txtNumChan(i%).Text = 0
   frmDataDisplay1.lblShowData(i%).Caption = 0
   frmDataDisplay1.lblShowParameter.Visible = 0
   frmDataDisplay1.lblParameter.Visible  = 0
   frmDataDisplay1.cmdStartConvert.Visible = 0
   frmDataDisplay1.cmdRestart.Visible = 1
   frmDataDisplay1.mnuFileStart.Enabled = False
   ' the Restart option is activated.
   frmDataDisplay1.mnuFileRestart.Enabled = True
   frmDataDisplay1.Show

   Case 1

   ' if the user clicks Help, the text as outlined below is set to appear in the blank label
   ‘ lblDD1Help.Caption in the form frmDD1Help.

   CaptionText1 = "You have entered a channel number that is not valid for this appl ication.  In_
```

```
    total, eight channels are available, but for the present application, only three are in use.  The_
    channels in use are:"
    CaptionText3 = "    Channel 0   Voltage"
    CaptionText4 = "    Channel 1   Current"
    CaptionText5 = "    Channel 2   Irradiance"
    CaptionText6 = "Specifying any channels other than these generates an 'invalid channel' _
    error, as long as the channel is within the range supported by the data access board. _
    Specifying a channel number outside the r ange supported by the board generates a_
    'ComputerBoards Error' and causes the program to abort.  Note that if you do not specify a_
    channel, i.e. if you leave the channel box blank, Channel 0 is assumed."
    CaptionText7 = "If you have a problem  remembering what each channel represents, it will be_
    helpful to use the options given in the 'Options' menu item to select the parameter that you_
     wish to monitor."
    frmDD1Help.lblDD1Help.Caption = CaptionText1 & Chr(10) & Chr(10) & CaptionTe xt3 &_
     Chr(10) & CaptionText4 & Chr(10) & CaptionText5 & Chr(10) & Chr(10) & CaptionText6_
      & Chr(10) & CaptionText7
    '  The Help dialogue is then shown with the above text on it.
    frmDD1Help.Show

End Select
End Sub
```

Code 2.6 CONFIRM1.FRM (frmConfPerf1)

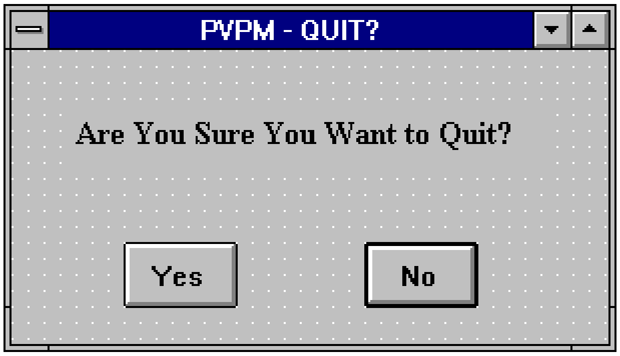


Are You Sure You Want to Quit?

```
Sub Form_Load ()

End Sub

Sub Label1_Click ()

End Sub

Sub cmdQuitYes_Click ()
   ' This dialogue appears prompting the user to confirm the quit command.
   ' If the user clicks Yes, the following code is executed.

   Close Filename  ' any open data files are closed

   mdifrmbase.cmdExit.Visible = True  ' the Exit icon button in the main form appears
   frmDataDisplay1.tmrConvert.Enabled = 0

   ' Reset all the values in the form frmDataDisplay1.
   frmDataDisplay1.lblShowData(0).Visible = False
   frmDataDisplay1.lblParameter.Visible = False
   frmDataDisplay1.lblShowParameter.Visible = False
   frmDataDisplay1.txtNumChan(0).Text = 0
   frmDataDisplay1.txtInterval.Text = 1000
   frmDataDisplay1.cmdStartConvert.Visible  = True
   frmDataDisplay1.cmdStopConvert.Visible = False
   frmDataDisplay1.cmdRestart.Visible = False
   frmDataDisplay1.cmdReset.Visible = False
   frmDataDisplay1.mnuFileStart.Enabled = True
   frmDataDisplay1.mnuFileRestart.Enabled = False
   frmDataDisplay1.mnuFileReset.Enabled = False
   frmDataDisplay1.mnuFileStop.Enabled = False

   Unload frmConfPerf1
   Unload frmDataDisplay1  '  Unload the forms
End Sub
```

```
Sub cmdQuitNo_Click ()
    frmDataDisplay1.Show  '  Show the form
    ' Enable the timer to continue executing the program
    frmDataDisplay1.tmrConvert.Enabled =  -1
End Sub
```

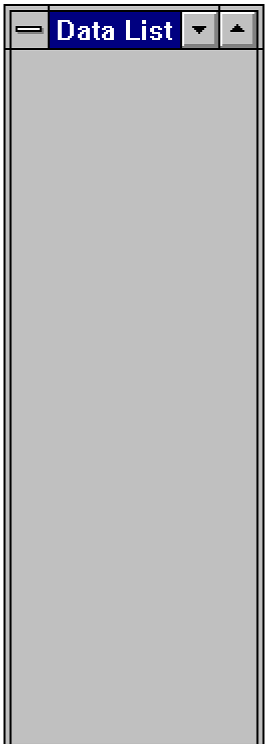
Code 2.8 CONFPERF.FRM (frmConfPerf)



Are You Sure You Want to Quit?

```
Sub Form_Load ()

End Sub

Sub Label1_Click ()

End Sub

Sub cmdQuitYes_Click ()
    '  This dialogue appears prompting the user to confirm the quit command.
    '  If the user clicks Yes, the following code is executed.

