Make your own software for the PCSU1000 with the Dynamic Link Library PCSU1000D.DLL

The PCSU1000D.DLL is a 32 bit Windows DLL. This document describes all functions and procedures of the DLL that are available for your application programme. Calling the functions and procedures exported by the DLL, you may write custom Windows (98SE, 2000, Me, XP) applications in Delphi, Visual Basic, C++ Builder or any other 32-bit Windows application development tool that supports calls to a DLL.

A complete overview of the procedures and functions that are exported by the PCSU1000D.DLL follows. At the end of this document there are listings of example programmes in order to gain an insight as to how to construct your own application programmes. The examples are written in Delphi and Visual Basic. In the listings there are full declarations for the DLL function and procedures.

Note that all the examples in the function and procedure description section are written for Delphi.

Overview of the Procedures and Functions of the PCSU1000D.DLL

General

`RunOn(Run: Boolean)`	*Set PCSU1000 Run mode on or off*
`SingleOn(Single: Boolean)`	*Set PCSU1000 Single mode on or off*
`Show_PCSU1000(Visible: Boolean);`	*Show or hide the PCSU1000 user interface*
`Start_PCSU1000;`	*Starts the PCSU1000.exe program*
`Stop_PCSU1000;`	*Closes the PCSU1000.exe program*

Setup

`Voltage1(Volts:Longint)`	*Set V/div scale of Ch1*
`Voltage2(Volts:Longint)`	*Set V/div scale of Ch2*
`Time(TpDiv:Longint)`	*Set Time/div scale*
`TrgLevel(TrgLevel:Longint)`	*Set trigger level*
`TrgEdge(Positive_Negative:Longint)`	*Set trigger edge*
`TrgOn(trg_on: Boolean)`	*Set trigger on or off*
`TrgSource(CH1_CH2_Ext:Longint)`	*Set trigger source*
`YPosition1(y_pos:Longint);`	*Set trace Y-position of Ch1*
`YPosition2(y_pos:Longint);`	*Set trace Y-position of Ch2*
`Coupling1(AC_DC_GND:Longint)`	*Set the input coupling of Ch1*
`Coupling2(AC_DC_GND:Longint)`	*Set the input coupling of Ch2*

Get setup

`GetSettings(Settings: Pointer)`	*Get the oscilloscope settings*

Data read

```
Use DSOLink.DLL functions:
     ReadCh1(Buffer: Pointer)
     ReadCh2(Buffer: Pointer)
     DataReady
```

Please note: The *PCLAB2000SE.EXE* program must be run (and closed) at least one time after its installation before using these function calls. After doing this your application software finds the path to the folder where you installed the Velleman software package. Then you need only to put the *PCSU1000D.DLL* and *DSOLINK.DLL* to your application folder or to the Windows' *SYSTEM32* folder.

Procedures and Functions of the PCSU1000D.DLL

Start_PCSU1000

Syntax
```
PROCEDURE Start_PCSU1000;
```

Description
Starts the PCSU1000.EXE program.

Example
```
procedure TForm1.Button1Click(Sender: TObject);
begin
  Start_PCSU1000;
end;
```

Stop_PCSU1000

Syntax
```
PROCEDURE Stop_PCSU1000;
```

Description
Closes the PCSU1000.EXE program.

Example
```
procedure TForm1.FormClose(Sender: TObject; var Action:
TCloseAction);
begin
  Stop_PCSU1000;
end;
```

Show_PCSU1000

Syntax
```
procedure Show_PCSU1000(Visible: Boolean);
```

Parameter
`Visible`: Boolean true displays the user interface. False hides the user interface.

Description
Displays or hides the PCSU1000 user interface on the screen.

Example
```
procedure TForm1.ShowPCSU1000Click(Sender: TObject);
begin
  Show_PCSU1000(true);
end;

procedure TForm1.HidePCSU1000Click(Sender: TObject);
begin
   Show_PCSU1000(false);
end;
```

RunOn

Syntax
```
PROCEDURE RunOn(Run: Boolean);
```

Parameter
`Run`: Boolean true sets the scope to Run mode. False stops Run mode.

Description
Set Run mode on or off.

