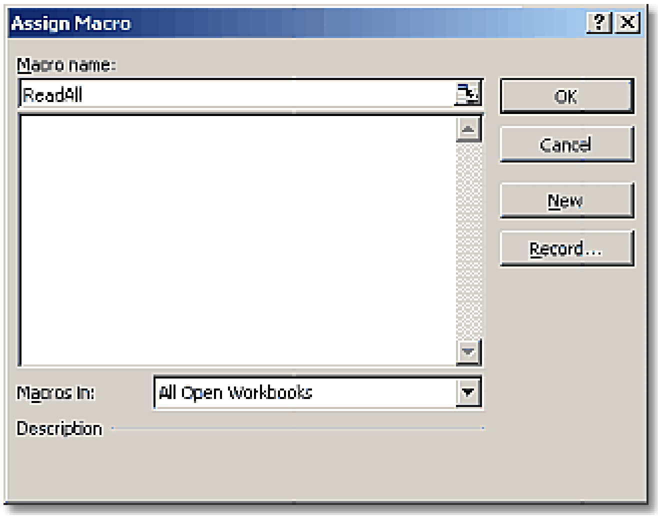
Direct Link from the Velleman PC Oscilloscopes to Microsoft Excel

Pc-Lab2000SE software package includes a DLL (Dynamic Link Library) *DSOLink.DLL,* installed to Windows' *SYSTEM32* folder.
This DLL allows you to write custom applications in Excel, Visual Basic, Delphi or any other 32-bit Windows application development tool that supports calls to a DLL.

Transferring Waveform Data to an Excel Spreadsheet
The example Excel macro shows you how to collect data directly into the spreadsheet from the Velleman PC oscilloscopes without the need for other software.

1. Open **Microsoft Excel** and start a new workbook document.
2. Select the **View / Toolbars** Menu and Select **Forms.**
   - The Forms toolbar appears.
3. Create a **Button**
   - In the Forms toolbar click on the "Button" button: the mouse-pointer will become a small cross.
   - On Excel worksheet, use the mouse to draw a rectangle to mark where you want your button to appear.
   - When you release the mouse after drawing the rectangle the "Assign Macro" dialog-box will appear.



Macro name: ReadAll

Macros in: All Open Workbooks

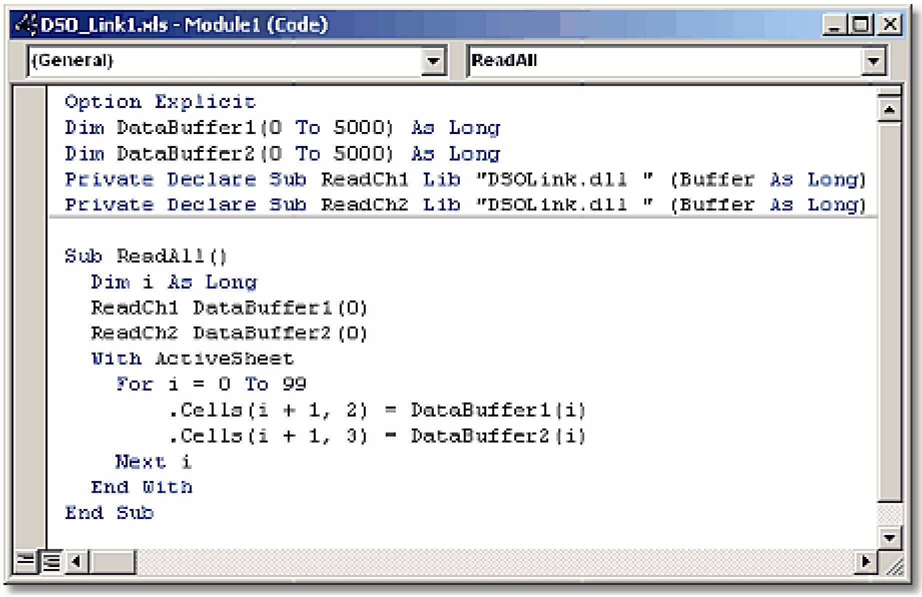
4. Type Macro name: ***ReadAll*** and click **New** button.
   - Microsoft Visual Basic edit window will open. A subroutine called *ReadAll* has been created.
5. Use Copy and Paste to replace the default text:

```
Sub ReadAll()
End Sub
```