    Close Filename  ' any open data files are closed

    mdifrmbase.cmdExit.Visible = True  ' the Exit icon button in the main form appears
    frmDatadisplay.tmrConvert.Enabled = 0   ' the timers in the data display forms are all disabled

    ' Reset all the values in the form frmDataDisplay.
    frmDatadisplay.lblShowIrrad.Visible = False
    frmDatadisplay.lblShowVoltage.Visible = False
    frmDatadisplay.lblShowCurrent.Visible = False
    frmDatadisplay.lblShowPower.Visible = False
    frmDatadisplay.txtInterval.Text = 1000
    frmDatadisplay.cmdStart.Visible = True
    frmDatadisplay.cmdStopConvert.Visible = False
    frmDatadisplay.mnuFileStar t.Enabled = True
    frmDatadisplay.mnuFileQuit.Enabled = False
    frmGraphP.GraphP.GraphData = 0

    Unload frmConfPerf
    Unload frmDatadisplay   ' Unload all the forms
    mdifrmbase.Show ' Display the parent form for the application.
End Sub

Sub cmdQuitNo_Click ()
        frmDatadisplay.Show   ' Show the form

    ' Enable the timer for the program to continue executing
    frmDatadisplay.tmrConvert.Enabled =  -1
End Sub
```

Code 2.9 DATAIRR.FRM (frmDataListI)



```
Sub Form_Load ()
   GridI.ColWidth(col1) = 9800
End Sub

Sub GridI_Click ()

End Sub
```

Code 2.10 DATALC.FRM (frmDataListC)

[Figure as in Code 2.9]

```
Sub Form_Load ()
   GridC.ColWidth(col1) = 9800
End Sub

Sub GridC_Click ()

End Sub
```

Code 2.11 DATALIST.FRM (frmDataListP)

[Figure as in Code 2.9]

```
Sub Form_Load ()
```

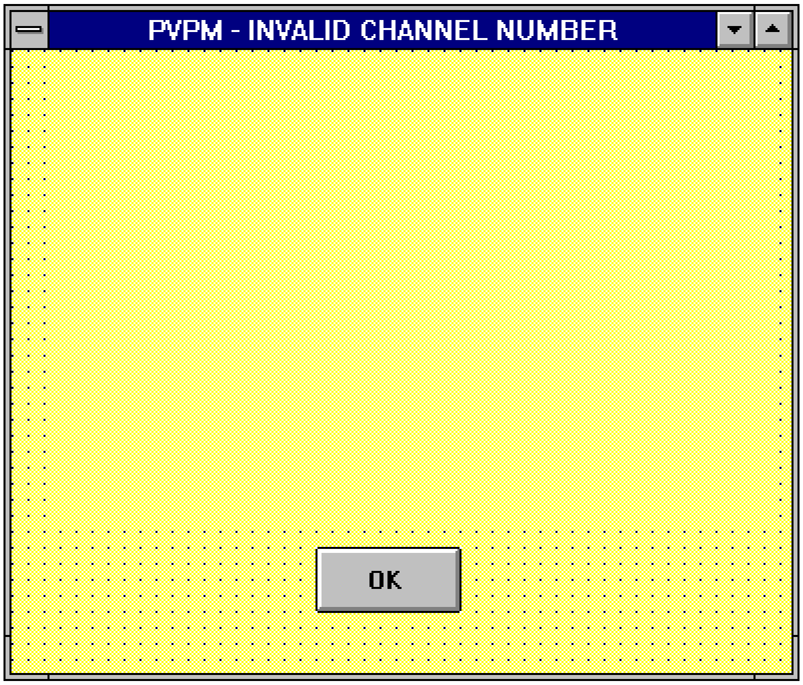
```
End Sub

Sub GridP_Click ()

End Sub
```

Code 2.12 DATAV.FRM (frmDataListV)

[Figure as in Code 2.9]

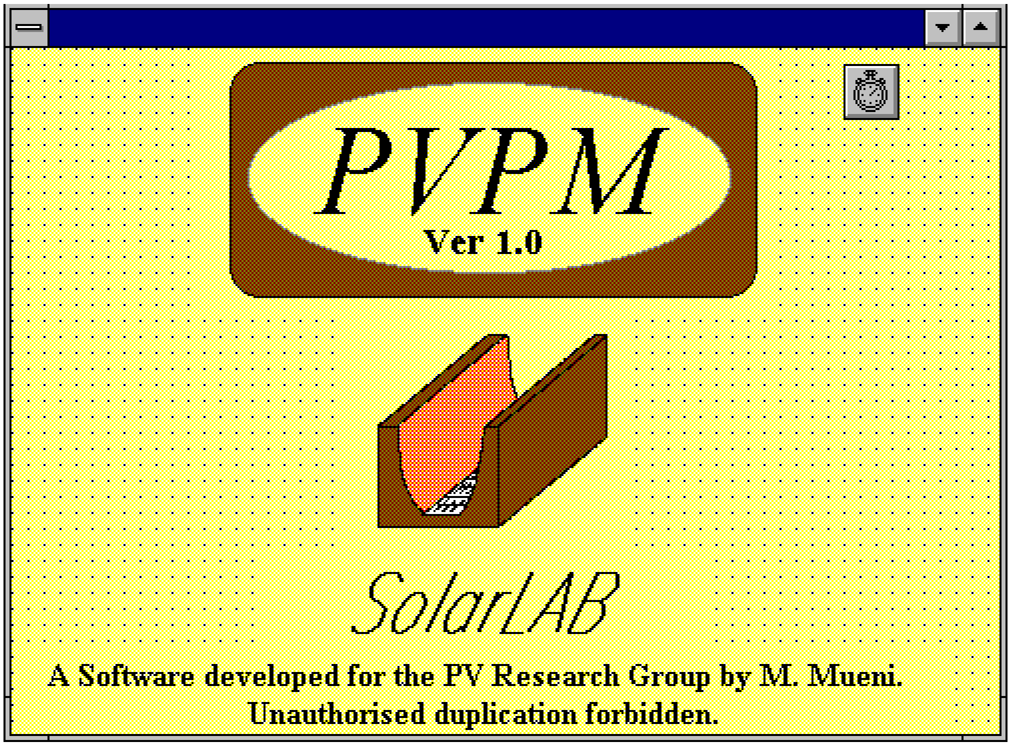
```
Sub Form_Load ()
  GridV.ColWidth(col1) = 9800
End Sub

