



Vogelaar Electronics

Dorpsstraat 90
3751 ES Bunschoten Netherlands
Telefoon +31 (0)33 2980727
Fax +31 (0)847 115096
E-mail info@vogelaar-electronics.com

DS_inf.doc
25-07-2005

DRAFT

DelphiStamp VE08201

als interface

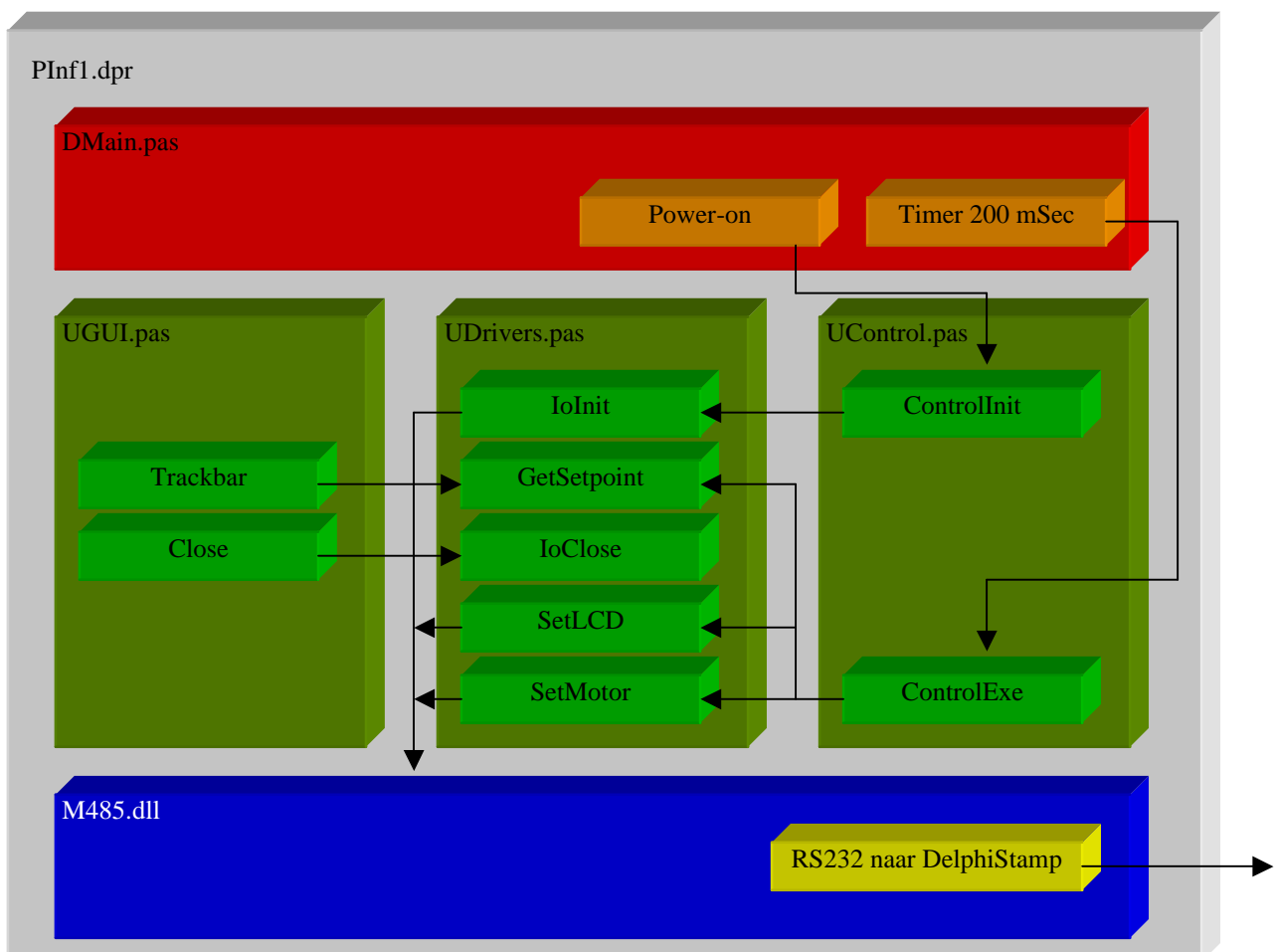
by Vogelaar Electronics
Bunschoten, Netherlands
25 – juli - 2005

De DelphiStamp als interface.