with the following text in the edit window:

```
Option Explicit
Dim DataBuffer1(0 To 5000) As Long
Dim DataBuffer2(0 To 5000) As Long
Private Declare Sub ReadCh1 Lib "DSOLink.dll " (Buffer As Long)
Private Declare Sub ReadCh2 Lib "DSOLink.dll " (Buffer As Long)
```

```
Sub ReadAll()
  Dim i As Long
  ReadCh1 DataBuffer1(0)
  ReadCh2 DataBuffer2(0)
  With ActiveSheet
    For i = 0 To 99
        .Cells(i + 1, 2) = DataBuffer1(i)
        .Cells(i + 1, 3) = DataBuffer2(i)
    Next i
  End With
End Sub
```



```
Option Explicit
Dim DataBuffer1(0 To 5000) As Long
Dim DataBuffer2(0 To 5000) As Long
Private Declare Sub ReadCh1 Lib "DSOLink.dll " (Buffer As Long)
Private Declare Sub ReadCh2 Lib "DSOLink.dll " (Buffer As Long)

Sub ReadAll()
  Dim i As Long
  ReadCh1 DataBuffer1(0)
  ReadCh2 DataBuffer2(0)
  With ActiveSheet
    For i = 0 To 99
        .Cells(i + 1, 2) = DataBuffer1(i)
        .Cells(i + 1, 3) = DataBuffer2(i)
    Next i
  End With
End Sub
```

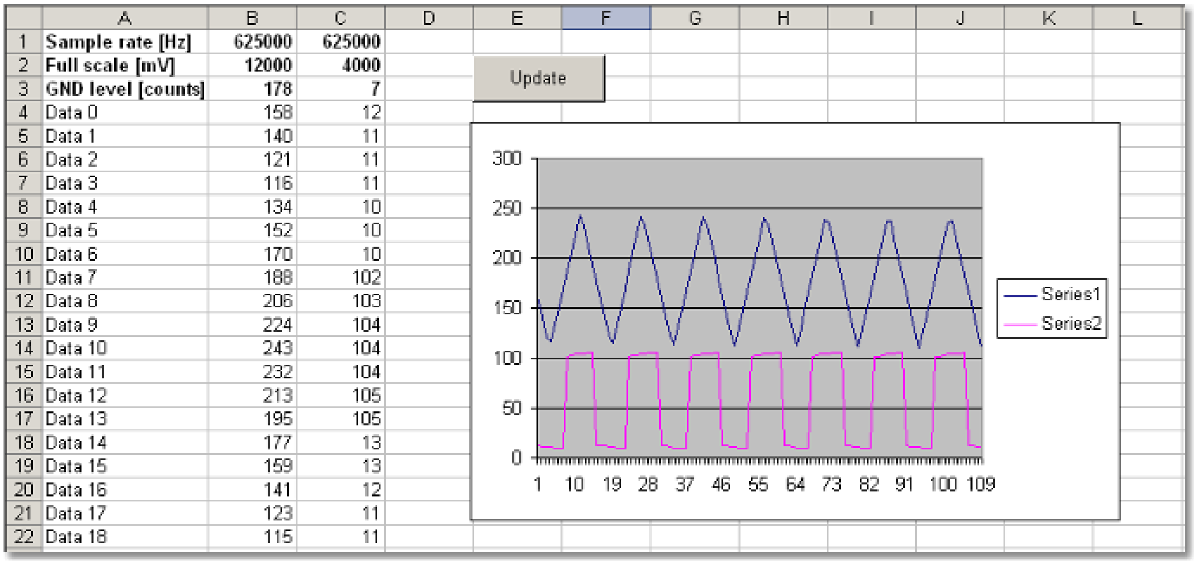
6. Press **Alt+F11** to return to Excel.
7. Type following texts to column **A**:

Sample rate [Hz]
Full scale [mV]
GND level [counts]
Data 0
Data 1
Data 2
...