Sub GridV_Click ()

End Sub
```

Code 2.13 HELPPERF.FRM (frmDD1Help)



```
Sub Form_Load ()

End Sub

Sub lblDD1Help_Click ()

End Sub

Sub cmdHelpOK_Click ()
    ' Exit the help dialogue and return to the error alert
    Unload frmDD1Help
    frmError.Show
End Sub
```

Code 2.14 LOADBMP.FRM (frmLoadBmp)

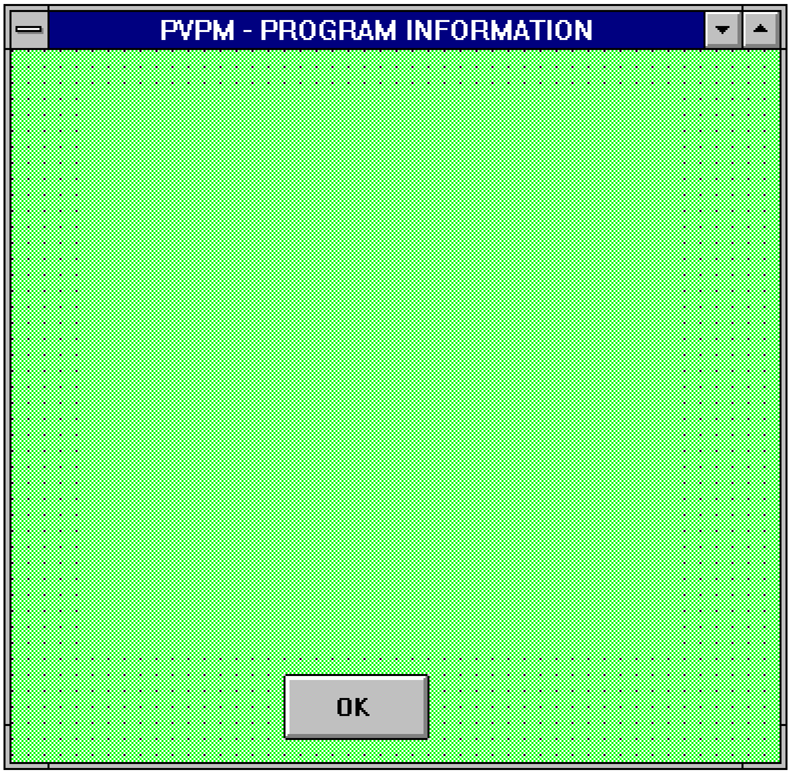


A Software developed for the PV Research Group by M. Mueni.
Unauthorised duplication forbidden.

```
Sub Form_Load ()
    ' This brief sub serves to position the form automatically
    ' in the desired position as soon as it loads
   Top = 300
   ' The statement above sets the vertical position of the form
  'from the top of the screen
   Left = Screen.Width / 2 - Width / 2
  '  This statement then sets the horizontal position of the
   '  form by centering it on the screen
End Sub

Sub Label1_Click ()

End Sub

Sub Label2_Click ()

End Sub

Sub Picture1_Click ()

End Sub

Sub Picture2_Click ()
```

```
End Sub

Sub Timer1_Timer ()
   ' Unload the form once the prescribed time has elapsed.
   Unload frmloadbmp
End Sub
```

Code 2.15 PERF4.FRM (frmProgInfo)



```
Sub Form_Load ()

End Sub

Sub lblProgInfo_Click ()

End Sub

Sub cmdOK_Click ()
   Unload frmProgInfo
   mdifrmBase.Show
End Sub
```

APPENDIX A: CBW.BAS

NOTE THAT THIS FILE SHOULD **NOT** BE MODIFIED AT ALL.

```
'**********************************************************************
'
'     Computer Boards Library
'
'  By: Computer Boards
'     125 High St.
'     Mansfield MA 02048
'     (508) 261_1123
'
'  File: CBW.BAS
'
'  (c) Copyright 1996 by Computer Boards
'     All Rights Reserved
'
' This file contains the Visual BASIC declarations for all Computer
' Boards library commands.   This file should be included in the
' project as a Global Module
'
'************************************************************************