We omschrijven eerst het te realiseren project. Er dient op de PC een form aanwezig te zijn dat dienst doet als console om een stappenmotor te besturen. De motor is via een driver op de DelphiStamp is aangesloten. Op het form is een schuifpotentiometer aanwezig om de motor in de positie 0 t/m 360 graden te zetten alsmede een drietal buttons voor een snelle positionering op 0, 180 en 360 graden. Op de DelphiStamp is ook een LCD aangesloten waarop de ingestelde hoek is af te lezen.

De DelphiStamp wordt hierbij gebruikt als interface. Het besturingsprogramma loopt op een PC en is in Delphi geschreven. Op de DelphiStamp loopt een interface server om parameters welke door het PC programma naar de DelphiStamp worden geschreven synchroon te houden met de aangesloten hardware.

Om de kans op fouten in complexe programma's zo klein mogelijk te houden is het belangrijk om het totale programma op te delen in logische blokken. Delphi biedt hiervoor de unit, library, form en datamodule aan.



Het Delphi project PInf1.dpr is verdeeld in vijf delen t.w.

1. De DataModule DMain.pas. Een Datamodule is een naast een unit een container voor niet-visuele componenten zoals Timers. Deze DataModule voorziet in een Power-On event en een Timer om iedere 200 mSec de DelphiStamp van nieuwe informatie te voorzien, indien noodzakelijk.
2. Het Form UGUI.pas. Een Form is een unit dat visuele componenten kan bevatten. Dit form bevat een schuifpotentiometer – TTrackBar en drie TButtons. De buttons worden een integraal deel van de schuifpotentiometer voor een snelle positionering op 0, 180 en 360 graden.
3. De Unit UControl.pas. Deze unit bevat de functionaliteit van de applicatie. De procedure ControlInit initialiseert alle hardware bij Power-On. De procedure ControlExe ververst iedere 200 mSec de interface server op de DelphiStamp. Voor communicatie met de DelphiStamp en het GUI wordt gebruik gemaakt van de drivers in unit UDrivers.pas.
4. De Unit UDrivers.pas. Deze unit bevat de basis aansturings-procedures voor de GUI en de DelphiStamp. Omdat hierin zelden wijzigingen optreden is deze unit veelal herbruikbaar. Op de meegeleverde CD-Rom staat een standaard unit voor communicatie met alle standaard hardware.
5. Het library M485.DLL. Een DLL is een library unit voor hergebruik voor verschillende applicaties. Deze meegeleverde library bevat de communicatie-software met de DelphiStamp gebruikmakende van een instelbare RS232 poort.

Hieronder wordt het hierboven geschetste project uitgewerkt en alle niet zo voor de hand liggende Delphi technieken nader verklaard.

Zoals gebruikelijk maken we binnen de Delphi IDE een nieuw project met :

File | New | Application

en wordt het bewaard op de hard disk met :

File | Save all | <Directory\DS Inf1> UGUI.pas en PInf1.dpr

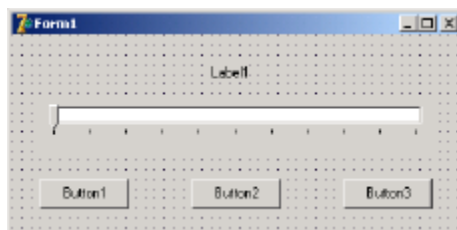
Plaats op het form :

Standaard | Label

Win32 | Trackbar

Standaard | Button (drie stuks)

