

```
1 unit Main;
2
3 interface
4
5 uses
6   Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
7   StdCtrls, OleCtrls, OlectraChart2D_TLB, CBW32, Unit2, RChart, Fourier;
8
9 type
10  TForm1 = class(TForm)
11    Button1: TButton;
12    Label1: TLabel;
13    Label2: TLabel;
14    Chart2D1: TChart2D;
15    RChart1: TRChart;
16    ComboBox1: TComboBox;
17    Label3: TLabel;
18    FastFourier1: TFastFourier;
19    CheckBox1: TCheckBox;
20    procedure FormCreate(Sender: TObject);
21    procedure Button1Click(Sender: TObject);
22    procedure ComboBox1Change(Sender: TObject);
23    procedure CheckBox1Click(Sender: TObject);
24  private
25    { Private declarations }
26  public
27    { Public declarations }
28  end;
29
30 var
31  Form1      : TForm1;
32  S          : Sample; // Deklarasjon av tråd
33  Count     : Integer; // Antall samples som skal hentes
34  Rate      : Integer; // Samplerate (Antall AD-samples pr. sekund)
35  GeoWait   : Integer; // Antall ms programmet skal vente så geofonen får stabilisert se
36
37  SampleWait : Integer; // Antall ms programmet skal vente til sampling er ferdig
38  OlectraX   : Variant; // X-array for Olectra-chart
39  OlectraY   : Variant; // Y-array for Olectra-chart
40  MemHandle  : Integer; // Minne-handle for å lagre AD-data
41  AData      : Array[1..20000] of Word; //Array for lagring av AD-data
42  AVolt      : Array[1..20000] of Real; //Array for lagring av AD-spenning
43
44 procedure FFT;
45
46 implementation
47
48 {$R *.DFM}
49
50 procedure TForm1.FormCreate(Sender: TObject);
51 begin
52   //Sette opp objektene plassering på Form
53   Form1.Chart2D1.Width      := Form1.Width-8;
54   Form1.Chart2D1.Height     := Form1.Height DIV 2;
55   Form1.Chart2D1.Top        := 0;
56   Form1.Button1.Left        := (Form1.Width div 2)-(Form1.Button1.Width div 2);
57   Form1.Button1.Top         := Form1.Height - Form1.Button1.Height - 30;
58   Form1.Label2.Top          := Form1.Button1.Top + 20;
59   Form1.Label2.Left         := 20;
60   Form1.Label1.Top          := Form1.Label2.Top;
61   Form1.Label1.Left         := Form1.Label2.Left + Form1.Label2.Width + 10;
62   Form1.RChart1.Width       := Form1.Width - 8;
63   Form1.RChart1.Height      := Form1.Height - Form1.Chart2D1.Height - 100;
64   Form1.RChart1.Top         := Form1.Chart2D1.Height;
65   Form1.Label3.Top          := Form1.Button1.Top + 5;
```

```
64   Form1.Label3.Left      := Form1.Button1.Left + Form1.Button1.Width + 20;
65   Form1.ComboBox1.Top    := Form1.Label3.Top + 17;
66   Form1.ComboBox1.Left   := Form1.Label3.Left;
67   Form1.CheckBox1.Left   := Form1.Label2.Left;
68   Form1.CheckBox1.Top    := Form1.Label2.Top - 20;
69
70 end;
71
72 procedure TForm1.Button1Click(Sender: TObject);
73 begin
74   GeoWait := 1000;
75   Rate    := 16000;
76   Count   := 16000;
77   S.Resume;
78   SampleWait := (Count div Rate)+500;
79 end;
80
81 procedure TForm1.ComboBox1Change(Sender: TObject);
82 begin
83   //Gjør ny FFT dersom man velger en annen vindustype
84   FFT;
85 end;
86
87 procedure FFT;
88 Var
89   CNT : Integer;
90   y : Double;
91 begin