' Current Revision Number
Global Const CURRENTREVNUM = 3.4

' Error Codes
Global Const NOERRORS = 0
Global Const BADBOARD = 1
Global Const DEADDIGITALDEV = 2
Global Const DEADCOUNTERDEV = 3
Global Const DEADDADEV = 4
Global Const DEADADDEV = 5
Global Const NOTDIGITALCONF = 6
Global Const NOTCOUNTERCONF = 7
Global Const NOTDACONF = 8
Global Const NOTADCONF = 9
Global Const NOTMUXCONF = 10
Global Const BADPORTNUM = 11
Global Const BADCOUNTERDEVNUM = 12
Global Const BADDADEVNUM = 13
Global Const BADSAMPLEMODE = 14
Global Const BADINT = 15
Global Const BADADCHAN = 16
Global Const BADCOUNT = 17
Global Const BADCNTRCONFIG = 18
Global Const BADDAVAL = 19
Global Const BADDACHAN =  20
Global Const ALREADYACTIVE = 22
Global Const BADRATE = 24
Global Const COMPATMODE = 25
Global Const TRIGSTATE = 26
Global Const ADSTATUSHUNG = 27
Global Const TOOFEW = 28
Global Const OVERRUN = 29
Global Const BADRANGE = 30
Global Const NOPROGGAIN = 31
Global Const BADFILENAME = 32
Global Const DISKISFULL = 33
Global Const COMPATWARN = 34
Global Const BADPOINTER = 35
Global Const RATEWARNING = 37
Global Const CONVERTDMA = 38
Global Const DTCONNECTERR = 39
Global Const FORECONTINUOUS = 40
Global Const BADBOARDTYPE = 41
Global Const WRONGDIGCONFIG = 42
Global Const NOTCONFIGURABLE = 43
Global Const BADPORTCONFIG = 44
Global Const BADFIRSTPOINT = 45
Global Const ENDOFFILE = 46
Global Const NOT8254CTR = 47
Global Const NOT9513CTR = 48
Global Const BADTRIGTYPE  = 49
Global Const BADTRIGVALUE = 50
Global Const BADOPTION = 52
Global Const BADPRETRIGCOUNT = 53
Global Const BADDIVIDER = 55
Global Const BADSOURCE = 56
Global Const BADCOMPARE = 57
Global Const BADTIMEOFDAY = 58
Global Const BADGATEINTERVAL = 59
Global Const BADGATECNTRL = 60
Global Const BADCOUNTEREDGE = 61
Global Const BADSPCLGATE = 62
Global Const BADRELOAD = 63
Global Const BADRECYCLEFLAG = 64
Global Const BADBCDFLAG = 65
Global Const BADDIRECTION = 66
Global Const BADOUTCONTROL = 67
Global Const BADBITNUMBER = 68
Global Const NONEENABLED = 69
Global Const BADCTRCONTROL = 70
Global Const BADMUXCHAN = 71
Global Const WRONGADRANGE = 72
Global Const OUTOFRANGE = 73
Global Const BADTEMPSCALE = 74
Global Const BADERRCODE = 75
Global Const NOQUEUE = 76
Global Const CONTINUOUSCOUNT = 77
Global Const UNDERRUN = 78
Global Const BADMEMMODE = 79
Global Const FREQOVERRUN = 80
Global Const NOCJCCHAN = 81
Global Const BADCHIPNUM = 82
Global Const DIGNOTENABLED = 83
Global Const CONVERT16BITS = 84
Global Const NOMEMBOARD = 85
Global Const DTACTIVE = 86
Global Const NOTMEMCONF = 87
Global Const ODDCHAN = 88
Global Const CTRNOINIT = 89
Global Const NOT8536CTR = 90
Global Const FREERUNNING = 91
Global Const INTERRUPTED = 92
Global Const NOSELECTORS = 93
Global Const NOBURSTMODE = 94
Global Const NOTWINDOWSFUNC = 95
Global Const NOTSIMULCONF = 96
Global Const EVENODDMISMATCH = 97
Global Const M1RATEWARNING = 98
Global Const NOTRS485 = 99
Global Const NOTDOSFUNC = 100
Global Const RANGEMISMATCH = 101
Global Const CLOCKTOOSLOW = 102
Global Const BADCALFACTORS = 103
Global Const BADCONFIGTYPE = 104
Global Const BADCONFIGITEM = 105
Global Const NOPCMCIABOARD = 106
Global Const NOBACKGROUND = 107
Global Const STRINGTOOSHORT = 108
Global Const CONVERTEXTMEM = 109
Global Const BADEUADD = 110
Global Const DAS16JRRATEWARNING = 111
Global Const DAS08TOOLOWRATE = 112
Global Const AMBIGSENSORONGP = 114
Global Const NOSENSORTYPEONGP = 115
Global Const NOEXTCONTINUOUS = 117

Global Const BADPCMSLOTREF = 118
Global Const AMBIGPCMSLOTREF =  119
Global Const INVALIDPRETRIGCONVERT = 120
Global Const CSSCALLFAILURE = 121
Global Const BADSENSORTYPE = 129

Global Const INTERNALERR = 200
```

```
' Windows error codes
Global Const CANTLOCKDMABUF = 301
Global Const DMAINUSE = 302
Global Const BADMEMHANDLE  = 303
Global Const NOENHANCEDMODE = 304
Global Const NOVDDINSTALLED = 305
Global Const NOWINDOWSMEMORY = 306
Global Const OUTOFDOSMEMORY = 307

' These are the commonly occurring remapped DOS error codes
Global Const DOSBADFUNC = 501
Global Const DOSFILENOTFOUND = 502
Global Const DOSPATHNOTFOUND = 503
Global Const DOSNOHANDLES = 504
Global Const DOSACCESSDENIED = 505
Global Const DOSINVALIDHANDLE = 506
Global Const DOSNOMEMORY = 507
Global Const DOSBADDRIVE = 515
Global Const DOSTOOMANYFILES = 518
Global Const DOSWRITEPROTECT = 519
Global Const DOSDRIVENOTREADY = 521
Global Const DOSSEEKERROR = 525
Global Const DOSWRITEFAULT = 529
Global Const DOSREADFAULT = 530
Global Const DOSGENERALFAULT = 531

Global Const WIN_UNK_INT = 607
Global Const WIN_CANNOT_SET_INT = 608
Global Const WIN_CANNOT_ENABLE_INT = 609
Global Const WIN_CANNOT_RESET_INT = 610
Global Const WIN_CANNOT_DISABLE_INT = 611

Global Const NOTUSED =  -1

' Maximum length of error string
Global Const ERRSTRLEN = 80

' Maximum length of board name string
Global Const BOARDNAMELEN = 25

' Status values
Global Const IDLE = 0
Global Const RUNNING = 1

' Option Flags
Global Const FOREGROUND = &H0
Global Const BACKGROUND = &H1

Global Const SINGLEEXEC = &H0
Global Const CONTINUOUS = &H2
Global Const TIMED = &H0
Global Const EXTCLOCK = &H4

Global Const NOCONVERTDATA = &H0
Global Const CONVERTDATA = &H8

Global Const NODTCONNECT = &H0
Global Const DTCONNECT = &H10

Global Const DEFAULTIO = &H0
Global Const SINGLEIO = &H20
Global Const DMAIO = &H40
Global Const BLOCKIO = &H60