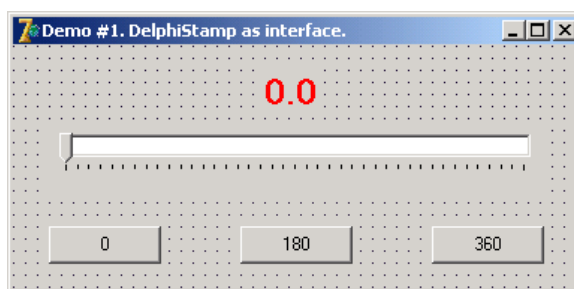
Het form ziet er nu als volgt uit:



Met behulp van de objectinspector worden onderstaande properties gewijzigd:

Object	Property	Value	Opmerking
Form1	Name	FGUI	Gebruik beschrijvende namen
	Caption	Demo #1	
Label1	Name	LbPos	
	Caption	0.0	
	Font.Size	18	Hoogte wordt 18 pixels
	Font.Style.fsBold	True	Dik
	Font.Color	clRed	Rood
Trackbar1	Name	TbrPos	
	Max	400	1 omwenteling 400 stappen
	Frequency	10	Schaalverdeling per 10 stappen
Button1	Name	Btn0	
	Caption	0	
Button2	Name	Btn180	
	Caption	180	
	Tag	200	180 graden is 200 stappen
Button3	Name	Btn360	
	Caption	360	
	Tag	400	360 graden is 400 stappen

Het form ziet er nu als volgt uit:



Het OnClick event van Btn0 wordt geïmplementeerd.

Het OnClick event van Btn180 en Btn360 is het zelfde als van Btn0. In de ObjectInspector van Btn180 en Btn360 kan in in de dropdown list van het OnClick event Btn0Click worden gespecificeerd.

```

Unit UGUI;
(* Demo program #1. Use DelphiStamp as interface.
   Provided as is, without warranty by Vogelaar Electronics, Bunschoten NL.
   Rev 0.10 23-07-05 Initial release *)

(* ===== Interface ===== *)
Interface

Uses    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
        Dialogs, StdCtrls, ComCtrls;

Type    TFGUI = Class (TForm)
        LbPos      : TLabel;
        TbrPos     : TTrackBar;
        Btn0       : TButton;
        Btn180     : TButton;
        Btn360     : TButton;
        Procedure Btn0Click (Sender : TObject);
        Procedure TbrPosChange (Sender : TObject);
        Procedure FormClose (Sender : TObject; Var Action : TCloseAction);
Private
        Procedure RefreshLabel;
Public
        End;

Var     FGUI : TFGUI;

(* ===== Implementation ===== *)
Implementation

{$R *.dfm}

Uses    UDrivers;

(* ===== Private ===== *)

Procedure TFGUI.RefreshLabel;
(* Update position label conform trackbar position *)

Begin
    LbPos.Caption := Format ('%.1f', [TbrPos.Position * 360 / 400])
End;

(* ===== Form ===== *)

Procedure TFGUI.FormClose (Sender : TObject; Var Action: TCloseAction);
(* Terminate connection with DelphiStamp *)

Begin
    IoClose
End;

Procedure TFGUI.Btn0Click (Sender : TObject);
(* User setting to 0, 180 or 360 degrees *)

Begin
    TbrPos.Position := (Sender As TComponent).Tag;
    RefreshLabel
End;

Procedure TFGUI.TbrPosChange (Sender : TObject);
(* Change of trackbar position by user *)

Begin
    RefreshLabel
End;

(* ===== End ===== *)

End.

```

In het OnClick event wordt de schuifpotentiometer op de waarde van de Tag ingesteld. Deze is 0, 200 of 400. Omdat de stappenmotor 400 stappen per omwenteling maakt komt dit overeen met 0, 180 en 360 graden.

Om het Label component de ingestelde hoek van de motor te tonen wordt de functie Format gebruikt. De format-strinf **%1f** geeft aan dat het resultaat een fixed aantal decimalen heeft t.w. 1. De te presenteren waarde staat tussen de vierkante haken []. De berekening $* 360 / 400$ converteert motor-stappen naar graden.