92
93   //Velger om det skal være log. Y-akse
94   If Form1.CheckBox1.Checked then Form1.RChart1.LogScaleY := True
95   else Form1.RChart1.LogScaleY := False;
96   Form1.RChart1.ClearGraf;
97
98   //Sette vindustype
99   If Form1.ComboBox1.Text = 'Rectangle' then
100     Form1.FastFourier1.WeightingWindow := fwRectangle;
101   If Form1.ComboBox1.Text = 'Triangle' then
102     Form1.FastFourier1.WeightingWindow := fwTriangle;
103   If Form1.ComboBox1.Text = 'Gaussian' then
104     Form1.FastFourier1.WeightingWindow := fwGauss;
105   If Form1.ComboBox1.Text = 'Hamming' then
106     Form1.FastFourier1.WeightingWindow := fwHamming;
107   If Form1.ComboBox1.Text = 'Blackman' then
108     Form1.FastFourier1.WeightingWindow := fwBlackman;
109   If Form1.ComboBox1.Text = 'cos2' then
110     Form1.FastFourier1.WeightingWindow := fwCos2;
111
112
113   Form1.FastFourier1.SpectrumSize := 1024; //Setter datamengde for FFT
114   Form1.FastFourier1.ClearReal;           //Tømmer FFT for reelle tall
115   Form1.FastFourier1.ClearImag;          //Tømmer FFT for imaginære tall
116
117   //Putter data inn i FFT
118   For CNT := 1 to 1024 do
119     Form1.FastFourier1.RealSpec[CNT] := Avolt[CNT*16]; //OlectraY[CNT+999];
120
121   //Utfører FFT
122   Form1.FastFourier1.Transform;
123
124   //Tømmer FFT-graf, og utfører div. oppsett
125   Form1.RChart1.ClearGraf;
126   Form1.RChart1.DataColor := clRed;
127   Form1.RChart1.MoveTo (Form1.FastFourier1.FreqOfLine(1,0.001),0);
```

```
128
129 //Plotter data til graf
130 For CNT := 1 to (Form1.FastFourier1.SpectrumSize div 2) do begin
131     y := Form1.FastFourier1.Magnitude[CNT];
132     Form1.RChart1.DrawTo (Form1.FastFourier1.FreqOfLine(CNT, 0.001),y)
133 end;
134
135 //Viser grafen
136 Form1.RChart1.ShowGraf;
137
138 end;
139
140 procedure TForm1.CheckBox1Click(Sender: TObject);
141 begin
142     //Kjører ny FFT og plotter FFT på nytt dersom man velger logaritmisk Y-akse
143     FFT;
144 end;
145
146 initialization
147     S := Sample.Create(True);
148
149 finalization
150     S.Terminate;
151     If MemHandle <> 0 Then
152         CWinBufFree(MemHandle);
153 end.
```

```
1 unit Unit2;
2
3 interface
4
5 uses
6   SysUtils, Graphics, Controls, Forms, Dialogs, StdCtrls, OleCtrls,
7   OlectraChart2D_TLB, Messages, Windows, Classes, CBW32;
8
9 type
10  Sample = class(TThread)
11  private
12    procedure Test;
13
14  protected
15    procedure Execute; override;
16  end;
17
18 implementation
19 uses Main;
20
21 procedure Sample.Test;
22 const
23   BoardNum : Integer = 0; //PCI-DAS1602/16 sin adresse
24
25 var
26   ULStat      : Integer;           //Status for Universal Library
27   ULStatString : String;          //Mellomlagring for feilmelding
28   I           : Integer;          //Teller-variabel for konverteri
29   ng av data
30 begin
31   //Sette knappens tekst til 'kjører', og koble ut knappen
32   Form1.Button1.Caption := 'Kjører';
33   Form1.Button1.Enabled := False;
34
35   //Sette opp Data Arrayer for Olectra
36   OlectraX := VarArrayCreate([1, Count], varDouble);
37   OlectraY := VarArrayCreate([1, Count], varDouble);