Global Const BYTEXFER = &H0
Global Const WORDXFER = &H100

Global Const INDIVIDUAL = &H0
Global Const SIMULTANEOUS = &H200

Global Const FILTER = &H0
Global Const NOFILTER = &H400

Global Const NORMMEMORY = &H0
Global Const EXTMEMORY = &H800

Global Const BURSTMODE = &H1000

Global Const NOTODINTS = &H2000

Global Const EXTTRIGGER = &H4000

Global Const NOCALIBRATEDATA = &H8000
Global Const CALIBRATEDATA = &H0

Global Const CBENABLED = 1
Global Const CBDISABLED = 0


' Types of error reporting
Global Const DONTPRINT = 0
Global Const PRINTWARNINGS = 1
Global Const PRINTFATAL = 2
Global Const PRINTALL = 3

' Types of error handling
Global Const DONTSTOP = 0
Global Const STOPFATAL = 1
Global Const STOPALL = 2
```

```
' Types of digital input ports
Global Const DIGITALOUT = 1
Global Const DIGITALIN = 2

' DT Modes for cbSetDTMode ()
Global Const DTIN = 0
Global Const DTOUT = 2

Global Const FROMHERE = -1
Global Const GETFIRST = -2
Global Const GETNEXT = -3

'  Temperature scales
Global Const CELSIUS = 0
Global Const FAHRENHEIT = 1
Global Const KELVIN = 2


' Types of digital I/O Ports
Global Const AUXPORT = 1
Global Const FIRSTPORTA = 10
Global Const FIRSTPORTB = 11
Global Const FIRSTPORTCL = 12
Global Const FIRSTPORTCH = 13
Global Const SECONDPORTA = 14
Global Const SECONDPORTB = 15
Global Const SECONDPORTCL = 16
Global Const SECONDPORTCH = 17
Global Const THIRDPORTA = 18
Global Const THIRDPORTB = 19
Global Const THIRDPORTCL = 20
Global Const THIRDPORTCH = 21
Global Const FOURTHPORTA = 22
Global Const FOURTHPORTB = 23
Global Const FOURTHPORTCL = 24
Global Const FOURTHPORTCH = 25
Global Const FIFTHPORTA = 26
Global Const FIFTHPORTB = 27
Global Const FIFTHPORTCL = 28
Global Const FIFTHPORTCH = 29
Global Const SIXTHPORTA = 30
Global Const SIXTHPORTB = 31
Global Const SIXTHPORTCL = 32
Global Const SIXTHPORTCH = 33
Global Const SEVENTHPORTA = 34
Global Const SEVENTHPORTB = 35
Global Const SEVENTHPORTCL = 36
Global Const SEVENTHPORTCH = 37
Global Const EIGHTHPORTA = 38
Global Const EIGHTHPORTB = 39
Global Const EIGHTHPORTCL = 40
Global Const EIGHTHPORTCH = 41


' Selectable A/D Ranges codes
Global Const BIP10VOLTS = 1                 ' Bipolar Ranges (-10 to +10 Volts)
Global Const BIP5VOLTS = 0
Global Const BIP2PT5VOLTS  = 2
Global Const BIP1PT25VOLTS = 3
Global Const BIP1VOLTS = 4
Global Const BIPPT625VOLTS = 5
Global Const BIPPT5VOLTS = 6
Global Const BIPPT1VOLTS = 7
Global Const BIPPT05VOLTS = 8
Global Const BIPPT01VOLTS = 9
Global Const BIPPT005VOLTS = 10
Global Const BIP1PT67VOLTS = 11

Global Const UNI10VOLTS = 100                ' Unipolar Ranges (0 to 10 Volts)
Global Const UNI5VOLTS = 101
Global Const UNI2PT5VOLTS = 102
Global Const UNI2VOLTS = 103
Global Const UNI1PT25VOLTS = 104
Global Const UNI1VOLTS = 105
Global Const UNIPT1VOLTS = 106
Global Const UNIPT01VOLTS = 107
Global Const UNIPT02VOLTS = 108
Global Const UNI1PT67VOLTS = 109

Global Const MA4TO20 = 200                  ' Current Ranges (4 to 20 ma )
Global Const MA2to10 = 201
Global Const MA1TO5 = 202
Global Const MAPT5TO2PT5 = 203

' Types of D/A
Global Const ADDA1 = 0
Global Const ADDA2 = 1

' 8536 counter output 1 control
Global Const NOTLINKED = 0
Global Const GATECTR2 = 1
Global Const TRIGCTR2 = 2
Global Const INCTR2 = 3

' Types of 8254 Counter configurations
Global Const HIGHONLASTCOUNT = 0
Global Const ONESHOT = 1
Global Const RATEGENERATOR = 2
Global Const SQUAREWAVE = 3
Global Const SOFTWARESTROBE = 4
Global Const HARDWARESTROBE = 5

' Where to reload from for 9513 counters
Global Const LOADREG = 0
Global Const LOADANDHOLDREG = 1

' Counter recycle modes
Global Const ONETIME = 0
Global Const RECYCLE = 1

' Direction of counting for 9513 counters
Global Const COUNTDOWN = 0
Global Const COUNTUP = 1

' Types of count detection for 9513 counters
Global Const POSITIVEEDGE = 0
Global Const NEGATIVEEDGE = 1

' Counter output control
Global Const ALWAYSLOW = 0
Global Const HIGHPULSEONTC = 1
Global Const TOGGLEONTC = 2
Global Const DISCONNECTED = 4
Global Const LOWPULSEONTC = 5
Global Const HIGHUNTILTC  = 6