De DataModule UMain implementeerd het PowerOn event zodra DMain wordt gecreeerd. Het OnTimer event dient iedere 200 mSec de procedure ControlExe aan te roepen. Hiertoe wordt het Interval property van Timer1 op 200 gezet.

```
Unit DMain;
(* Demo program #1. Main controlling datamodule.
   Provided as is, without warranty by Vogelaar Electronics, Bunschoten NL.
   Rev 0.10 23-07-05 Initial release *)

(* ===== Interface ===== *)
Interface

Uses    SysUtils, Classes, ExtCtrls;

Type    TDmMain = Class (TDataModule)
          Timer1      : TTimer;
          Procedure DataModuleCreate (Sender : TObject);
          Procedure Timer1Timer      (Sender : TObject);
        Private
          Public
        End;

Var     DmMain : TDmMain;

(* ===== Implementation ===== *)
Implementation

{$R *.dfm}

Uses    UControl;

(* ===== Global ===== *)

Procedure TDmMain.DataModuleCreate (Sender : TObject);

Begin
  ControlInit
End;

Procedure TDmMain.Timer1Timer (Sender : TObject);

Begin
  ControlExe
End;

(* ===== End ===== *)

End.
```

De unit UDrivers voorzien in de basis communicatie tussen het user-interface en de hardware.

De functie GetSetpoint geeft de positie van de schuifpotentiometer op UGUI in motor-stappen.

In de DelphiStamp werkt de Interface Server met een parameter record. Dit record is in UDrivers gedeclareerd als TInfData. In de DelphiStamp start dit record op adres \$10D. Om de juiste adressen ook binnen Delphi beschikbaar te hebben is de constante PInfData gedeclareerd als pointer naar de TInfData structuur en is geinitialiseerd op \$10D.

Communicatie geschiedt via RS232 poort ComPort. In UDrivers is dit op Com1 gezet. Bij gebruik van een andere Com port kan deze constante worden gewijzigd.

De communicatie met de DelphiStamp wordt verzorgd door de library M485.DLL. De in deze DLL aanwezige routines DS485_XXXXX zijn daarom als external gedeclareerd.

```
Unit UDrivers;
(* Demo program #1. Unit containing drivers.
   Provided as is, without warranty by Vogelaar Electronics, Bunschoten NL.
   Rev 0.10 23-07-05 Initial release *)

(* ===== Interface ===== *)
Interface

Type    TStr8   = String [ 8];
        TStr16  = String [16];

Function GetSetpoint : Integer;
Procedure IoInit;
Procedure IoClose;
Procedure SetMotor (Pos : Integer);
Procedure SetLCD (S : TStr16);

(* ===== Implementation ===== *)
Implementation

Uses    UGUI, Dialogs, Forms, StrUtils;

Const   ComPort = 1;           // DelphiStamp connected to Com1

(* ===== DelphiStamp interface server ===== *)

Type    TInfDta = Packed Record
        Leds           : Byte;
        LcdStr1        : String [8];
        LcdStr2        : String [8];
        MotorSts       : Byte;
        MotorSol        : Word;
        MotorIst        : Word;
        Potm            : Word;
        Analog2         : Word;
        Dice            : Byte;
        Dta485          : Byte;
        Tx485Req        : ByteBool;
        Button          : ByteBool;
        Jumpers         : Byte;
    End;

Const   PInfDta : ^TInfDta = Pointer ($010C);

(* ===== DLL procedures ===== *)

Procedure DS485_Open (Port : Integer);           External 'M485.dll';
Procedure DS485_Close;                           External 'M485.dll';
Procedure DS485_Rd (N, MAddr : Word; Var Buf);   External 'M485.dll';
Procedure DS485_Wr (N, MAddr : Word; Var Buf);   External 'M485.dll';

(* ===== Global ===== *)

Var     InfDta : TInfDta;
        LcdS1  : TStr8 = ''; // Copy of LCD string 1
        LcdS2  : TStr8 = ''; // Copy of LCD string 2

Function GetSetpoint : Integer;
(* Provide sliderposition from UGUI *)

Begin
    Result := FGUI.TbrPos.Position;
End;

Procedure IoInit;
(* Energize stepper motor and clear LCD *)

Begin
    Try DS485_Open (ComPort);           // Open a connection with the DelphiStamp
```

```

Except
  ShowMessage ('Error. Unable to connect to DelphiStamp');
  Application.Terminate
End;
InfDta.MotorSts := $81;           // Energize motor
DS485_Wr (1, Word (@PInfDta.MotorSts), InfDta.MotorSts)
End;

Procedure IoClose;

Begin
  InfDta.MotorSts := $80;           // De-energize motor
  DS485_Wr (1, Word (@PInfDta.MotorSts), InfDta.MotorSts);
  DS485_Close;                       // Close communication port
End;

Procedure SetMotor (Pos : Integer);
(* Request for motorposition Pos *)

Begin
  InfDta.MotorSol := Pos;           // Sollwert := Pos
  DS485_Wr (2, Word (@PInfDta.MotorSol), InfDta.MotorSol)
End;

Procedure SetLCD (S : TStr16);
(* Write S to LCD *)

Var   SLcd   : TStr8;

Begin
  S := S + '                        ';           // S := 16 char's
  With InfDta Do
  Begin
    SLcd := LeftStr (S, 8);           // Left half of S
    If SLcd <> LcdStr1 Then           // If left section changed
    Begin
      LcdStr1 := SLcd;
      LcdStr1 [0] := Char (Byte (LcdStr1 [0]) Or $80);           // Set write request
      DS485_Wr (9, Word (@PInfDta.LcdStr1), InfDta.LcdStr1 [0]);
      LcdS1 := SLcd;
    End;
    SLcd := RightStr (S, 8);           // Right half of S
    If SLcd <> LcdStr2 Then           // If right section changed
    Begin
      LcdStr2 := SLcd;
      LcdStr2 [0] := Char (Byte (LcdStr2 [0]) Or $80);           // Set write request
      DS485_Wr (9, Word (@PInfDta.LcdStr2), InfDta.LcdStr2);
      LcdS2 := SLcd
    End;
  End
End;

(* ===== End ===== *)

End.

