Software code

ACDCStepping.exe

.h header files

global2.h
global.h
Unit1.h
Unit2.h
Unit3.h
Unit4.h

```
 1: #ifndef global2H
 2:
 3: #define global2H
 4:
 5:
 6:
 7: #ifndef _Sum
 8: int Sum=0;
 9: #endif
10:
11: #ifndef _Teller
12: int Teller=0;
13: #endif
14:
15: #endif
```

```
 1:
 2: //*************************************************************
 3: //*                                                           *
 4: //* ----------------------------------------------------------*
 5: //*  Filename: global.h                                       *
 6: //*  Part of:  ACDCStepping.exe                               *
 7: //*                                                           *
 8: //*  Compiler: Borland C++ Builder 6 servicepack 4            *
 9: //*            CPort VCL , JVCL                               *
10: //*  Made by:  Eric Halmans                                   *
11: //*  For:      Fontys Highschool Eindhoven, Mechatronica      *
12: //*  Date:     April 2006                                     *
13: //*  Version:  1.0 beta test version                          *
14: //*                                                           *
15: //*  Description:                                             *
16: //*    This file is part a windows program,                   *
17: //*    which is used for universal AC & DC motor control,     *
18: //*    with a HCS12 TBoard from Elektronic Laden              *
19: //*    (Hbridge with PWM on PP0 & PP1, SCI1 input,IIC Output) *
20: //* ----------------------------------------------------------*
21: //*                                                           *
22: //*************************************************************
23:
24:
25: // global variables declaration
26:
27: #ifndef _Button1True
28: bool Button1True= false;
29: #endif
30:
31: #ifndef _ConnectionTrue
32: bool ConnectionTrue= false;
33: #endif
34:
35: #ifndef _ACFirstChar
36: char ACFirstChar;
37: #endif
38:
39: #ifndef _ACSeccondChar
40: char ACSeccondChar;
41: #endif
42:
43: #ifndef _ACThirdChar
44: char ACThirdChar;
45: #endif
46:
47: #ifndef _ACForthChar
48: char ACForthChar;
49: #endif
50:
51: #ifndef _ACFifthChar
52: char ACFifthChar;
53: #endif
54:
55: #ifndef _TIMER2TRUE
56: bool TIMER2TRUE=false;
57: #endif
58:
59: // delay function...
60: // used in debugging and was mainly used to slow down
```

```
61: // the communication with th HCS12...
62: // it was thought for a wile characters send in a row had to be slowed down.
63: #ifndef _Delay
64: void Delay(void)
65: {
66:  //     int i;
67:   //    for (i=0;i<1000;i++)
68:       // MaxInt =2^32-1
69:   //    {;}
70: }
71: #endif
72:
```

```
 1:
 2: //***************************************************************
 3: //*                                                             *
 4: //* ----------------------------------------------------------- *
 5: //*  Filename: Unit1.h                                          *
 6: //*  Part of:  ACDCStepping.exe                                 *
 7: //*                                                             *
 8: //*  Compiler: Borland C++ Builder 6 servicepack 4              *
 9: //*            CPort VCL , JVCL                                 *
10: //*  Made by:  Eric Halmans                                     *
11: //*  For:      Fontys Highschool Eindhoven, Mechatronica        *
12: //*  Date:     April 2006                                       *
13: //*  Version:  1.0 beta test version                            *
14: //*                                                             *
15: //*  Description:                                               *
16: //*    This file is part a windows program,                     *
17: //*    which is used for universal AC & DC motor control,       *
18: //*    with a HCS12 TBoard from Elektronic Laden                *
19: //*    (Hbridge with PWM on PP0 & PP1, SCI1 input,IIC Output)   *
20: //* ----------------------------------------------------------- *
21: //*                                                             *
22: //***************************************************************
23:
24: #ifndef Unit1H
25: #define Unit1H
26: //---------------------------------------------------------------------------
27: #include <Classes.hpp>
28: #include <Controls.hpp>
29: #include <StdCtrls.hpp>
30: #include <Forms.hpp>
31: #include "JvComponent.hpp"
32: #include "JvDialButton.hpp"
33: #include "JvExControls.hpp"
34: #include "JvSegmentedLEDDisplay.hpp"
35: #include <ExtCtrls.hpp>
36: #include "CPort.hpp"
37: #include "JvTimer.hpp"
38: #include <Menus.hpp>
39: #include <ComCtrls.hpp>
40: #include <ActnList.hpp>
41: #include "JvCreateProcess.hpp"
42: //---------------------------------------------------------------------------
43: class TForm1 : public TForm
44: {
45: __published:    // IDE-managed Components
46:         TComPort *ComPort1;
47:         TMainMenu *MainMenu1;
48:         TMenuItem *N1;
49:         TMenuItem *Exit1;
50:         TMenuItem *Connections1;
51:         TMenuItem *ComPort2;
52:         TMenuItem *Costumbaudrate1;
53:         TMenuItem *Help1;
54:         TMenuItem *Help2;
55:         TMenuItem *About1;
56:         TMenuItem *Connect1;
57:         TPageControl *PageControl1;
