Description of the DLL for the Recorder/Logger K8047 & PCS10

All the communication routines of the Recorder/Logger are contained in a Dynamic Link Library (DLL) K8047D.DLL. This document describes the DLL procedures that are available for your application program. Using the DLL allows you to write custom Windows 98, 2000, ME or Windows XP based applications in Visual Basic, Delphi or any other 32-bit Windows application development tool that supports calls to a DLL.

A complete overview of all the procedures that are used by the "K8047D.DLL" follows. The example program can also be carefully studied in order to gain an insight as to how to construct your customized application programs.
Note that all examples in this description are written for Delphi.

At the end of this document there are full declarations for the DLL function and procedures for Delphi and Visual Basic.

Running the Recorder/Logger DLL Demo under Visual Basic

Unzip the file **VB_Recorder_Logger_Setup.ZIP** and run the **SETUP.EXE**.
This installs the files FASTTIME32.DLL, K8047D.DLL and K8047E.EXE to Windows' SYSTEM32 folder.

The file **VB_Recorder_Logger_Demo.ZIP** contains the source code for the Recorder/Logger DLL Demo. You may unzip it to any folder and use Visual Basic to examine, edit and compile the files.

Running the Recorder/Logger DLL Demo under Delphi

The file **Delphi_Recorder_Logger_Demo.ZIP** contains ready to run **Recorder_Logger_Demo.EXE** and its source code.
You may unzip it to any folder and use Delphi to examine, edit and compile the files.

Please note:
The files **FASTTIME32.DLL**, **K8047D.DLL** and **K8047E.EXE** must be located in the program file folder or in the Windows' SYSTEM32 folder.

Procedures of the K8047D.DLL

General procedures

`StartDevice`	*Opens a link to the Recorder/Logger device*
`StopDevice`	*Closes the link to the device*

Input procedure

`ReadData(Buffer)`	*Reads the timer counter status and the A/D data from the K8047 to a buffer in the application program*

Output procedures

`SetGain(Channel, Gain)`	*Sets the input amplifier gain of selected channel 1...4. Gain values 1, 2, 5 and 10 are the only valid values.*
`LEDon`	*Turns the Record LED on*
`LEDoff`	*Turns the Record LED off*

Description of the procedures

StartDevice

Syntax
```
PROCEDURE StartDevice;
```

Description
Initializes the communication routines for the K8047 unit. Loads the drivers needed to communicate via the USB port. **This procedure must be performed in the beginning of the application program.**

Example
```
  BEGIN
     StartDevice;
  END;
```

SopDevice

Syntax
```
PROCEDURE StopDevice;
```

Description
Unloads the communication routines for K8047 unit and unloads the drivers needed to communicate via the USB port. **This is the last action of the application program before termination.**

Example
```
  BEGIN
     StopDevice;
  END;
```

ReadData

Syntax

```
PROCEDURE ReadData(Buffer:Pointer);
```

Parameter

`Buffer`: A pointer to the data array of 8 integers where the data will be read.

Description

Reads the timer counter status and the A/D data from the K8047 to a buffer in the application program. The timer counter is incremented every 10ms. The new data from the A/D converter channels 1...4 is updated every time the timer counter is incremented.
Data structure:

Timer data		**A/D converter data**				**Reserved**	
LSB	MSB	CH1	CH2	CH3	CH4	0	0

Example

```
var  // global variables
  DataBuffer: ARRAY[0..7] OF Integer;

procedure TForm1.Button1Click(Sender: TObject);
var p:pointer;
i:integer;
s:string;
begin
  p:=@DataBuffer;  // Address of the data buffer
  ReadData(p);     // Read the data from K8047
  memo1.clear;
  s:='';
  for i:=0 to 7 do s:=s +inttostr(DataBuffer[i]+chr(9);
  memo1.lines.add(s); // Display the data
end;
```

SetGain

Syntax

```
PROCEDURE SetGain(Channel: Longint ; Gain: Longint);
```

Parameter

`Channel no`: Value between 1 and 4 which corresponds to the input channel number whose gain data is to be changed.
`Data`: The input amplifier gain value. **Valid values are only: 1, 2, 5 and 10**.

Description

Selects one of the input amplifier hardware pre-selected gain settings. The corresponding full scale (A/D output 255) volteges are following:

Gain	**Full scale input voltage**
1	30
2	15
5	6
10	3

Example

```
  BEGIN
    SetGain(3,5);   // CH 3 gain is set to 5
  END;
```

LEDon, LEDoff

Syntax

```
PROCEDURE LEDon;
PROCEDURE LEDoff;
```

Description

`LEDon` turns the Record LED on.

`LEDoff` turns the Record LED off.