```

De procedure IoInit maakt verbinding met de DelphiStamp. Er zijn omstandigheden dat dit tot een crash leidt, bijvoorbeeld als de Com poort niet aanwezig is of als het bezet is door een ander programma. De Try .. Except constructie vangt deze exception af, geeft een foutmelding en breekt dan het programma af.

Het LCD met 16 karakters bestaat uit twee 8 karakter regels welke tegen elkaar zijn gemonteerd. Deze twee 8 karakter regels moeten apart naar de interface server worden gestuurd. De regels worden alleen verzonden als de string is veranderd, zie SLcd <> LcdStr1 in bovenstaande code.

UControl bevat de functionaliteit van dit programma:

```
Unit UControl;
(* Demo program #1. Unit containing control algorithms.
   Provided as is, without warranty by Vogelaar Electronics, Bunschoten NL.
   Rev 0.10 23-07-05 Initial release *)

(* ===== Interface ===== *)
Interface

Procedure ControlInit;
Procedure ControlExe;

(* ===== Implementation ===== *)
Implementation

Uses    UDrivers, SysUtils;

Var     SPLast  : Word = 0;

(* ===== Global ===== *)

Procedure ControlInit;
(* Called at power-on *)

Begin
  IoInit
End;

Procedure ControlExe;
(* Called at T = 200 mSec *)

Var     SP      : Word;

Begin
  SP := GetSetpoint;
  If SP <> SPLast Then
  Begin
    SetMotor (SP);
    SPLast := SP;
    SetLCD (Format ('Phie = %.1f', [SP * 360 / 400]));
  End
End;

(* ===== End ===== *)

End.
```

Na het compileren van bovenstaand programma wordt de DelphiStamp met behulp van de on-board schakelaars gereset en de interface server geactiveerd. Zodra bovenstaand programma wordt gestart kan de stappenmotor worden bestuurd.

Kijk voor meer informatie op www.vogelaar-electronics.com en www.learningdelphi.info