58:         TTabSheet *TabSheet1;
59:         TTabSheet *TabSheet2;
60:         TLabel *Label1;
```

```
 61:         TLabel *Label3;
 62:         TLabel *Label4;
 63:         TLabel *Label5;
 64:         TLabel *Label6;
 65:         TButton *Button1;
 66:         TJvDialButton *JvDialButton1;
 67:         TLabel *Label7;
 68:         TLabel *Label8;
 69:         TLabel *Label9;
 70:         TRadioGroup *RadioGroup1;
 71:         TPanel *Panel2;
 72:         TJvSegmentedLEDDisplay *JvSegmentedLEDDisplay1;
 73:         TMenuItem *StartTerminal1;
 74:         TJvCreateProcess *JvCreateProcess1;
 75:         TJvDialButton *JvDialButton2;
 76:         TPanel *Panel1;
 77:         TJvSegmentedLEDDisplay *JvSegmentedLEDDisplay2;
 78:         TButton *Button2;
 79:         TLabel *Label10;
 80:         TLabel *Label11;
 81:         TLabel *Label12;
 82:         TLabel *Label13;
 83:         TLabel *Label14;
 84:         TLabel *Label15;
 85:         TLabel *Label16;
 86:         TLabel *Label17;
 87:         TLabel *Label18;
 88:         TLabel *Label19;
 89:         TLabel *Label20;
 90:         TJvCreateProcess *JvCreateProcess2;
 91:         TMenuItem *StartDCFlow1;
 92:         TTabSheet *TabSheet3;
 93:         TRadioGroup *RadioGroup2;
 94:         TButton *Button3;
 95:         TLabel *Label2;
 96:         TMenuItem *ShowReceiveField1;
 97:         TRadioGroup *RadioGroup3;
 98:         TButton *Button4;
 99:         void __fastcall JvDialButton1Change(TObject *Sender);
100:         void __fastcall Button1Click(TObject *Sender);
101:         void __fastcall FormActivate(TObject *Sender);
102:         void __fastcall ComPort2Click(TObject *Sender);
103:         void __fastcall Exit1Click(TObject *Sender);
104:         void __fastcall About1Click(TObject *Sender);
105:         void __fastcall Costumbaudrate1Click(TObject *Sender);
106:         void __fastcall Connect1Click(TObject *Sender);
107:         void __fastcall FormDestroy(TObject *Sender);
108:         void __fastcall StartTerminal1Click(TObject *Sender);
109:         void __fastcall JvDialButton2Change(TObject *Sender);
110:         void __fastcall RadioGroup1Enter(TObject *Sender);
111:         void __fastcall Button2Click(TObject *Sender);
112:         void __fastcall PageControl1Change(TObject *Sender);
113:         void __fastcall StartDCFlow1Click(TObject *Sender);
114:         void __fastcall Button3Click(TObject *Sender);
115:         void __fastcall ComPort1RxChar(TObject *Sender, int Count);
116:         void __fastcall Button4Click(TObject *Sender);
117:         void __fastcall ShowReceiveField1Click(TObject *Sender);
118: private:    // User declarations
119: public:     // User declarations
120:         __fastcall TForm1(TComponent* Owner);
```

```
121: };
122: //---------------------------------------------------------------------------
123: extern PACKAGE TForm1 *Form1;
124: //---------------------------------------------------------------------------
125: #endif
126:
127: //---------------------------------------------------------------------------
128: // end of file Unit1.h
129:
130:
```

```
 1:
 2: //***************************************************************
 3: //*                                                             *
 4: //* ----------------------------------------------------------- *
 5: //*  Filename: Unit2.h                                          *
 6: //*  Part of:  ACDCStepping.exe                                 *
 7: //*                                                             *
 8: //*  Compiler: Borland C++ Builder 6 servicepack 4              *
 9: //*            CPort VCL , JVCL                                 *
10: //*  Made by:  Eric Halmans                                     *
11: //*  For:      Fontys Highschool Eindhoven, Mechatronica        *
12: //*  Date:     April 2006                                       *
13: //*  Version:  1.0 beta test version                            *
14: //*                                                             *
15: //*  Description:                                               *
16: //*    This file is part a windows program,                     *
17: //*    which is used for universal AC & DC motor control,       *
18: //*    with a HCS12 TBoard from Elektronic Laden                *
19: //*    (Hbridge with PWM on PP0 & PP1, SCI1 input,IIC Output)   *
20: //* ----------------------------------------------------------- *
21: //*                                                             *
22: //***************************************************************
23:
24: #ifndef Unit2H
25: #define Unit2H
26: //---------------------------------------------------------------------------
27: #include <Classes.hpp>
28: #include <Controls.hpp>
29: #include <StdCtrls.hpp>
30: #include <Forms.hpp>
31: //---------------------------------------------------------------------------
32: class TForm2 : public TForm
33: {
34: __published:    // IDE-managed Components
35:         TLabel *Label1;
36:         TLabel *Label2;
37:         TLabel *Label3;
38:         TLabel *Label4;
39:         TLabel *Label5;
40:         TLabel *Label6;
41:         TLabel *Label7;
42:         TButton *Button1;
43:         TLabel *Label8;
44:         void __fastcall Button1Click(TObject *Sender);