Example
```
procedure TForm1.RunClick(Sender: TObject);
begin
    RunOn(Run.down);
end;
```

SingleOn

Syntax
```
PROCEDURE SingleOn(Single: Boolean);
```

Parameter
`Single`: Boolean true sets the scope to Single mode. False stops Single mode.

Description
Set Single mode on or off.

Example
```
procedure TForm1.SingleClick(Sender: TObject);
begin
  SingleOn(true);
end;
```

Voltage1, Voltage2

Syntax
```
PROCEDURE Voltage1(Volts:Longint);
PROCEDURE Voltage2(Volts:Longint);
```

Parameter
`Volts`: The index of the V/div range.
0 = 2 V/div
1 = 1 V/div
2 = 0.5 V/div
3 = 0.2 V/div
4 = 0.1 V/div
5 = 50 mV/div
6 = 20 mV/div
7 = 10 mV/div
8 = 5 mV/div

Description
Set the V/div setting of the PCSU1000 oscilloscope.

Example
```
procedure TForm1.VoltageRangeClick(Sender: TObject);
begin
  Voltage((sender as TSpeedButton).tag);
```

```
end;
```

Time

Syntax
```
PROCEDURE Time(TpDiv:Longint);
```

Parameter
`TpDiv:` The index of the time/div setting rate.
0 = 100 ms/div
1 = 50 ms/div
2 = 20 ms/div
3 = 10 ms/div
4 = 5 ms/div
5 = 2 ms/div
6 = 1 ms/div
7 = 0.5 ms/div
8 = 0.2 ms/div
9 = 0.1 ms/div
10 = 50 us/div
11 = 20 us/div
12 = 10 us/div
13 = 5 us/div
14 = 2 us/div
15 = 0.2 us/div at 1 GS/s
16 = 0.1us/div at 1 GS/s
17 = 0.05 us/div at 1 GS/s
18 = 0.02 us/div at 1 GS/s

Description
Set the Time/div setting of the PCSU1000 oscilloscope.

Example
```
procedure TForm1.TimeRangeClick(Sender: TObject);
begin
  Time((sender as TSpeedButton).tag);
end;
```

Coupling

Syntax
```
PROCEDURE Coupling(AC_DC_GND:Longint);
```

Parameter
`AC_DC_GND:` The index of the coupling type.
0 = AC
1 = DC
2 = GND

Description
Set the input couplin setting of the PCSU1000 oscilloscope.

Example
```
procedure TForm1.SelectCouplingClick(Sender: TObject);
begin
  Coupling((sender as TSpeedButton).tag);
end;
```

TriggerLevel

Syntax
```
PROCEDURE TriggerLevel(TrgLevel:Longint);
```

Parameter
`TrgLevel:` The triggering level value between 0 and 255.

Description
Set the triggering level of the PCSU1000 oscilloscope in steps of half a division on the screen.

Example
```
procedure TForm1.TrgLevelChange(Sender: TObject);
begin
  TriggerLevel(TrgLevel.position);
end;
```

YPosition1, YPosition2

Syntax
```
PROCEDURE YPosition1(Position.position);
PROCEDURE YPosition2(Position.position);
```

Parameter
`Position:` The Y-position of the trace (ground reference), value between -128 and 127.

Description
Adds an offset to the input signal.

Example
```
procedure TForm1.SetPositionChange(Sender: TObject);
begin
  YPosition(SetPosition.position);
end;
```

TrgOn

Syntax
```
PROCEDURE TrgOn(trg_on: Boolean);
```

Parameter
`trg_on:` Boolean TRUE sets the triggering on and FALSE sets the triggering off.

Description
Set the PCSU1000 trigger on or off.

Example
```
procedure TForm1.TriggerOnClick(Sender: TObject);
begin
  TrgOn(true);
end;

procedure TForm1.TriggerOffClick(Sender: TObject);
begin
  TrgOn(false);
```

```
end;
```

TrgEdge

Syntax
```
PROCEDURE TrgEdge(Positive_Negative:Longint)
```

Parameter
`Positive_Negative:` Index of the trigger edge.
0 = Negative
1 = Positive

Description
Set the PCSU1000 trigger edge.