8. Start oscilloscope program for PCS500, PCS100 or K8031 and click **Run** or **Single** button.
9. Click the button on the Excel worksheet. The created macro will execute and the data described in column **A** will appear to the worksheet columns **B** and **C**.
   - Rows 4...4099 contain the acquired data in A/D converter counts (0...255) for PCS500.
   - Rows 4...4083 contain the acquired data in A/D converter counts (0...255) for PCS100 and K8031.
   - The triggering point of PCS500 is on row 1030 and of PCS100 and K8031 on row 4.

The three first rows contain the scope settings and the rest of the rows contain the raw oscilloscope data in A/D converter counts (0...255).
Using the ***Sample rate***, ***Full scale*** and ***GND level*** values it is possible to reconstruct the

waveform data into engineering units (e.g. Volts and seconds) for further analysis.



	A	B	C
1	**Sample rate [Hz]**	**625000**	**625000**
2	**Full scale [mV]**	**12000**	**4000**
3	**GND level [counts]**	**178**	**7**
4	Data 0	158	12
5	Data 1	140	11
6	Data 2	121	11
7	Data 3	116	11
8	Data 4	134	10
9	Data 5	152	10
10	Data 6	170	10
11	Data 7	188	102
12	Data 8	206	103
13	Data 9	224	104
14	Data 10	243	104
15	Data 11	232	104
16	Data 12	213	105
17	Data 13	195	105
18	Data 14	177	13
19	Data 15	159	13
20	Data 16	141	12
21	Data 17	123	11
22	Data 18	115	11

Data acquisition to other applications

The DSOLink.DLL gives you direct access to real-time data and settings information from the oscilloscope to custom applications in Visual Basic, Delphi, Borland C++ Builder or any other 32-bit Windows application development tool that supports calls to a DLL..

The complete example programs are located on the Velleman CD. Those may be used as a starting point how to construct your customized application programs.

Note: Before running the following example programs: The oscilloscope software must be running and "Run" or "Single" button pressed and trace displayed on the oscilloscope screen.

Description of the procedures of the DSOLink.DLL

ReadCh1
ReadCh2

Syntax
```
PROCEDURE ReadCh1(Buffer: Pointer);
PROCEDURE ReadCh2(Buffer: Pointer);
```

Parameter
`Buffer`: A pointer to the data array of 5000 long integers where the data will be read.

Description
Read all the data and the settings of channel 1 or channel 2 of the PCS500, PCS100 or K8031.
As a return the following data is put to the buffer:
[0] : Sample rate in Hz
[1] : Full scale voltage in mV
[2] : Ground level in A/D converter counts. The value may be beyond the 0...255 range if GND level is adjusted beyond the waveform display area.
[3...4098] : The acquired data in A/D converter counts (0...255), from PCS500.
[3...4082] : The acquired data in A/D converter counts (0...255), from PCS100 and K8031.

The triggering point of the PCS500 is at the data location [1027] and of the PCS100 and K8031at the location [3].

DataReady

Syntax
```
FUNCTION DataReady : Boolean;
```

Description
Indicates if there is fresh data available. This function can be used to check that triggering has occurred.

Result
`Boolean`: TRUE means that there is new waveform data available from the oscilloscope.

Example

```
    if DataReady then  label1.caption:='Yes' else label1.caption:='No'
```

Running the DSOLink in Delphi

Check the ***\PC-Lab2000SE tools\PCS500 - PCS100 - K8031\Data transfer DSOLink_DLL\DSOLink_Demo_VB\*** folder on the Velleman CD to locate the demo files.
This folder contains a ready to run *DSOLink_Demo.EXE* program and its source code.
You may copy the files to any folder and use Delphi to examine, edit and compile the files.