45: private:    // User declarations
46: public:     // User declarations
47:         __fastcall TForm2(TComponent* Owner);
48: };
49: //---------------------------------------------------------------------------
50: extern PACKAGE TForm2 *Form2;
51: //---------------------------------------------------------------------------
52: #endif
53:
54: //---------------------------------------------------------------------------
55: // end of file Unit2.h
56:
```

```
 1:
 2: //*************************************************************
 3: //*                                                           *
 4: //* ----------------------------------------------------------*
 5: //*  Filename: Unit3.h                                        *
 6: //*  Part of:  ACDCStepping.exe                               *
 7: //*                                                           *
 8: //*  Compiler: Borland C++ Builder 6 servicepack 4            *
 9: //*            CPort VCL , JVCL                               *
10: //*  Made by:  Eric Halmans                                   *
11: //*  For:      Fontys Highschool Eindhoven, Mechatronica      *
12: //*  Date:     April 2006                                     *
13: //*  Version:  1.0 beta test version                          *
14: //*                                                           *
15: //*  Description:                                             *
16: //*    This file is part a windows program,                   *
17: //*    which is used for universal AC & DC motor control,     *
18: //*    with a HCS12 TBoard from Elektronic Laden              *
19: //*    (Hbridge with PWM on PP0 & PP1, SCI1 input,IIC Output) *
20: //* ----------------------------------------------------------*
21: //*                                                           *
22: //*************************************************************
23:
24: #ifndef Unit3H
25: #define Unit3H
26: //---------------------------------------------------------------------------
27: #include <Classes.hpp>
28: #include <Controls.hpp>
29: #include <StdCtrls.hpp>
30: #include <Forms.hpp>
31: //---------------------------------------------------------------------------
32: class TForm3 : public TForm
33: {
34: __published:    // IDE-managed Components
35:         TLabel *Label1;
36:         TEdit *Edit1;
37:         TButton *Button1;
38:         TLabel *Label2;
39:         TLabel *Label3;
40:         void __fastcall Button1Click(TObject *Sender);
41: private:    // User declarations
42: public:     // User declarations
43:         __fastcall TForm3(TComponent* Owner);
44: };
45: //---------------------------------------------------------------------------
46: extern PACKAGE TForm3 *Form3;
47: //---------------------------------------------------------------------------
48: #endif
49:
50: //---------------------------------------------------------------------------
51: // end of file Unit3.h
52:
```

```
 1: //#include "global2.h"
 2: #include <Classes.hpp>
 3: #include <Controls.hpp>
 4: #include <StdCtrls.hpp>
 5:
 6: #ifndef Unit4H
 7: #define Unit4H
 8: //---------------------------------------------------------------------------
 9: #include <Classes.hpp>
10: #include <Controls.hpp>
11: #include <StdCtrls.hpp>
12: #include <Forms.hpp>
13:
14: //---------------------------------------------------------------------------
15: class TForm4 : public TForm
16: {
17: __published:    // IDE-managed Components
18:         TMemo *Memo1;
19:         TButton *Button1;
20:         TLabel *Label1;
21:         TLabel *Label2;
22:         TLabel *Label3;
23:         TLabel *Label4;
24:         void __fastcall Button1Click(TObject *Sender);
25: private:    // User declarations
26: public:     // User declarations
27:         __fastcall TForm4(TComponent* Owner);
28: };
29: //---------------------------------------------------------------------------
30: extern PACKAGE TForm4 *Form4;
31: extern Sum;
32: extern Teller;
33: //---------------------------------------------------------------------------
34: #endif
35:
```

Software code

ACDCStepping.exe

.cpp code files

ACDCStepping.cpp
Unit1.cpp
Unit2.cpp
Unit3.cpp
Unit4.cpp

```
 1: //---------------------------------------------------------------------------
 2:
 3: #include <vcl.h>
 4: #pragma hdrstop
 5: //---------------------------------------------------------------------------
 6: USEFORM("Unit1.cpp", Form1);
 7: USEFORM("Unit2.cpp", Form2);
 8: USEFORM("Unit3.cpp", Form3);
 9: USEFORM("Unit4.cpp", Form4);
10: //---------------------------------------------------------------------------
11: WINAPI WinMain(HINSTANCE, HINSTANCE, LPSTR, int)
12: {
13:         try
14:         {
15:                 Application->Initialize();
16:                 Application->CreateForm(__classid(TForm1), &Form1);
17:                 Application->CreateForm(__classid(TForm2), &Form2);
18:                 Application->CreateForm(__classid(TForm3), &Form3);
19:                 Application->CreateForm(__classid(TForm4), &Form4);
20:                 Application->Run();
21:         }
22:         catch (Exception &exception)
23:         {
24:                 Application->ShowException(&exception);
25:         }
26:         catch (...)