Example
```
procedure TForm1.TriggerEdgeClick(Sender: TObject);
begin
  TrgEdge((sender as TSpeedButton).tag);
end;
```

TrgSource

Syntax
```
PROCEDURE TrgSource(CH1_CH2_Ext:Longint)
```

Parameter
`CH1_CH2_Ext:` Index of the trigger source.
0 = Ch1
1 = Ch2
2 = Ext

Description
Set the PCSU1000 trigger source.

Example
```
procedure TForm1.SpeedButton27Click(Sender: TObject);
begin
  TrgSource((sender as TSpeedButton).tag);
end;
```

GetSettings

Syntax
```
FUNCTION GetSettings(SettingsArray: Pointer): Boolean
```

Parameter
`SettingsArray:` Pointer to an array of 11 long integers (32 bit). The elements of the array indicate the current control settings on the PCSU1000 user interface. The values corresponding the settings are same as previously described in this document.
```
SettingsArray[0]: Volts/Div setting of CH1
SettingsArray[1]: Volts/Div setting of CH2
SettingsArray[2]: Time/Div setting
SettingsArray[3]: Y-position of CH1
SettingsArray[4]: Y-position of CH2
```

```
SettingsArray[5]: Coupling of CH1
SettingsArray[6]: Coupling of CH2
SettingsArray[7]: Trigger On/Off
SettingsArray[8]: Trigger source
SettingsArray[9]: Trigger edge
SettingsArray[10]: Trigger level
```

Description
Get the current settings of each of the user-modifiable controls on the PCSU1000 user interface.

Result
`Boolean`: TRUE means that the PCSU1000 oscilloscope program is running, FALSE means that it is not running.

Example

```
procedure TForm1.Button4Click(Sender: TObject);
var SettingsArray:array [0..10] of longint;
begin
  if GetSettings(@SettingsArray) then
  begin
    case SettingsArray[0] of      // get the V/div setting of CH1
      0: SpeedButton1.Down:=true;
      1: SpeedButton2.Down:=true;
      2: SpeedButton3.Down:=true;
      3: SpeedButton4.Down:=true;
      4: SpeedButton5.Down:=true;
      5: SpeedButton6.Down:=true;
      6: SpeedButton7.Down:=true;
      7: SpeedButton8.Down:=true;
      8: SpeedButton9.Down:=true;
    end;
  end;
end;
```