Example

```
  BEGIN
    LEDon; // Record LED on
  END;
```

Using the DLL in Delphi

Here is a sample program including the declaration of the K8047D.DLL procedures and the array variables. The two most important procedures of the application program are `FormCreate` and `FormClose`.

```
unit K8047;

interface

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

type
  TForm1 = class(TForm)
    Button1: TButton;
    CheckBox1: TCheckBox;
    GroupBox1: TGroupBox;
    RadioGroup1: TRadioGroup;
    RadioGroup2: TRadioGroup;
    RadioGroup3: TRadioGroup;
    RadioGroup4: TRadioGroup;
    Memo1: TMemo;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    procedure FormCreate(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure Button1Click(Sender: TObject);
    procedure CheckBox1Click(Sender: TObject);
    procedure RadioGroup1Click(Sender: TObject);
    procedure RadioGroup2Click(Sender: TObject);
    procedure RadioGroup3Click(Sender: TObject);
    procedure RadioGroup4Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;
  DataBuffer: ARRAY[0..7] OF LongInt;

implementation

{$R *.DFM}
PROCEDURE StartDevice; stdcall; external 'K8047d.dll';
PROCEDURE StopDevice; stdcall; external 'K8047d.dll';
PROCEDURE ReadData(Buffer:Pointer); stdcall; external 'K8047d.dll';
PROCEDURE SetGain(Channel: Longint ; Gain: Longint); stdcall; external 'K8047d.dll';
PROCEDURE LEDon; stdcall; external 'K8047d.dll';
PROCEDURE LEDoff; stdcall; external 'K8047d.dll';

procedure TForm1.FormCreate(Sender: TObject);
begin
  StartDevice;
end;

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

procedure TForm1.Button1Click(Sender: TObject);
```

```
var p:pointer;
i:integer;
s:string;
begin
  p:=@DataBuffer;
  ReadData(p);
  s:='';
  for i:=0 to 5 do s:=s +inttostr(DataBuffer[i])+chr(9);
  memo1.lines.add(s);
end;

procedure TForm1.CheckBox1Click(Sender: TObject);
begin
  if CheckBox1.checked then LEDon else LEDoff;
end;

procedure TForm1.RadioGroup1Click(Sender: TObject);
begin
  case RadioGroup1.itemindex of
    0: SetGain(1,1);
    1: SetGain(1,2);
    2: SetGain(1,5);
    3: SetGain(1,10);
  end;
end;

procedure TForm1.RadioGroup2Click(Sender: TObject);
begin
  case RadioGroup2.itemindex of
    0: SetGain(2,1);
    1: SetGain(2,2);
    2: SetGain(2,5);
    3: SetGain(2,10);
  end;
end;

procedure TForm1.RadioGroup3Click(Sender: TObject);
begin
  case RadioGroup3.itemindex of
    0: SetGain(3,1);
    1: SetGain(3,2);
    2: SetGain(3,5);
    3: SetGain(3,10);
  end;
end;

procedure TForm1.RadioGroup4Click(Sender: TObject);
begin
  case RadioGroup4.itemindex of
    0: SetGain(4,1);
    1: SetGain(4,2);
    2: SetGain(4,5);
    3: SetGain(4,10);
  end;
end;

end.
```

Using the DLL in Visual Basic

Here is the declaration of the K8047D.DLL procedures and array variables. The two most important procedures of the application program are `Form_Load()` and `Form_Terminate()`.

```
Option Explicit

'Declare use of the DLL
'K8047D.DLL interface

'GENERAL PROCEDURES
Private Declare Sub StartDevice Lib "k8047d.dll" ()
Private Declare Sub StopDevice Lib "k8047d.dll" ()
Private Declare Sub LEDon Lib "k8047d.dll" ()
Private Declare Sub LEDoff Lib "k8047d.dll" ()

'INPUT PROCEDURE
Private Declare Sub ReadData Lib "k8047d.dll" (Array_Pointer  As Long)

'OUTPUT PROCEDURE
Private Declare Sub SetGain Lib "k8047d.dll" (ByVal Channel_no As Long, ByVal Gain As Long)

'Declare variables
Dim DataBuffer(0 To 7) As Long

Private Sub Check1_Click()
If Check1.Value = 1 Then LEDon Else LEDoff
End Sub

Private Sub Form_Load()
    StartDevice
End Sub

Private Sub Form_Terminate()
    StopDevice
End Sub
Private Sub Command1_Click()
    Dim i As Integer
    Dim s As String
    ReadData DataBuffer(0)
    s = ""
    For i = 0 To 5
        s = s + Str(DataBuffer(i)) + Chr(9)
    Next i
    List1.AddItem s
End Sub

Private Sub Option1_Click(Index As Integer)
    SetGain 1, Index
End Sub

Private Sub Option2_Click(Index As Integer)
    SetGain 2, Index
End Sub

Private Sub Option3_Click(Index As Integer)
    SetGain 3, Index
End Sub

Private Sub Option4_Click(Index As Integer)
    SetGain 4, Index
End Sub
```