27:         {
28:                 try
29:                 {
30:                         throw Exception("");
31:                 }
32:                 catch (Exception &exception)
33:                 {
34:                         Application->ShowException(&exception);
35:                 }
36:         }
37:         return 0;
38: }
39: //---------------------------------------------------------------------------
40:
```

```
 1:
 2: //***************************************************************
 3: //*                                                             *
 4: //* ----------------------------------------------------------- *
 5: //*  Filename: Unit1.cpp                                        *
 6: //*  Part of:  ACDCStepping.exe                                 *
 7: //*                                                             *
 8: //*  Compiler: Borland C++ Builder 6 servicepack 4              *
 9: //*            CPort VCL , JVCL                                 *
10: //*  Made by:  Eric Halmans                                     *
11: //*  For:      Fontys Highschool Eindhoven, Mechatronica        *
12: //*  Date:     April 2006                                       *
13: //*  Version:  1.0 beta test version                            *
14: //*                                                             *
15: //*  Description:                                               *
16: //*    This file is part a windows program,                     *
17: //*    which is used for universal AC & DC motor control,       *
18: //*    with a HCS12 TBoard from Elektronic Laden                *
19: //*    (Hbridge with PWM on PP0 & PP1, SCI1 input,IIC Output)   *
20: //* ----------------------------------------------------------- *
21: //*                                                             *
22: //***************************************************************
23:
24:
25: #include <vcl.h>
26: #include <dstring.h>
27: #pragma hdrstop
28:
29: #ifndef Unit1H
30: #include "Unit1.h"
31: #endif
32: #ifndef Unit2H
33: #include "Unit2.h"
34: #endif
35: #ifndef Unit3H
36: #include "Unit3.h"
37: #endif
38: #ifndef Unit4H
39: #include "Unit4.h"
40: #endif
41: #ifndef globalH
42: #include "global.h"
43: #endif
44: #ifndef global2H
45: #include "global2.h"
46: #endif
47:
48: //---------------------------------------------------------------------------
49: #pragma package(smart_init)
50: #pragma link "JvComponent"
51: #pragma link "JvDialButton"
52: #pragma link "JvExControls"
53: #pragma link "JvSegmentedLEDDisplay"
54: #pragma link "CPort"
55: #pragma link "JvTimer"
56: #pragma link "JvCreateProcess"
57: #pragma resource "*.dfm"
58:
59: TForm1 *Form1;
60: //---------------------------------------------------------------------------
```

```
__fastcall TForm1::TForm1(TComponent* Owner)
        : TForm(Owner)
{
}


//---------------------------------------------------------------------------

// Short description:

// If JvDialButton1Change changes, JvSegmentedLEDDisplay1
// changes according to the dialbutton

// The dialbutton is divided in 3 zones,
// Zone 1: 256 steps PWM duty turn left
// Zone 2: Zero value zone
// Zone 3: 256 steps PWM duty turn right

// JvSegmentedLEDDisplay1 changes color according to change of a zone,
// and displays the step value in %.

// Pre: Old postion of JvDialButton1

// Post: new postion of JvDialButton1
//       Changed value (and color) of JvSegmentedLEDDisplay1

//---------------------------------------------------------------------------
void __fastcall TForm1::JvDialButton1Change(TObject *Sender)
{
  if (JvDialButton1->Position<=256)
   {
      JvSegmentedLEDDisplay1->SegmentLitColor=clLime;
      JvSegmentedLEDDisplay1->Text=((256-JvDialButton1->Position)/2.56);
   }
  else if (JvDialButton1->Position>256 && JvDialButton1->Position<=344)
   {
      JvSegmentedLEDDisplay1->SegmentLitColor=clYellow;
      JvSegmentedLEDDisplay1->Text=0,0;
   }
  else if (JvDialButton1->Position>344)
   {
      JvSegmentedLEDDisplay1->SegmentLitColor=clRed;
      JvSegmentedLEDDisplay1->Text=((JvDialButton1->Position-344)/2.56);
   }
}


//---------------------------------------------------------------------------

// Short description:

// If ComPort is open, and Button1 is clicked,
// a maximum of 2 characters,are send to the HCS12.
// This is done by reading out the JvDialButton1->Position.