Using the PCSU1000D.DLL in Delphi

In this application example there are the declarations of the *PCSU1000D.DLL* and *DSOLINK.DLL* procedures and functions and an example how to use these function calls.

```
unit PCS_Runx;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, Buttons, ComCtrls, ExtCtrls;

type
  TForm1 = class(TForm)
    Memo1: TMemo;
    Run: TSpeedButton;
    Single: TButton;
    ShowPCS100: TButton;
    HidePCS100: TButton;
    DataRead: TButton;
    GroupBox1: TGroupBox;
    Label11: TLabel;
    Label14: TLabel;
    Label15: TLabel;
    Label17: TLabel;
    LabelCh2: TLabel;
    LabelCh1: TLabel;
    TrackBar1: TTrackBar;
    TrackBar2: TTrackBar;
```

```
SpeedButton1: TSpeedButton;
SpeedButton2: TSpeedButton;
SpeedButton3: TSpeedButton;
SpeedButton4: TSpeedButton;
SpeedButton5: TSpeedButton;
SpeedButton6: TSpeedButton;
SpeedButton7: TSpeedButton;
SpeedButton8: TSpeedButton;
SpeedButton9: TSpeedButton;
SpeedButton10: TSpeedButton;
SpeedButton11: TSpeedButton;
SpeedButton12: TSpeedButton;
SpeedButton13: TSpeedButton;
SpeedButton14: TSpeedButton;
SpeedButton15: TSpeedButton;
SpeedButton16: TSpeedButton;
SpeedButton17: TSpeedButton;
SpeedButton18: TSpeedButton;
SpeedButton19: TSpeedButton;
SpeedButton20: TSpeedButton;
SpeedButton21: TSpeedButton;
SpeedButton22: TSpeedButton;
SpeedButton23: TSpeedButton;
SpeedButton24: TSpeedButton;
Time_div: TGroupBox;
Trigger: TGroupBox;
Label4: TLabel;
Label42: TLabel;
Label43: TLabel;
Label16: TLabel;
TrackBar3: TTrackBar;
SpeedButton25: TSpeedButton;
SpeedButton26: TSpeedButton;
SpeedButton27: TSpeedButton;
SpeedButton28: TSpeedButton;
SpeedButton29: TSpeedButton;
SpeedButton30: TSpeedButton;
SpeedButton31: TSpeedButton;
SpeedButton32: TSpeedButton;
SpeedButton33: TSpeedButton;
SpeedButton34: TSpeedButton;
SpeedButton35: TSpeedButton;
SpeedButton36: TSpeedButton;
SpeedButton37: TSpeedButton;
SpeedButton38: TSpeedButton;
SpeedButton39: TSpeedButton;
SpeedButton40: TSpeedButton;
SpeedButton41: TSpeedButton;
SpeedButton42: TSpeedButton;
SpeedButton43: TSpeedButton;
SpeedButton44: TSpeedButton;
SpeedButton45: TSpeedButton;
SpeedButton46: TSpeedButton;
Button1: TButton;
Button2: TButton;
Button3: TButton;
Label1: TLabel;
SpeedButton47: TSpeedButton;
SpeedButton48: TSpeedButton;
SpeedButton49: TSpeedButton;
SpeedButton50: TSpeedButton;
Label2: TLabel;
Button4: TButton;
procedure DataReadClick(Sender: TObject);
procedure RunClick(Sender: TObject);
procedure SingleClick(Sender: TObject);
procedure ShowPCS100Click(Sender: TObject);
procedure HidePCS100Click(Sender: TObject);
procedure TrackBar1Change(Sender: TObject);
procedure TrackBar2Change(Sender: TObject);
procedure SpeedButton1Click(Sender: TObject);
procedure SpeedButton10Click(Sender: TObject);
procedure TrackBar3Change(Sender: TObject);
procedure SpeedButton32Click(Sender: TObject);
procedure SpeedButton25Click(Sender: TObject);
procedure SpeedButton26Click(Sender: TObject);
procedure SpeedButton27Click(Sender: TObject);
```

```
    procedure SpeedButton13Click(Sender: TObject);
    procedure SpeedButton22Click(Sender: TObject);
    procedure SpeedButton30Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure Button4Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;