38
39   //Allokere minne-handle for AD-data
40   MemHandle := cbWinBufAlloc(Count+1);
41   If MemHandle = 0 Then Begin
42     MessageDlg('Memory allocation error',mtWarning,[mbOK],0);
43     Exit;
44   End;
45
46   //Sette Digital A til utgang
47   ULStat := cbDConfigPort(BoardNum,FIRSTPORTA,DIGITALOUT);
48   Form1.Label1.Caption := IntToStr(ULStat);
49   If ULStat <> 0 Then begin
50     ULStatString := IntToStr(ULStat);
51     MessageDlg('En feil oppstod ved konfigurering av digital-port: '+IntToStr(ULStat), mt
Warning,[mbOK],0);
52     Exit;
53   End;
54
55   //Sette Digital A0 høy
56   ULStat := cbDBitOut(BoardNum,FIRSTPORTA,0,1);
57   Form1.Label1.Caption := IntToStr(ULStat);
58   If ULStat <> 0 Then begin
59     MessageDlg('En feil oppstod ved setting av digital-port: '+IntToStr(ULStat),mtWarning
,[mbOK],0);
60     Exit;
61   End;
```

```
62
63 //Vente til geofon har stabilisert seg
64 Sleep(GeoWait);
65
66 // "slippe" geofonen
67 ULStat := cbDBitOut(BoardNum, FIRSTPORTA, 0, 0);
68 Form1.Label1.Caption := IntToStr(ULStat);
69 If ULStat <> 0 Then begin
70     MessageDlg('En feil oppstod ved setting av digital-port: '+IntToStr(ULStat), mtWarnin
g, [mbOK], 0);
71     Exit;
72 End;
73
74
75 //Start sampling, legg som bakgrunnsjobb.
76 ULStat := cbAInScan(BoardNum, 0, 0, Count, Rate, BIP1PT25VOLTS, MemHandle, 0);
77 Form1.Label1.Caption := IntToStr(ULStat);
78 If ULStat <> 0 Then begin
79     MessageDlg('En feil oppstod ved innhenting av data: '+IntToStr(ULStat), mtWarning, [m
bOK], 0);
80     Exit;
81 End;
82
83 //Flytte samplede data fra minne til data-array
84 ULStat := cbWinBufToArray(MemHandle, Adata[1], 1, Count);
85 Form1.Label1.Caption := IntToStr(ULStat);
86 If ULStat <> 0 Then begin
87     MessageDlg('Kunne flytte data fra Mem til array: '+IntToStr(ULStat), mtWarning, [mbOK
], 0);
88     Exit;
89 End;
90
91 //Konvertere data fra 16-bit til real-volt, og flytte data fra Array "Adata" til VarAr
ray Y, og sette X-verdier
92 For I := 1 to Count do begin
93     AVolt[I] := ((Adata[I]*2.5)/65535)-1.25;
94     OlectraY[I] := AVolt[I];
95     OlectraX[I] := I;
96 end;
97
98 //Plotte verdier i Chart
99 Form1.Chart2D1.ChartGroups.Item[1].Data.NumSeries := 1;
100 Form1.Chart2D1.ChartGroups.Item[1].Data.NumPoints[1] := Count;
101 Form1.Chart2D1.ChartArea.Axes.Item['X'].Max.Value := Count;
102 Form1.Chart2D1.ChartArea.Axes.Item['X'].DataMax.Value := Count;
103 Form1.Chart2D1.ChartGroups.Item[1].Data.CopyXVectorIn(1, OlectraX);
104 Form1.Chart2D1.ChartGroups.Item[1].Data.CopyYVectorIn(1, OlectraY);
105 Form1.Chart2D1.ChartGroups.Item[1].Data.NumPoints[1] := Count-1;
106
107 //Frigjøre MemHandle
108 If MemHandle <> 0 Then
109     CWinBufFree(MemHandle);
110
111 //Sette knappens tekst tilbake til 'Start test', og koble knappen inn i gjen
112 Form1.Button1.Caption := '&Start test';
113 Form1.Button1.Enabled := True;
114
115
116 end;
117
118 procedure Sample.Execute;
119 Begin
120     Repeat
121     Synchronize(Test);
```

```
122     FFT;  
123     Suspend;  
124     Until Terminated;  
125 End;  
126  
127 end.
```