// The dialbutton is divided in 3 zones,
// Zone 1: 256 steps PWM duty turn left
// Zone 2: Zero value zone
// Zone 3: 256 steps PWM duty turn right

```

```
121: // If JvDialButton1->Position = Zone1:
122: //    first char   = 0x52;
123: //    Seccond char = 0x00..0xff (duty left)
124: // If JvDialButton1->Position = Zone2:
125: //    first char   = 0x00;   Stops PWM
126: //    NO Seccond char
127: // If JvDialButton1->Position = Zone3:
128: //    first char   = 0x4c;
129: //    Seccond char = 0x00..0xff (duty Right)
130:
131: // Pre: postion of JvDialButton1
132:
133: // Post: Translates postion of JvDialButton1
134: //       in a 2 character coded value for the PWM channels of the HCS12
135:
136: //---------------------------------------------------------------------------
137: void __fastcall TForm1::Button1Click(TObject *Sender)
138: {
139:  if (ConnectionTrue==True)
140:  {
141:   unsigned char Temp;
142:   unsigned char* PTemp=&Temp;
143:   if (JvDialButton1->Position<=255)
144:     {
145:       Temp=0x52;
146:       Delay();
147:       ComPort1->Write(PTemp,1);
148:       Label3->Caption="0x52";
149:       Temp=(255-JvDialButton1->Position);
150:       Delay();
151:       ComPort1->Write(PTemp,1);
152:       Label4->Caption="0x" + IntToHex(Temp,2);
153:     }
154:   else if (JvDialButton1->Position>255 && JvDialButton1->Position<=345)
155:     {
156:       Temp=0x00;
157:       Delay();
158:       ComPort1->Write(PTemp,1);
159:       Label3->Caption="0x00";
160:       Label4->Caption="0x";
161:     }
162:   else if (JvDialButton1->Position>345)
163:     {
164:       Temp=0x4c;
165:       Delay();
166:       ComPort1->Write(PTemp,1);
167:       Label3->Caption="0x4c";
168:       Temp = (JvDialButton1->Position-345);
169:       Delay();
170:       ComPort1->Write(PTemp,1);
171:       Label4->Caption="0x" + IntToHex(Temp,2);
172:     }
173:  }
174: }
175:
176:
177: //---------------------------------------------------------------------------
178:
179: // Short description:
180:
```

```
181: // Program initialisation
182: // If the program is been executed, at startup the undifined
183: // Custom baudrate of the CPort VCL is set to 500.000 Bit / sec.
184: // (0.5 Mbit /sec.)
185:
186: // Pre:  undefined ComPort1->CustomBaudRate
187:
188: // Post: defined ComPort1->CustomBaudRate
189: //       Custom Baudrate = 500.000
190:
191: //---------------------------------------------------------------------------
192: void __fastcall TForm1::FormActivate(TObject *Sender)
193: {
194:   ComPort1->CustomBaudRate=500000;
195: }
196:
197:
198: //---------------------------------------------------------------------------
199:
200: // Short description:
201:
202: // Choosing this menu option, a setup dialog for the Comport VCL is shown.
203: // In this dialog, you can change:
204: // - Port number
205: // - Baudrate
206: // - Data bits
207: // - Stop bits
208: // - Parity
209: // - Flow control
210:
211: // Pre:  ComPort1
212:
213: // Post: setup dialog shown and changes effect ComPort1
214:
215: //---------------------------------------------------------------------------
216: void __fastcall TForm1::ComPort2Click(TObject *Sender)
217: {
218:   ComPort1->ShowSetupDialog();
219: }
220:
221:
222: //---------------------------------------------------------------------------
223:
224: // Short description:
225:
226: // Program end
227:
228: // Pre:  Project1.exe is running
229:
230: // Post: Project1.exe is stopped
231:
232: //---------------------------------------------------------------------------
233: void __fastcall TForm1::Exit1Click(TObject *Sender)
234: {
235:   exit(0);
236: }
237:
238:
239: //---------------------------------------------------------------------------
240:
```

```
241: // Short description:
242:
243: // Show about window
244:
245: // Pre:  Project1.exe is running
246:
247: // Post: On top of Project1.exe is the about window shown
248:
249: //---------------------------------------------------------------------------
250: void __fastcall TForm1::About1Click(TObject *Sender)
251: {
252:   Form2->Show();
253: }
254:
255:
256: //---------------------------------------------------------------------------
257:
258: // Short description:
259:
260: // Show set custom baud
261:
262: // Pre:  Project1.exe is running
263:
264: // Post: On top of Project1.exe is the custom baud window shown
265:
266: //---------------------------------------------------------------------------
267:
268: void __fastcall TForm1::Costumbaudrate1Click(TObject *Sender)
269: {
270:   Form3->Show();
271: }
272:
273:
274: //---------------------------------------------------------------------------
275:
276: // Short description:
277:
278: // This menu option if checked opens a connection with the Comport VCL
279:
280: // Pre:  If Connect1 = Checked Comport is open
281: //       If Connect1 = not Checked Comport is closed
282:
283: // Post: If Connect1 was Checked Comport is now closed
284: //       If Connect1 was not Checked Comport is now open
285:
286: //---------------------------------------------------------------------------
287: void __fastcall TForm1::Connect1Click(TObject *Sender)
288: {
289:  if (ComPort1->Connected==0)
290:   {
291:     ConnectionTrue = true;
292:     Connect1->Checked = true;
293:     ComPort1->Open();
294:   }
295:
296:  else if (ComPort1->Connected)
297:   {
298:     ConnectionTrue = false;
299:     Connect1->Checked = false;
300:     ComPort1->Close();
```

```
301:   }
302: }
303:
304:
305: //---------------------------------------------------------------------------
306:
307: // Short description:
308:
309: // This function is entered when the program is terminated.