Example (in Delphi)

```
var
  data: array[0..5000] of longint;

procedure ReadCh1(Buffer: Pointer); stdcall; external 'DSOLink.dll';

procedure TForm1.Button1Click(Sender: TObject);
var i: longint;
p:pointer;
begin
  p:= @data[0];
  ReadCh1(p);
  memo1.clear;
  memo1.lines.add('Sample rate [Hz]'+chr(9)+inttostr(data[0]));
  memo1.lines.add('Full scale [mV]'+chr(9)+inttostr(data[1]));
  memo1.lines.add('GND level [counts]'+chr(9)+inttostr(data[2]));
  memo1.lines.add('');
  begin
    for i:=0 to 20 do
    memo1.lines.add( 'Data ('+inttostr(i)+ ')'+chr(9)+chr(9)+inttostr(data[i+3]));
  end;
end;
```

Running the DSOLink in Visual Basic

Make sure that the file *DSOLink.DLL* is copied to Windows' *SYSTEM32* folder.
Check the ***\PC-Lab2000SE tools\PCS500 - PCS100 - K8031\Data transfer DSOLink_DLL\DSOLink_Demo_VB\*** folder on the VELSOFT CD to locate the demo files.
This folder contains a ready to run *DSOLink_Demo.EXE* program and its source code.
You may copy the files to any folder and use Visual Basic to examine, edit and compile the files.

Example (in Visual Basic)

```
Option Explicit

Dim DataBuffer(0 To 5000) As Long
Private Declare Sub ReadCh1 Lib "DSOLink.dll " (Buffer As Long)

'This reads the settingsd and 4096 bytes of data from CH1 to the data buffer.
'The first 21 values are displayed.

Private Sub Read_CH1_Click(Index As Integer)
    Dim i As Long
    List1.Clear
    ReadCh1 DataBuffer(0)
    List1.AddItem "Sample rate [Hz]" + Chr(9) + Str(DataBuffer(0))
    List1.AddItem "Full scale [mV]" + Chr(9) + Str(DataBuffer(1))
    List1.AddItem "GND level [counts]" + Chr(9) + Str(DataBuffer(2))
    List1.AddItem ""
    For i = 0 To 20
      List1.AddItem "Data(" + Str(i) + ")" + Chr(9) + Chr(9) + Str(DataBuffer(i + 3))
    Next
End Sub
```

Running the DSOLink in Borland C++ Builder

The following files are available in the ***\PC-Lab2000SE tools\PCS500 - PCS100 - K8031\Data transfer DSOLink_DLL\DSOLink_Demo_BCB\*** folder on the VELSOFT CD for development with Borland C++Builder:

DSOLink.dll the dynamically linked library
DSOLink.h the C/C++ header file for function prototypes
DSOLink.lib the import library
DSOLink_demo.cpp demo source

1. Create a new project in Borland C++ Builder.
2. Add the import library to your project using **Project | Add to Project** menu option.
3. Add a `#include` statement in the main unit that includes *DSOLink.H*.
4. Finally, add code that calls the DLL functions.

DSOLink.h

```
//---------------------------------------------------------------------------
// DSOLink.h
#ifdef __cplusplus
extern "C" {           /* Assume C declarations for C++ */

#endif

#define FUNCTION __declspec(dllimport)

FUNCTION __stdcall ReadCh1(int* ptr);
FUNCTION __stdcall ReadCh2(int* ptr);

#ifdef __cplusplus
}
#endif
//---------------------------------------------------------------------------
```

Example (in Borland C++Builder)

```
//---------------------------------------------------------------------------
// DSOLink_demo.cpp

#include <vcl.h>
#pragma hdrstop
#include "DSOLink.h"
#include "DSOLink_demo.h"
//---------------------------------------------------------------------------
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
//---------------------------------------------------------------------------
__fastcall TForm1::TForm1(TComponent* Owner)
        : TForm(Owner)
{
}
//---------------------------------------------------------------------------
void __fastcall TForm1::Button1Click(TObject *Sender)
{
        int data[5000];
        ReadCh1(data);

        Memo1->Clear();
        Memo1->Lines->Add("Sample rate [Hz]:   "+IntToStr(data[0]));
        Memo1->Lines->Add("Full scale [mV]:    "+IntToStr(data[1]));
        Memo1->Lines->Add("GND level [counts]: "+IntToStr(data[2]));
        Memo1->Lines->Add("");

        for (int i = 0; i < 20; i++)
        {
               Memo1->Lines->Add("Data "+IntToStr(i)+char(9)+IntToStr(data[i+3]));
        }
}
//---------------------------------------------------------------------------
```