' Counter input sources
Global Const TCPREVCTR = 0
Global Const CTRINPUT1 = 1
Global Const CTRINPUT2 = 2
Global Const CTRINPUT3 = 3
Global Const CTRINPUT4 = 4
Global Const CTRINPUT5 = 5
Global Const GATE1 = 6
Global Const GATE2 = 7
Global Const GATE3  = 8
Global Const GATE4 = 9
Global Const GATE5 = 10
Global Const FREQ1 = 11
Global Const FREQ2 = 12
Global Const FREQ3 = 13
Global Const FREQ4 = 14
Global Const FREQ5 = 15
Global Const CTRINPUT6 = 101
Global Const CTRINPUT7 = 102
Global Const CTRINPUT8 = 103
Global Const CTRINPUT9 = 104
Global Const CTRINPUT10 = 105
Global Const GATE6 = 106
Global Const GATE7 = 107
Global Const GATE8 = 108
Global Const GATE9 = 109
Global Const GATE10 = 110
Global Const FREQ6 = 111
Global Const FREQ7 = 112
Global Const FREQ8  = 113
Global Const FREQ9 = 114
Global Const FREQ10 = 115

Global Const CTRINPUT11 = 201
Global Const CTRINPUT12 = 202
Global Const CTRINPUT13 = 203
Global Const CTRINPUT14 = 204
Global Const CTRINPUT15 = 205
Global Const GATE11 = 206
Global Const GATE12 =  207
Global Const GATE13 = 208
Global Const GATE14 = 209
Global Const GATE15 = 210
Global Const FREQ11 = 211
Global Const FREQ12 = 212
Global Const FREQ13 = 213
Global Const FREQ14 = 214
Global Const FREQ15 = 215
Global Const CTRINPUT16 = 301
Global Const CTRINPUT17 = 302
Global Const CTRINPUT18 = 303
Global Const CTRINPUT19 = 304
Global Const CTRINPUT20 = 305
Global Const GATE16 = 306
Global Const GATE17 = 307
Global Const GATE18 = 308
Global Const GATE19 = 309
Global Const GATE20 = 310
Global Const FREQ16  = 311
Global Const FREQ17 = 312
Global Const FREQ18 = 313
Global Const FREQ19 = 314
Global Const FREQ20 = 315

' Counter registers
Global Const LOADREG1 = 1
Global Const LOADREG2 = 2
Global Const LOADREG3 = 3
Global Const LOADREG4 = 4
Global Const LOADREG5  = 5
Global Const LOADREG6 = 6
Global Const LOADREG7 = 7
Global Const LOADREG8 = 8
Global Const LOADREG9 = 9
Global Const LOADREG10 = 10

Global Const LOADREG11 = 11
Global Const LOADREG12 = 12
Global Const LOADREG13 = 13
Global Const LOADREG14 = 14
Global Const LOADREG15 = 15
Global Const LOADREG16 = 16
Global Const LOADREG17 = 17
Global Const LOADREG18 = 18
Global Const LOADREG19 = 19
Global Const LOADREG20 = 20

Global Const HOLDREG1 = 101
Global Const HOLDREG2 = 102
Global Const HOLDREG3 = 103
Global Const HOLDREG4 = 104
Global Const HOLDREG5 = 105
Global Const HOLDREG6 = 106
Global Const HOLDREG7 = 107
Global Const HOLDREG8 = 108
Global Const HOLDREG9 = 109
Global Const HOLDREG10 = 110

Global Const HOLDREG11 = 111
Global Const HOLDREG12 = 112
Global Const HOLDREG13 = 113
Global Const HOLDREG14 = 114
Global Const HOLDREG15 = 115
Global Const HOLDREG16 = 116
Global Const HOLDREG17 = 117
Global Const HOLDREG18 = 118
Global Const HOLDREG19 = 119
Global Const HOLDREG20 = 120

Global Const ALARM1CHIP1 = 201
Global Const ALARM2CHIP1 = 202
Global Const ALARM1CHIP2 = 301
Global Const ALARM2CHIP2 = 302
Global Const ALARM1CHIP3 = 401
Global Const ALARM2CHIP3 = 402
Global Const ALARM1CHIP4 = 501
Global Const ALARM2CHIP4 = 502

'  Counter Gate Control
Global Const NOGATE = 0
Global Const AHLTCPREVCTR = 1
Global Const AHLNEXTGATE = 2
Global Const AHLPREVGATE = 3
Global Const AHLGATE = 4
Global Const ALLGATE = 5
Global Const AHEGATE = 6
Global Const ALEGATE = 7

' Types of triggers
Global Const TRIGABOVE = 0
Global Const TRIGBELOW = 1

' Types of configuration information
Global Const GLOBALINFO = 1
Global Const BOARDINFO = 2
Global Const DIGITALINFO = 3
Global Const COUNTERINFO = 4
Global Const EXPANSIONINFO = 5
Global Const MISCINFO = 6

' Types of global configuration information
Global Const GIVERSION = 36          ' Config file format version number
Global Const GINUMBOARDS = 38        ' Maximum number of boards
Global Const GINUMEXPBOARDS = 40     ' Maximum number of expansion boards

' Types of board configuration information
Global Const BIBASEADR = 0           ' Base Address
Global Const BIBOARDTYPE = 1         ' Board Type (0x101 - 0x7FFF)
Global Const BIINTLEVEL = 2          ' Interrupt level
Global Const BIDMACHAN = 3           ' DMA channel
Global Const BIINITIALIZED = 4       ' TRUE or FALSE
Global Const BICLOCK = 5             ' Clock freq (1, 10 or bus)
Global Const BIRANGE = 6             ' Switch selectable range
Global Const BINUMADCHANS = 7        ' Number of A/D channels
Global Const BIUSESEXPS = 8          ' Supports expansion boards TRUE/FALSE
Global Const BIDINUMDEVS = 9         ' Number of digital devices
Global Const BIDIDEVNUM = 10         ' Index into digital information
Global Const BICINUMDEVS = 11        ' Number of counter devices
Global Const BICIDEVNUM = 12         ' Index into counter information
Global Const BINUMDACHANS = 13       ' Number of D/A channels
Global Const BIWAITSTATE = 14        ' Wait state enabled TRUE/FALSE
Global Const BINUMIOPORTS = 15       ' I/O address space used by board
Global Const BIPARENTBOARD = 16      ' Board number of parent board
Global Const BIDTBOARD = 17          ' Board number of connected DT board