310: // Before termination it sends a stop character to the HCS12,
311: // if connected.
312:
313: // Pre:  Comport is connected or not
314:
315: // Post: If connected: 0x00 stop char is send and Comport1 is closed
316: //       If not connected: Comport1 is opened,
317: //       0x00 stop char is send and Comport1 is closed.
318:
319: //---------------------------------------------------------------------------
320: void __fastcall TForm1::FormDestroy(TObject *Sender)
321: {
322:   unsigned char Temp;
323:   unsigned char* PTemp=&Temp;
324:   if (ConnectionTrue==True)
325:    {
326:      Temp =0x00;
327:      Delay();
328:      ComPort1->Write(PTemp,1);
329:      ConnectionTrue = false;
330:      Connect1->Checked = false;
331:      ComPort1->Close();
332:    }
333:   else if (ConnectionTrue==False)
334:    {
335:      ComPort1->Open();
336:      Temp =0x00;
337:      Delay();
338:      ComPort1->Write(PTemp,1);
339:      ConnectionTrue = false;
340:      Connect1->Checked = false;
341:      ComPort1->Close();
342:    }
343: }
344:
345:
346: //---------------------------------------------------------------------------
347:
348: // Short description:
349:
350: // This function is opens an executable named Terminal.exe
351: // Before opening Terminal.exe, if connected this function,
352: // sends a stop character (0x00) to the HCS12,
353: // and closes the comport connection.
354: // Terminal.exe is a terminal program which gives maximum control,
355: // on what to send, what is received and with what speed it should happen
356:
357:
358: // Pre:  Comport is either connected or not
359:
360: // Post: If connected:
```

```
361: //         -  0x00 stop char is send and Comport1 is closed.
362: //         -  Terminal.exe is been started
363: //       If not connected, just Terminal.exe is been started
364:
365: //---------------------------------------------------------------------------
366: void __fastcall TForm1::StartTerminal1Click(TObject *Sender)
367: {
368:   unsigned char Temp;
369:   unsigned char* PTemp=&Temp;
370:
371:   if (ConnectionTrue==true)
372:    {
373:      Temp =0x00;
374:      Delay();
375:      ComPort1->Write(PTemp,1);
376:      ConnectionTrue=false;
377:      Connect1->Checked = false;
378:      ComPort1->Close();
379:      JvCreateProcess1->Run();
380:    }
381:   else
382:    {
383:      JvCreateProcess1->Run();
384:    }
385: }
386:
387:
388: //---------------------------------------------------------------------------
389:
390: // Short description:
391:
392: // This function is entered when the program is terminated.
393: // Before termination it sends a stop character to the HCS12,
394: // if connected.
395:
396: // Pre:  Comport is connected or not
397:
398: // Post: If connected: 0x00 stop char is send and Comport1 is closed
399: //       If not connected: Comport1 is opened,
400: //       0x00 stop char is send and Comport1 is closed.
