

```
1 unit ADDA;
2
3 interface
4
5 uses
6   Classes;
7
8 type
9   T_ADDA = class(TThread)
10  private
11    { Private declarations }
12  protected
13    procedure Execute; override;
14    procedure ADDANow;
15  end;
16
17 implementation
18 Uses
19   CBW32, Variable, Dialogs, Windows, Main, SysUtils;
20
21 Procedure T_ADDA.ADDANow;
22 Var
23   ULStat : Integer;
24 Begin
25   // Sender ut data til DA
26   ULStat := cbAOut(BoardNum, Channel, BIP5VOLTS,SinusArray[OutPos]);
27
28   //Sjekker om det ble satt spenning ut på DA, hvis ikke gis feilmelding og programmet av
  sluttes
29   If ULStat <> 0 Then Begin
30     MessageDlg('Feil 01: Klarer ikke sende ut!',mtWarning,[mbOK],0);
31     Exit;
32   End;
33
34   // Leser inn data fra AD
35   ULStat := cbAIn(BoardNum, Channel, BIP5VOLTS,InputArray[InPos]);
36
37   //Sjekker om det ble samplet fra AD, hvis ikke gis feilmelding og programmet avsluttes
38   If ULStat <> 0 Then Begin
39     MessageDlg('Feil 02: Klarer ikke lese inn!',mtWarning,[mbOK],0);
40     Exit;
41   End;
42
43
44   //Øke rundeteller for DA, og sjekke om det er gitt ut en hel sinus
45   Inc(OutPos);
46   If OutPos >= 100 Then OutPos := 1; //Hvis en sinus er komplett, begynn på en ny
47
48   //Øke rundeteller for AD
49   Inc(InPos);
50
51   //Sjekke om riktig antall datapunkter er hentet inn
52   If InPos = 5000 Then Begin
53     InPos := 0; //Setter AD-teller tilbake til 0
54     TPlot.Resume; //Starter tråd for plotting av AD-data
55     TADDA.Suspend; //Tråden stopper seg selv
56   End;
57 End;
58
59 Procedure T_ADDA.Execute;
60 Var
61   T : LongInt;
62 Begin
63   FreeOnTerminate := True;
```

```
64   While not Terminated Do Begin
65       ADDANow;
66       //Tidsforsinkelse som regulerer hastigheten (frekvensen) på ADDA
67       For T := 1 To 156000 Do Begin
68           End;
69   End;
70 End;
71
72 End.
```

```
1 unit Main;
2
3 interface
4
5 uses
6   Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
7   StdCtrls, OleCtrls, OlectraChart2D_TLB, ComCtrls, ExtCtrls, RChart,
8   Gauges, Fourier;
9
10 type
11   TFrmMain = class(TForm)
12     Chart2D1: TChart2D;
13     Button1: TButton;
14     ScrollBar1: TScrollBar;
15     Label1: TLabel;
16     Label2: TLabel;
17     Label3: TLabel;
18     Label6: TLabel;
19     ScrollBar2: TScrollBar;
20     Label7: TLabel;
21     RChart1: TRChart;
22     FastFourier1: TFastFourier;
23     ComboBox1: TComboBox;
24     Label4: TLabel;
25     procedure FormCreate(Sender: TObject);
26     procedure Button1Click(Sender: TObject);
27     procedure ScrollBar1Change(Sender: TObject);
28     procedure ScrollBar2Change(Sender: TObject);
29     procedure ComboBox1Change(Sender: TObject);
30
31   private
32     { Private declarations }
33   public
34     { Public declarations }
35   end;
36
37 var
38   FrmMain: TFrmMain;
39   OlectraX      : Variant; // X-array for Olectra-chart
40   OlectraY      : Variant; // Y-array for Olectra-chart
41   ULStat        : Integer; // Variabel som inneholder status for AD-kort
42   Procedure FFT;
43 implementation
44 Uses
45   Variable, ADDA, Plot;
46 {$R *.DFM}
47
48 procedure TFrmMain.FormCreate(Sender: TObject);
49 begin
50
51
52   //Setter opp størrelse og plassering av objekter
53   FrmMain.Chart2D1.Width      := FrmMain.Width-8;
54   FrmMain.Chart2D1.Height     := FrmMain.Height - 450;
55   FrmMain.Button1.Left       := (FrmMain.Width div 2)-(FrmMain.Button1.Width div 2);
56   FrmMain.Button1.Top        := FrmMain.Chart2D1.Height + 40;
57   FrmMain.ScrollBar1.Left     := FrmMain.Button1.Left - 250;
58   FrmMain.ScrollBar1.Top      := FrmMain.Button1.Top + 4;
59   FrmMain.Scrollbar2.Left     := FrmMain.Button1.Left + 113;
60   FrmMain.Scrollbar2.Top      := FrmMain.Button1.Top + 14;
61   FrmMain.Label2.Left        := FrmMain.ScrollBar1.Left;
62   FrmMain.Label2.Top          := FrmMain.ScrollBar1.Top - 14;
63   FrmMain.Label3.Left        := FrmMain.ScrollBar1.Left + 190;
64   FrmMain.Label3.Top         := FrmMain.Label2.Top;
```

```
65   FrmMain.Label1.Left       := FrmMain.ScrollBar1.Left + 85;
66   FrmMain.Label1.Top        := FrmMain.ScrollBar1.Top - 28;
67   FrmMain.Label7.Left       := FrmMain.Button1.Left + FrmMain.Button1.Width + 5;
68   FrmMain.Label7.Top        := FrmMain.Chart2D1.Height + 1;