' Types of digital device information
Global Const DIBASEADR = 0           ' Base address
Global Const DIINITIALIZED = 1       ' TRUE or FALSE
Global Const DIDEVTYPE = 2           ' AUXPORT or xPORTA - CH
Global Const DIMASK = 3              ' Bit mask for this port
Global Const DIREADWRITE = 4         ' Read required before write
Global Const DICONFIG = 5            ' Current configuration
Global Const DINUMBITS =  6          ' Number of bits in port
Global Const DICURVAL = 7          ' Current value of outputs

' Types of counter device information
Global Const CIBASEADR = 0          ' Base address
Global Const CIINITIALIZED = 1      ' TRUE or FALSE
Global Const CICTRTYPE = 2          ' Counter type 8254, 9513 or 8536
Global Const CICTRNUM = 3           ' Which counter on chip
Global Const CICONFIGBYTE = 4       ' Configuration byte

' Types of expansion board information
Global Const XIBOARDTYPE = 0        ' Expansion board type
Global Const XIMUXADCHAN1 = 1       ' 0 - 15
Global Const XIMUXADCHAN2 = 2       ' 0 - 15 or NOTUSED
Global Const XIRANGE1 = 3           ' Range (gain) of low 16 chans
Global Const XIRANGE2 = 4           ' Range (gain) of high 16 chans
Global Const XICJCCHAN = 5          ' 0 - 15 or NOTUSED
Global Const XITHERMTYPE = 6        ' TYPEJ, TYPEK, TYPEB, TYPET, TYPEE, TYPER,_
   or TYPES
Global Const XINUMEXPCHANS = 7      ' Number of expansion channels on board
Global Const XIPARENTBOARD = 8      ' Board number of parent A/D board
Global Const XISPARE0 = 9           ' 16 words of misc options

Declare Function cbAConvertData% Lib "cbw.dll" (ByVal BoardNum%, ByVal NumPoints&,_
  ADData%, ChanTags%)
Declare Function cbACalibrateData% Lib "cbw.dll" (ByVal BoardNum%,  ByVal NumPoints&,_
  ByVal Gain%, ADData%)
Declare Function cbAConvertPretrigData% Lib "cbw.dll" (ByVal BoardNum%, ByVal_
  PretrigCount&, ByVal TotalCount&, ADData%, ChanTags%)
Declare Function cbAIn% Lib "cbw.dll" (ByVal BoardNum%, ByVal Chan%, ByVal  Gain%,_
  DataValue%)
Declare Function cbAInScan% Lib "cbw.dll" (ByVal BoardNum%, ByVal LowChan%, ByVal_
  HighChan%, ByVal CBCount&, CBRate&, ByVal Gain%, ByVal MemHandle%, ByVal_
  Options%)
Declare Function cbALoadQueue% Lib "cbw.dll" (ByVal BoardNum %, ChanArray%,_
  GainArray%, ByVal NumChans%)
Declare Function cbAOut% Lib "cbw.dll" (ByVal BoardNum%, ByVal Chan%, ByVal Gain%,_
  ByVal DataValue%)
Declare Function cbAOutScan% Lib "cbw.dll" (ByVal BoardNum%, ByVal LowChan%,_
  ByVal HighChan%, ByVal CBCount&, CBRate&, ByVal Gain%, ByVal MemHandle%,_
  ByVal Options%)
Declare Function cbAPretrig% Lib "cbw.dll" (ByVal BoardNum%, ByVal LowChan%, ByVal_
 HighChan%, PretrigCount&, CBCount&, CBRate&, ByVal Gain%, ByVal MemHandle%,_
 ByVal Options%)
Declare Function cbATrig% Lib "cbw.dll" (ByVal BoardNum%, ByVal Chan%, ByVal_
 TrigType%, ByVal TrigValue%, ByVal Gain%, DataValue%)
Declare Function cbC8254Config% Lib "cbw.dll" (ByVal BoardNum%, ByVal CounterNum%,_
 ByVal Config%)
Declare Function cbC8536C onfig% Lib "cbw.dll" (ByVal BoardNum%, ByVal CounterNum%,_
 ByVal OutputControl%, ByVal RecycleMode%, ByVal Retrigger%)
Declare Function cbC9513Config% Lib "cbw.dll" (ByVal BoardNum%, ByVal CounterNum%,_
  ByVal GateControl%, ByVal CounterEdge%, ByVal Cou ntSource%, ByVal SpecialGate%,_
  ByVal Reload%, ByVal RecycleMode%, ByVal BCDMode%, ByVal CountDirec%, ByVal_
  OutputCtrl%)
Declare Function cbC8536Init% Lib "cbw.dll" (ByVal BoardNum%, ByVal ChipNum%,_
  ByVal Ctr1Output%)
Declare Function cbC9513Init% Lib "cbw.dll" (ByVal BoardNum%, ByVal ChipNum%,_
  ByVal FOutDivider%, ByVal FOutSource%, ByVal Compare1%, ByVal Compare2%, ByVal_
  TimeOfDay%)
Declare Function cbCStoreOnInt% Lib "cbw.dll" (ByVal BoardNum%, ByVal IntCount%,_
  CntrControl%, ByVal DataBuf fer%)
Declare Function cbCFreqIn% Lib "cbw.dll" (ByVal BoardNum%, ByVal SigSource%, ByVal_
  GateInterval%, CBCount%, Freq&)
Declare Function cbCIn% Lib "cbw.dll" (ByVal BoardNum%, ByVal CounterNum%,_
  CBCount%)
Declare Function cbCLoad% Lib "cbw.dll" (By Val BoardNum%, ByVal RegNum%, ByVal_
  LoadValue%)
Declare Function cbDBitIn% Lib "cbw.dll" (ByVal BoardNum%, ByVal PortType%, ByVal_
  BitNum%, BitValue%)
Declare Function cbDBitOut% Lib "cbw.dll" (ByVal BoardNum%, ByVal PortType%, ByVal_
  BitNum%, ByVal BitValue%)
Declare Function cbDConfigPort% Lib "cbw.dll" (ByVal BoardNum%, ByVal PortNum%,_
  ByVal Direction%)
Declare Function cbDeclareRevision% Lib "cbw.dll" (RevNum!)
Declare Function cbDIn% Lib "cbw.dll" (ByVal BoardNum%, ByVal PortNum%, DataValue%)
Declare Function cbDInScan% Lib "cbw.dll" (ByVal BoardNum%, ByVal PortNum%, ByVal_
  CBCount&, CBRate&, ByVal MemHandle%, ByVal Options%)
Declare Function cbDOut% Lib "cbw.dll" (ByVal BoardNum%, ByVal PortNum%, ByVal_
  DataValue%)
Declare Function cbDOut Scan% Lib "cbw.dll" (ByVal BoardNum%, ByVal PortNum%, ByVal_
  CBCount&, CBRate&, ByVal MemHandle%, ByVal Options%)
Declare Function cbErrHandling% Lib "cbw.dll" (ByVal ErrReporting%, ByVal ErrHandling%)
Declare Function cbFileAInScan% Lib "cbw.dll" (ByVal  BoardNum%, ByVal LowChan%,_
  ByVal HighChan%, ByVal CBCount&, CBRate&, ByVal Gain%, ByVal FileName$, ByVal_
  Options%)
Declare Function cbFileGetInfo% Lib "cbw.dll" (ByVal FileName$, LowChan%, HighChan%,_
  PretrigCount&, TotalCount&, CBRate&, Gain%)
Declare Function cbFilePretrig% Lib "cbw.dll" (ByVal BoardNum%, ByVal LowChan%,_
  ByVal HighChan%, PretrigCount&, CBCount&, CBRate&, ByVal Gain%, ByVal FileName$,_
  ByVal Options%)
Declare Function cbFileRead% Lib "cbw.dll" (ByVal FileName$, ByVal FirstPoi nt&,_
  NumPoints&, DataBuffer%)
Declare Function cbGetErrMsg% Lib "cbw.dll" (ByVal ErrCode%, ByVal ErrMsg$)
Declare Function cbGetRevision% Lib "cbw.dll" (DLLRevNum!, VXDRevNum!)
Declare Function cbGetStatus% Lib "cbw.dll" (ByVal BoardNum%, Status%, CurCo unt&,_
  CurIndex&)
Declare Function cbStopBackground% Lib "cbw.dll" (ByVal BoardNum%)
Declare Function cbMemSetDTMode% Lib "cbw.dll" (ByVal BoardNum%, ByVal Mode%)
Declare Function cbMemReset% Lib "cbw.dll" (ByVal BoardNum%)
Declare Function cbMemRead% Li b "cbw.dll" (ByVal BoardNum%, DataBuffer%, ByVal_
  FirstPoint&, ByVal CBCount&)
```