  data1, data2: array[0..5000] of longint;
  s:string;

implementation

{$R *.DFM}
{DLL procedures and functions}
PROCEDURE Voltage1(VpDiv:Longint);stdcall; external 'PCSU1000D.dll';
PROCEDURE Voltage2(VpDiv:Longint);stdcall; external 'PCSU1000D.dll';
PROCEDURE Time(TpDiv:Longint);stdcall; external 'PCSU1000D.dll';
PROCEDURE RunOn(Run: Boolean);stdcall; external 'PCSU1000D.dll';
PROCEDURE SingleOn(Single: Boolean);stdcall; external 'PCSU1000D.dll';
PROCEDURE YPosition1(y_pos:Longint);stdcall; external 'PCSU1000D.dll';
PROCEDURE YPosition2(y_pos:Longint);stdcall; external 'PCSU1000D.dll';
PROCEDURE TrgOn(trg_on: Boolean);stdcall; external 'PCSU1000D.dll';
PROCEDURE TrgLevel(TrgLevel:Longint);stdcall; external 'PCSU1000D.dll';
PROCEDURE TrgSource(CH1_CH2_Ext:Longint);stdcall; external 'PCSU1000D.dll';
PROCEDURE TrgEdge(Positive_Negative:Longint);stdcall; external 'PCSU1000D.dll';
PROCEDURE Coupling1(AC_DC_GND:Longint);stdcall; external 'PCSU1000D.dll';
PROCEDURE Coupling2(AC_DC_GND:Longint);stdcall; external 'PCSU1000D.dll';
procedure Show_PCSU1000(Visible: Boolean); stdcall; external 'PCSU1000D.dll';
procedure Start_PCSU1000; stdcall; external 'PCSU1000D.dll';
procedure Stop_PCSU1000; stdcall; external 'PCSU1000D.dll';
function GetSettings(SettingsArray: Pointer): Boolean ; stdcall; external
'PCSU1000D.dll';

procedure ReadCh1(Buffer: Pointer); stdcall; external 'DSOLink.dll';
procedure ReadCh2(Buffer: Pointer); stdcall; external 'DSOLink.dll';
function DataReady:Boolean; stdcall; external 'DSOLink.dll';

procedure TForm1.DataReadClick(Sender: TObject);
var i: longint;
p:pointer;
begin
  p:= @data1[0];
  ReadCh1(p);
  p:= @data2[0];
  ReadCh2(p);
  memo1.clear;
  memo1.lines.add('Sample rate
[Hz]'+chr(9)+inttostr(data1[0])+chr(9)+inttostr(data2[0]));
  memo1.lines.add('Full scale
[mV]'+chr(9)+inttostr(data1[1])+chr(9)+inttostr(data2[1]));
  memo1.lines.add('GND level
[counts]'+chr(9)+inttostr(data1[2])+chr(9)+inttostr(data2[2]));
  memo1.lines.add('');
  begin
    for i:=0 to 8 do
    memo1.lines.add('Data
('+inttostr(i)+')'+chr(9)+chr(9)+inttostr(data1[i+3])+chr(9)+inttostr(data2[i+3]));
  end;
end;

procedure TForm1.RunClick(Sender: TObject);
begin
   RunOn(Run.Down)
end;

procedure TForm1.SingleClick(Sender: TObject);
begin
  SingleOn(true);
```

```
end;

procedure TForm1.ShowPCS100Click(Sender: TObject);
begin
  Show_PCSU1000(true);
end;

procedure TForm1.HidePCS100Click(Sender: TObject);
begin
   Show_PCSU1000(false);
end;

procedure TForm1.TrackBar1Change(Sender: TObject);
begin
  YPosition1(TrackBar1.position);
end;

procedure TForm1.TrackBar2Change(Sender: TObject);
begin
  YPosition2(TrackBar2.position);
end;

procedure TForm1.SpeedButton1Click(Sender: TObject);
begin
  Voltage1((sender as TSpeedButton).tag);
end;

procedure TForm1.SpeedButton13Click(Sender: TObject);
begin
  Voltage2((sender as TSpeedButton).tag);
end;

procedure TForm1.SpeedButton10Click(Sender: TObject);
begin
  Coupling1((sender as TSpeedButton).tag);
end;

procedure TForm1.SpeedButton22Click(Sender: TObject);
begin
  Coupling2((sender as TSpeedButton).tag);
end;

procedure TForm1.TrackBar3Change(Sender: TObject);
begin
  TrgLevel(TrackBar3.position);
end;