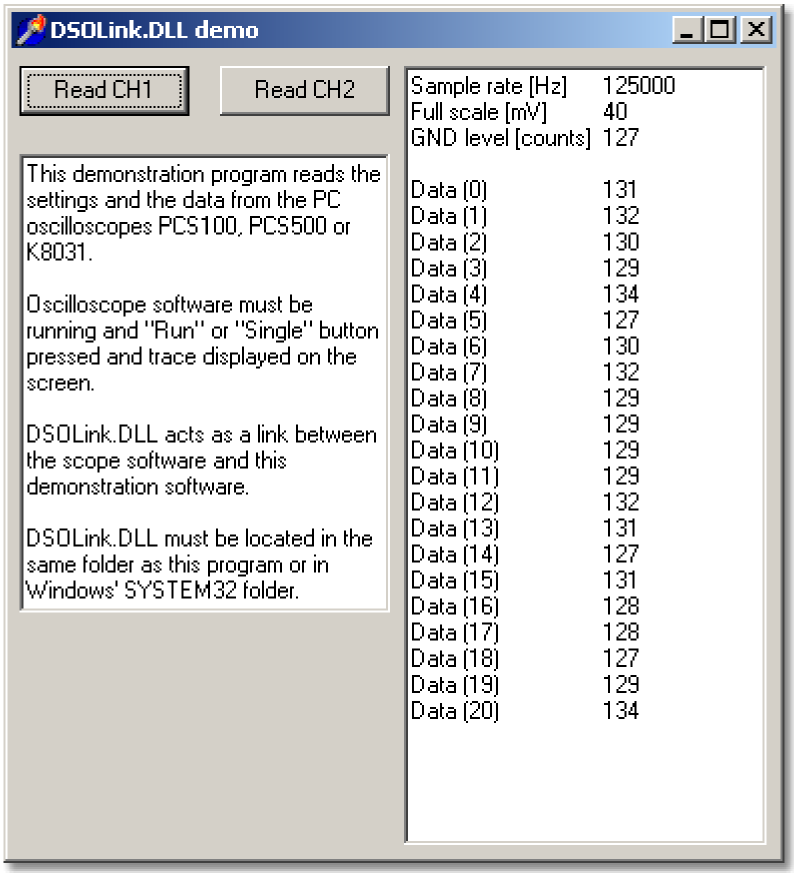
Note: If the import library is not compatible to your Borland C++ version, you can create an import library by running IMPLIB on the DLL.

IMPLIB works like this:

```
  IMPLIB (destination lib name) (source dll)
```

For example,

```
  IMPLIB DSOLink.lib DSOLink.dll
```



This demonstration program reads the settings and the data from the PC oscilloscopes PCS100, PCS500 or K8031.

Oscilloscope software must be running and "Run" or "Single" button pressed and trace displayed on the screen.

DSOLink.DLL acts as a link between the scope software and this demonstration software.

DSOLink.DLL must be located in the same folder as this program or in Windows' SYSTEM32 folder.

```
Sample rate [Hz]  125000
Full scale [mV]   40
GND level [counts] 127

Data (0)          131
Data (1)          132
Data (2)          130
Data (3)          129
Data (4)          134
Data (5)          127
Data (6)          130
Data (7)          132
Data (8)          129
Data (9)          129
Data (10)         129
Data (11)         129
Data (12)         132
Data (13)         131
Data (14)         127
Data (15)         131
Data (16)         128
Data (17)         128
Data (18)         127
Data (19)         129
Data (20)         134
```