```
Declare Function cbMemWrite% Lib "cbw.dll" (ByVal BoardNum%, DataBuffer%, ByVal_
  FirstPoint&, ByVal CBCount&)
Declare Function cbMemReadPretrig% Lib "cbw.dll" (ByVal BoardNu m%, DataBuffer%,_
  ByVal FirstPoint&, ByVal CBCount&)
Declare Function cbRS485% Lib "cbw.dll" (ByVal BoardNum%, ByVal Transmit%, ByVal_
  Receive%)
Declare Function cbTIn% Lib "cbw.dll" (ByVal BoardNum%, ByVal Chan%, ByVal_
   CBScale%, TempValue!, ByVal Options%)
Declare Function cbTInScan% Lib "cbw.dll" (ByVal BoardNum%, ByVal LowChan%, ByVal_
   HighChan%, ByVal CBScale%, DataBuffer!, ByVal Options%)
Declare Function cbWinBufToArray% Lib "cbw.dll" (ByVal MemHandle%, DataBuffer%,_
   ByVal FirstPoint&, ByVal CBCount&)
Declare Function cbWinArrayToBuf% Lib "cbw.dll" (DataBuffer%, ByVal MemHandle%,_
   ByVal FirstPoint&, ByVal CBCount&)
Declare Function cbWinBufAlloc% Lib "cbw.dll" (ByVal NumPoints&)
Declare Function cbWinBufFree% Lib "cbw.dll" (ByVal MemHa ndle%)
Declare Function cbInByte% Lib "cbw.dll" (ByVal BoardNum%, ByVal PortNum%)
Declare Function cbOutByte% Lib "cbw.dll" (ByVal BoardNum%, ByVal PortNum%, ByVal_
  PortVal%)
Declare Function cbInWord% Lib "cbw.dll" (ByVal BoardNum%, ByVal PortNum%)
Declare Function cbOutWord% Lib "cbw.dll" (ByVal BoardNum%, ByVal PortNum%, ByVal_
  PortVal%)
Declare Function cbGetConfig% Lib "cbw.dll" (ByVal InfoType%, ByVal BoardNum%, ByVal_
  DevNum%, ByVal ConfigItem%, ConfigVal%)
Declare Function cbSetConfig% Lib "cb w.dll" (ByVal InfoType%, ByVal BoardNum%, ByVal_
  DevNum%, ByVal ConfigItem%, ByVal ConfigVal%)
Declare Function cbToEngUnits% Lib "cbw.dll" (ByVal BoardNum%, ByVal Range%, ByVal_
  DataVal%, EngUnits!)
Declare Function cbFromEngUnits% Lib "cbw.dll" (ByVa l BoardNum%, ByVal Range%,_
  ByVal EngUnits!, DataVal%)
Declare Function cbGetBoardName% Lib "cbw.dll" (ByVal BoardNum%, ByVal BoardName$)
```