procedure TForm1.SpeedButton32Click(Sender: TObject);
begin
  Time((sender as TSpeedButton).tag);
end;

procedure TForm1.SpeedButton25Click(Sender: TObject);
begin
  TrgOn(true);
end;

procedure TForm1.SpeedButton26Click(Sender: TObject);
begin
  TrgOn(false);
end;

procedure TForm1.SpeedButton27Click(Sender: TObject);
begin
  TrgSource((sender as TSpeedButton).tag);
end;

procedure TForm1.SpeedButton30Click(Sender: TObject);
begin
  TrgEdge((sender as TSpeedButton).tag);
end;

procedure TForm1.Button3Click(Sender: TObject);
begin
  if DataReady then  label1.caption:='Yes' else label1.caption:='No'
end;
```

```
procedure TForm1.Button1Click(Sender: TObject);
begin
  Start_PCSU1000;
end;

procedure TForm1.Button2Click(Sender: TObject);
begin
  Stop_PCSU1000;
end;

procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);
begin
  Stop_PCSU1000;
end;

procedure TForm1.Button4Click(Sender: TObject);
var SettingsArray:array [0..10] of longint;
begin
  if GetSettings(@SettingsArray) then
  begin
    case SettingsArray[0] of
      0: SpeedButton1.Down:=true;
      1: SpeedButton2.Down:=true;
      2: SpeedButton3.Down:=true;
      3: SpeedButton4.Down:=true;
      4: SpeedButton5.Down:=true;
      5: SpeedButton6.Down:=true;
      6: SpeedButton7.Down:=true;
      7: SpeedButton8.Down:=true;
      8: SpeedButton9.Down:=true;
    end;
    case SettingsArray[1] of
      0: SpeedButton13.Down:=true;
      1: SpeedButton14.Down:=true;
      2: SpeedButton15.Down:=true;
      3: SpeedButton18.Down:=true;
      4: SpeedButton17.Down:=true;
      5: SpeedButton16.Down:=true;
      6: SpeedButton19.Down:=true;
      7: SpeedButton20.Down:=true;
      8: SpeedButton21.Down:=true;
    end;
    case SettingsArray[2] of
      0: SpeedButton32.Down:=true;
      1: SpeedButton33.Down:=true;
      2: SpeedButton34.Down:=true;
      3: SpeedButton35.Down:=true;
      4: SpeedButton36.Down:=true;
      5: SpeedButton37.Down:=true;
      6: SpeedButton38.Down:=true;
      7: SpeedButton39.Down:=true;
      8: SpeedButton40.Down:=true;
      9: SpeedButton41.Down:=true;
      10: SpeedButton42.Down:=true;
      11: SpeedButton43.Down:=true;
      12: SpeedButton44.Down:=true;
      13: SpeedButton45.Down:=true;
      14: SpeedButton46.Down:=true;
      15: SpeedButton47.Down:=true;
      16: SpeedButton48.Down:=true;
      17: SpeedButton49.Down:=true;
      18: SpeedButton50.Down:=true;
    end;
    TrackBar1.position:=SettingsArray[3];
    TrackBar2.position:=SettingsArray[4];

    case SettingsArray[5] of
      0: SpeedButton10.Down:=true;
      1: SpeedButton11.Down:=true;
      2: SpeedButton12.Down:=true;
    end;
    case SettingsArray[6] of
      0: SpeedButton22.Down:=true;
      1: SpeedButton23.Down:=true;
      2: SpeedButton24.Down:=true;
    end;
    case SettingsArray[7] of
```

```
      0: SpeedButton26.Down:=true;
      1: SpeedButton25.Down:=true;
    end;
    case SettingsArray[8] of
      0: SpeedButton27.Down:=true;
      1: SpeedButton28.Down:=true;
      2: SpeedButton29.Down:=true;
    end;
    case SettingsArray[9] of
      0: SpeedButton31.Down:=true;
      1: SpeedButton30.Down:=true;
    end;
    TrackBar3.position:=SettingsArray[10];
  end;
end;

end.
```