69   FrmMain.ScrollBar2.Top     := FrmMain.Label7.Top + 15;
70   FrmMain.ScrollBar2.Left    := FrmMain.Button1.Left + FrmMain.Button1.Width + 5;
71   FrmMain.Label4.Top        := FrmMain.ScrollBar2.Top + 20;
72   FrmMain.Label4.Left       := FrmMain.Button1.Left + FrmMain.Button1.Width + 5;
73   FrmMain.ComboBox1.Top     := FrmMain.Label4.Top + 15;
74   FrmMain.ComboBox1.Left    := FrmMain.Button1.Left + FrmMain.Button1.Width + 5;
75   FrmMain.RChart1.Width     := FrmMain.Width-8;
76   FrmMain.RChart1.Height    := FrmMain.Height - FrmMain.RChart1.Top - 50;
77   FrmMain.RChart1.Top       := FrmMain.Button1.Top + FrmMain.Button1.Height + 20;
78
79   //Setter scrollbar1 til 10000
80   FrmMain.ScrollBar1.Position := 10000;
81 end;
82
83 procedure TFrmMain.Button1Click(Sender: TObject);
84 var
85     I      : Integer; //Telle-variabel for utregning av Sinus-kurve
86 begin
87     If FrmMain.Button1.Caption = '&Start Test' then begin
88         //Lage sinus-kurve
89         For I := 1 to 100 Do Begin
90             SinusArray[I] := Trunc(FrmMain.ScrollBar1.Position*Sin((I*2*Pi)/100))+32768;
91         End;
92
93
94         //Initiere telle-variabler
95         InPos  := 1;
96         OutPos := 1;
97
98         //Nedtoner knappen og setter dens tekst til 'kjører test'
99         FrmMain.Button1.Caption := 'Kjører Test...';
100        FrmMain.Button1.Enabled := False;
101
102        //Starter tråden for ADDA
103        TADDA.Resume;
104    End;
105 end;
106
107 procedure TFrmMain.ScrollBar1Change(Sender: TObject);
108 begin
109     //Viser scrollbarens posisjon i programvinduet
110     Label6.Caption := IntToStr(FrmMain.Scrollbar1.Position);
111 end;
112
113 procedure TFrmMain.ScrollBar2Change(Sender: TObject);
114 begin
115     //Endrer visning på chart for signalplotting
116     FrmMain.Chart2D1.ChartArea.Axes.Item['X'].Min.Value := FrmMain.ScrollBar2.Position;
117     FrmMain.Chart2D1.ChartArea.Axes.Item['X'].Max.Value := FrmMain.ScrollBar2.Position + 1
118     000;
119 end;
120
121 procedure TFrmMain.ComboBox1Change(Sender: TObject);
122 begin
123     //Kjører ny FFT hvis vindustype blir endret
124     FFT;
125 end;
126 Procedure FFT;
127 Var
```

```
128   i : Integer;
129   y : Double;
130 begin
131
132   //Sette vindutype
133   If FrmMain.ComboBox1.Text = 'Rectangle' then
134     FrmMain.FastFourier1.WeightingWindow := fwRectangle;
135   If FrmMain.ComboBox1.Text = 'Triangle' then
136     FrmMain.FastFourier1.WeightingWindow := fwTriangle;
137   If FrmMain.ComboBox1.Text = 'Gaussian' then
138     FrmMain.FastFourier1.WeightingWindow := fwGauss;
139   If FrmMain.ComboBox1.Text = 'Hamming' then
140     FrmMain.FastFourier1.WeightingWindow := fwHamming;
141   If FrmMain.ComboBox1.Text = 'Blackman' then
142     FrmMain.FastFourier1.WeightingWindow := fwBlackman;
143   If FrmMain.ComboBox1.Text = 'cos2' then
144     FrmMain.FastFourier1.WeightingWindow := fwCos2;
145
146   FrmMain.RChart1.LogScaleY := True;           //Setter logaritmisk Y-akse på FFT-graf
147   FrmMain.FastFourier1.SpectrumSize := 4096; //Setter datamengde for FFT
148   FrmMain.FastFourier1.ClearReal;             //Tømmer FFT for reelle tall
149   FrmMain.FastFourier1.ClearImag;            //Tømmer FFT for imaginære tall
150
151   //Putter data inn i FFT
152   For i := 5 to 5000 do
153     FrmMain.FastFourier1.RealSpec[i] := OlectraY[i];
154
155   //Utfører FFT
156   FrmMain.FastFourier1.Transform;
157
158   //Tømmer FFT-graf, og utfører div. oppsett
159   FrmMain.RChart1.ClearGraf;
160   FrmMain.RChart1.RangeLoY := ChartYRanges[0,1];
161   FrmMain.RChart1.RangeHiY := ChartYRanges[0,2];
162   FrmMain.RChart1.DataColor := clRed;
163   FrmMain.RChart1.MoveTo (FrmMain.FastFourier1.FreqOfLine(1,0.001),0);
164
165   //Plotter data til graf
166   For i := 1 to (FrmMain.FastFourier1.SpectrumSize div 2) do begin
167     y := FrmMain.FastFourier1.Magnitude[i];
168     FrmMain.RChart1.DrawTo (FrmMain.FastFourier1.FreqOfLine(i, 0.001),y)
169   end;
170
171   //Viser grafen
172   FrmMain.RChart1.ShowGraf;
173
174 end;
175
176 Initialization
177   TADDA := T_ADDA.Create(True);
178   TADDA.Priority := tpTimeCritical;
179   TPlot := T_Plot.Create(True);
180
181 Finalization
182   TADDA.Terminate;
183
184 end.