401:
402: //---------------------------------------------------------------------------
403: void __fastcall TForm1::JvDialButton2Change(TObject *Sender)
404: {
405: // change off JvDialButton2
406:   if  (RadioGroup1->ItemIndex==0)
407:    {
408:       ACFirstChar=0x00;
409:       Label19->Caption="0x";
410:       Label16->Caption="0x";
411:       Label15->Caption="0x";
412:       Label12->Caption="0x";
413:       Label11->Caption="0x00";
414:    }
415:
416:   else if  (RadioGroup1->ItemIndex==1)
417:    {
418:       JvSegmentedLEDDisplay2->Text=
419:       FloatToStrF((((float)(JvDialButton2->Position)*4.0)/1000+1.0),ffFixed,4,1);
420:       ACFifthChar=StrToInt(JvSegmentedLEDDisplay2->Text[3]);
```

```
421:       ACForthChar=StrToInt(JvSegmentedLEDDisplay2->Text[1]);
422:       ACThirdChar=0x00;
423:       ACSeccondChar=0x00;
424:       ACFirstChar=0x55;
425:       Label19->Caption="0x"+IntToHex(StrToInt(JvSegmentedLEDDisplay2->Text[3]),2);
426:       Label16->Caption="0x"+IntToHex(StrToInt(JvSegmentedLEDDisplay2->Text[1]),2);
427:       Label15->Caption="0x"+IntToHex(0,2);
428:       Label12->Caption="0x"+IntToHex(0,2);
429:       Label11->Caption="0x"+IntToHex(0x55,2);
430:     }
431:    else if  (RadioGroup1->ItemIndex==2)
432:     {
433:       JvSegmentedLEDDisplay2->Text=
434:       FloatToStrF((((float)(JvDialButton2->Position)*5.0)/1000+5.0),ffFixed,4,1);
435:       if (((((JvDialButton2->Position)*5.0)/1000)+5.0)<9.95)
436:         {
437:           ACFifthChar=StrToInt(JvSegmentedLEDDisplay2->Text[3]);
438:           ACForthChar=StrToInt(JvSegmentedLEDDisplay2->Text[1]);
439:           ACThirdChar=0x00;
440:           ACSeccondChar=0x00;
441:           ACFirstChar=0x55;
442:           Label19->Caption="0x"+IntToHex(StrToInt(JvSegmentedLEDDisplay2->Text[3]),2);
443:           Label16->Caption="0x"+IntToHex(StrToInt(JvSegmentedLEDDisplay2->Text[1]),2);
444:           Label15->Caption="0x"+IntToHex(0,2);
445:           Label12->Caption="0x"+IntToHex(0,2);
446:           Label11->Caption="0x"+IntToHex(0x55,2);
447:         }
448:       if (((((JvDialButton2->Position)*5.0)/1000)+5.0)>=9.95)
449:        {
450:          ACFifthChar=StrToInt(JvSegmentedLEDDisplay2->Text[4]);
451:          ACForthChar=StrToInt(JvSegmentedLEDDisplay2->Text[2]);
452:          ACThirdChar=StrToInt(JvSegmentedLEDDisplay2->Text[1]);
453:          ACSeccondChar=0x00;
454:          ACFirstChar=0x55;
455:          Label19->Caption="0x"+IntToHex(StrToInt(JvSegmentedLEDDisplay2->Text[4]),2);
456:          Label16->Caption="0x"+IntToHex(StrToInt(JvSegmentedLEDDisplay2->Text[2]),2);
457:          Label15->Caption="0x"+IntToHex(StrToInt(JvSegmentedLEDDisplay2->Text[1]),2);
458:          Label12->Caption="0x"+IntToHex(0,2);
459:          Label11->Caption="0x"+IntToHex(0x55,2);
460:        }
461:      }
462:    else if  (RadioGroup1->ItemIndex==3)
463:     {
464:       JvSegmentedLEDDisplay2->Text=
465:       FloatToStrF((((float)(JvDialButton2->Position)*10.0)/1000)+10.0,ffFixed,4,1);
466:       ACFifthChar=StrToInt(JvSegmentedLEDDisplay2->Text[4]);
467:       ACForthChar=StrToInt(JvSegmentedLEDDisplay2->Text[2]);
468:       ACThirdChar=StrToInt(JvSegmentedLEDDisplay2->Text[1]);
469:       ACSeccondChar=0x00;
470:       ACFirstChar=0x55;
471:       Label19->Caption="0x"+IntToHex(StrToInt(JvSegmentedLEDDisplay2->Text[4]),2);
472:       Label16->Caption="0x"+IntToHex(StrToInt(JvSegmentedLEDDisplay2->Text[2]),2);
473:       Label15->Caption="0x"+IntToHex(StrToInt(JvSegmentedLEDDisplay2->Text[1]),2);
474:       Label12->Caption="0x"+IntToHex(0,2);
475:       Label11->Caption="0x"+IntToHex(0x55,2);
476:     }
477:    else if  (RadioGroup1->ItemIndex==4)
478:     {
479:       JvSegmentedLEDDisplay2->Text=
480:       FloatToStrF((((float)(JvDialButton2->Position)*20.0)/1000)+20.0,ffFixed,4,1);
```

```