Using the PCSU1000D.DLL in Visual Basic

In this application example there are the declarations of the *PCSU1000D.DLL* and *DSOLINK.DLL* procedures and functions and an example how to use these function calls.

Note: Make sure that the files *PCSU1000D.DLL* and *DSOLINK.DLL* are copied to the Windows' *SYSTEM32* folder:

```
Option Explicit
Dim DataBuffer1(0 To 5000) As Long
Dim DataBuffer2(0 To 5000) As Long
Dim SettingsArray(0 To 10) As Long
Private Declare Sub ReadCh1 Lib "DSOLink.dll" (Buffer As Long)
Private Declare Sub ReadCh2 Lib "DSOLink.dll" (Buffer As Long)
Private Declare Function DataReady Lib "DSOLink.dll" () As Boolean
Private Declare Sub Voltage1 Lib "PCSU1000D.dll" (ByVal Volts As Long)
Private Declare Sub Voltage2 Lib "PCSU1000D.dll" (ByVal Volts As Long)
Private Declare Sub Time Lib "PCSU1000D.dll" (ByVal Rate As Long)
Private Declare Sub RunOn Lib "PCSU1000D.dll" (ByVal Run_On As Long)
Private Declare Sub SingleOn Lib "PCSU1000D.dll" (ByVal Single_On As Long)
Private Declare Sub YPosition1 Lib "PCSU1000D.dll" (ByVal y_pos As Long)
Private Declare Sub YPosition2 Lib "PCSU1000D.dll" (ByVal y_pos As Long)
Private Declare Sub TrgOn Lib "PCSU1000D.dll" (ByVal trg_On As Long)
Private Declare Sub TrgLevel Lib "PCSU1000D.dll" (ByVal TrgLevel As Long)
Private Declare Sub TrgSource Lib "PCSU1000D.dll" (ByVal TrgLevel As Long)
Private Declare Sub TrgEdge Lib "PCSU1000D.dll" (ByVal Positive_Negative As Long)
Private Declare Sub Coupling1 Lib "PCSU1000D.dll" (ByVal AC_DC_GND As Long)
Private Declare Sub Coupling2 Lib "PCSU1000D.dll" (ByVal AC_DC_GND As Long)
Private Declare Sub Show_PCSU1000 Lib "PCSU1000D.dll" (ByVal Visible As Long)
Private Declare Sub Start_PCSU1000 Lib "PCSU1000D.dll" ()
Private Declare Sub Stop_PCSU1000 Lib "PCSU1000D.dll" ()
Private Declare Function GetSettings Lib "PCSU1000D.dll" (SettingsArray As Long) As
Boolean

Private Sub Command1_Click()
    Start_PCSU1000
End Sub

Private Sub Command2_Click()
    Stop_PCSU1000
End Sub

Private Sub Command10_Click()
    Show_PCSU1000 1
End Sub

Private Sub Command11_Click()
    Show_PCSU1000 0
End Sub

Private Sub Check1_Click()
    RunOn Check1.Value
End Sub
Private Sub Command13_Click()
```

```
    SingleOn 1
End Sub

Private Sub Command3_Click()
    If DataReady Then Label10.Caption = "Yes" Else Label10.Caption = "No"
End Sub

Private Sub Command4_Click()
    If GetSettings(SettingsArray(0)) Then
        Option1(SettingsArray(0)).Value = True
        Option2(SettingsArray(1)).Value = True
        Option5(SettingsArray(2)).Value = True
        VScroll1.Value = SettingsArray(3)
        VScroll2.Value = SettingsArray(4)
        Option3(SettingsArray(5)).Value = True
        Option4(SettingsArray(6)).Value = True
        Option6(SettingsArray(7)).Value = True
        Option7(SettingsArray(8)).Value = True
        Option8(SettingsArray(9)).Value = True
        VScroll3.Value = SettingsArray(10)
    End If
End Sub

Private Sub Command9_Click()
    Dim i As Long
    List1.Clear
    ReadCh1 DataBuffer1(0)
    ReadCh2 DataBuffer2(0)
    List1.AddItem "Sample rate [Hz]" + Chr(9) + Str(DataBuffer1(0)) + Chr(9) +
Str(DataBuffer2(0))
    List1.AddItem "Full scale [mV]" + Chr(9) + Str(DataBuffer1(1)) + Chr(9) +
Str(DataBuffer2(1))
    List1.AddItem "GND level [counts]" + Chr(9) + Str(DataBuffer1(2)) + Chr(9) +
Str(DataBuffer2(2))
    List1.AddItem ""
    For i = 0 To 8
      List1.AddItem "Data(" + Str(i) + ")" + Chr(9) + Chr(9) + Str(DataBuffer1(i + 3))
+ Chr(9) + Str(DataBuffer2(i + 3))
    Next
End Sub

Private Sub Form_Terminate()
    Stop_PCSU1000
End Sub

Private Sub Option1_Click(Index As Integer)
    Voltage1 Index
End Sub

Private Sub Option2_Click(Index As Integer)
    Voltage2 Index
End Sub

Private Sub Option3_Click(Index As Integer)
    Coupling1 Index
End Sub

Private Sub Option4_Click(Index As Integer)
    Coupling2 Index
End Sub

Private Sub Option5_Click(Index As Integer)
    Time Index
End Sub

Private Sub Option6_Click(Index As Integer)
    TrgOn Index
End Sub

Private Sub Option7_Click(Index As Integer)
  TrgSource Index
End Sub

Private Sub Option8_Click(Index As Integer)
    TrgEdge Index
End Sub
```

```
Private Sub VScroll1_Change()
    YPosition1 VScroll1.Value
End Sub

Private Sub VScroll2_Change()
    YPosition2 VScroll2.Value
End Sub

Private Sub VScroll3_Change()
    TrgLevel VScroll3.Value
End Sub
```