```

```
1 unit Plot;
2
3 interface
4
5 uses
6   Classes, Dialogs, CBW32;
7
8 type
9   T_Plot = class(TThread)
10  private
11    { Private declarations }
12  protected
13    procedure Execute; override;
14    procedure PlotNow;
15  end;
16
17 implementation
18 Uses
19   Variable, Main, Graphics, Fourier;
20 Procedure T_Plot.PlotNow;
21 var
22   P           : Integer; // Teller for konvertering av AD-verdi
23 Begin
24   //Lage data-arrayer for Olectra
25   OlectraX := VarArrayCreate([1, 5000], varDouble);
26   OlectraY := VarArrayCreate([1, 5000], varDouble);
27
28   //Konvertere 16-bit AD-verdi til Volt
29   For P := 1 To 5000 Do Begin
30     OlectraY[P] := (((InputArray[P]*20)/65535)-10);
31     OlectraX[P] := P;
32   End;
33
34   // Setter verdi på DA til 0, så det ikke går strøm i geofonen
35   ULStat := cbAOut(BoardNum, Channel, BIP10VOLTS,32767);
36
37   // Sjekker at verdien er satt på DA, hvis ikke gis feilmelding og programmet stoppes
38   If ULStat <> 0 Then Begin
39     MessageDlg('Feil 01: Klarer ikke sende ut!',mtWarning,[mbOK],0);
40     Exit;
41   End;
42
43   //Sende data til chart
44   FrmMain.Chart2D1.ChartGroups.Item[1].Data.NumSeries := 1;
45   FrmMain.Chart2D1.ChartGroups.Item[1].Data.NumPoints[1] := 5000;
46   FrmMain.Chart2D1.ChartArea.Axes.Item['X'].Max.Value := 1000;
47   FrmMain.Chart2D1.ChartArea.Axes.Item['X'].DataMax.Value := 5000;
48   FrmMain.Chart2D1.ChartGroups.Item[1].Data.CopyXVectorIn(1,OlectraX);
49   FrmMain.Chart2D1.ChartGroups.Item[1].Data.CopyYVectorIn(1,OlectraY);
50   FrmMain.Chart2D1.ChartGroups.Item[1].Data.NumPoints[1] := FrmMain.Chart2D1.ChartGroups.
Item[1].Data.NumPoints[1] -1;
51
52   //Starte FFT-prosedyre
53   FFT;
54
55   FrmMain.Button1.Caption := '&Start Test';//Setter knappens tekst tilbake til 'start tes
t'
56   FrmMain.Button1.Enabled := True;           //Fjerner nedtoning av knapp
57
58 End;
59
60 procedure T_Plot.Execute;
61 begin
62   FreeOnTerminate := True;
```

```
63   While not Terminated Do Begin
64       Synchronize(PlotNow);
65       Suspend;
66   End;
67 end;
68
69 end.
70
```

```
1 unit Variable;
2
3 interface
4 Uses ADDA, Plot;
5 Var
6     SinusArray      : Array [1..100] of Word; //Array til å lagre sinus-verdier i
7     InputArray      : Array [0..5000] of Word; //Array til å lagre samplede data i
8     InPos           : Integer;                //Teller hvor langt programmet er kommet i AD-
samplingen
9     OutPos          : Integer;                //Teller for hvilket punk programmet er kommet
til på sinusen
10    TADDA           : T_ADDA;                 //Deklarerer tråden T_ADDA
11    TPlot           : T_Plot;                 //Deklarerer tråden T_PLOT
12
13 Const
14     BoardNum       : Integer = 0;           //PCI-DAS1602/16s nummer
15     Channel        : Integer = 0;           //Kanalen som skal brukes (0 på både AD og DA)
16
17 implementation
18
19
20 end.
```