481:      ACFifthChar=StrToInt(JvSegmentedLEDDisplay2->Text[4]);
482:      ACForthChar=StrToInt(JvSegmentedLEDDisplay2->Text[2]);
483:      ACThirdChar=StrToInt(JvSegmentedLEDDisplay2->Text[1]);
484:      ACSeccondChar=0x00;
485:      ACFirstChar=0x55;
486:      Label19->Caption="0x"+IntToHex(StrToInt(JvSegmentedLEDDisplay2->Text[4]),2);
487:      Label16->Caption="0x"+IntToHex(StrToInt(JvSegmentedLEDDisplay2->Text[2]),2);
488:      Label15->Caption="0x"+IntToHex(StrToInt(JvSegmentedLEDDisplay2->Text[1
489:      Label12->Caption="0x"+IntToHex(0,2);
490:      Label11->Caption="0x"+IntToHex(0x55,2);
491:    }
492:   else if  (RadioGroup1->ItemIndex==5)
493:    {
494:      JvSegmentedLEDDisplay2->Text=
495:      FloatToStrF((((float)(JvDialButton2->Position)*40.0)/1000)+40.0,ffFixed,4,1);
496:      ACFifthChar=StrToInt(JvSegmentedLEDDisplay2->Text[4]);
497:      ACForthChar=StrToInt(JvSegmentedLEDDisplay2->Text[2]);
498:      ACThirdChar=StrToInt(JvSegmentedLEDDisplay2->Text[1]);
499:      ACSeccondChar=0x00;
500:      ACFirstChar=0x55;
501:      Label19->Caption="0x"+IntToHex(StrToInt(JvSegmentedLEDDisplay2->Text[4]),2);
502:      Label16->Caption="0x"+IntToHex(StrToInt(JvSegmentedLEDDisplay2->Text[2]),2);
503:      Label15->Caption="0x"+IntToHex(StrToInt(JvSegmentedLEDDisplay2->Text[1]),2);
504:      Label12->Caption="0x"+IntToHex(0,2);
505:      Label11->Caption="0x"+IntToHex(0x55,2);
506:    }
507:   else if  (RadioGroup1->ItemIndex==6)
508:    {
509:      JvSegmentedLEDDisplay2->Text=
510:      FloatToStrF((((float)(JvDialButton2->Position)*80.0)/1000)+80.0,ffFixed,4,1);
511:      if (((((JvDialButton2->Position)*80.0)/1000)+80.0)<99.95)
512:       {
513:          ACFifthChar=StrToInt(JvSegmentedLEDDisplay2->Text[4]);
514:          ACForthChar=StrToInt(JvSegmentedLEDDisplay2->Text[2]);
515:          ACThirdChar=StrToInt(JvSegmentedLEDDisplay2->Text[1]);
516:          ACSeccondChar=0x00;
517:          ACFirstChar=0x55;
518:          Label19->Caption="0x"+IntToHex(StrToInt(JvSegmentedLEDDisplay2->Text[4]),2);
519:          Label16->Caption="0x"+IntToHex(StrToInt(JvSegmentedLEDDisplay2->Text[2]),2);
520:          Label15->Caption="0x"+IntToHex(StrToInt(JvSegmentedLEDDisplay2->Text[1]),2);
521:          Label12->Caption="0x"+IntToHex(0,2);
522:          Label11->Caption="0x"+IntToHex(0x55,2);
523:       }
524:      else if ((((JvDialButton2->Position)*80.0)/1000)+80.0)>=99.95)
525:      {
526:         ACFifthChar=StrToInt(JvSegmentedLEDDisplay2->Text[5]);
527:         ACForthChar=StrToInt(JvSegmentedLEDDisplay2->Text[3]);
528:         ACThirdChar=StrToInt(JvSegmentedLEDDisplay2->Text[2]);
529:         ACSeccondChar=StrToInt(JvSegmentedLEDDisplay2->Text[1]);
530:         ACFirstChar=0x55;
531:         Label19->Caption="0x"+IntToHex(StrToInt(JvSegmentedLEDDisplay2->Text[5]),2);
532:         Label16->Caption="0x"+IntToHex(StrToInt(JvSegmentedLEDDisplay2->Text[3]),2);
533:         Label15->Caption="0x"+IntToHex(StrToInt(JvSegmentedLEDDisplay2->Text[2]),2);
534:         Label12->Caption="0x"+IntToHex(StrToInt(JvSegmentedLEDDisplay2->Text[1]),2);
535:         Label11->Caption="0x"+IntToHex(0x55,2);
536:      }
537:   }
538:   else if  (RadioGroup1->ItemIndex==7)
539:      {
540:         JvSegmentedLEDDisplay2->Text=
```

```
541:         FloatToStrF((((float)(JvDialButton2->Position)*160.0)/1000)+160.0,ffFixed,4,1);
542:         ACFifthChar=StrToInt(JvSegmentedLEDDisplay2->Text[5]);
543:         ACForthChar=StrToInt(JvSegmentedLEDDisplay2->Text[3]);
544:         ACThirdChar=StrToInt(JvSegmentedLEDDisplay2->Text[2]);
545:         ACSeccondChar=StrToInt(JvSegmentedLEDDisplay2->Text[1]);
546:         ACFirstChar=0x55;
547:         Label19->Caption="0x"+IntToHex(StrToInt(JvSegmentedLEDDisplay2->Text[5]),2);
548:         Label16->Caption="0x"+IntToHex(StrToInt(JvSegmentedLEDDisplay2->Text[3]),2);
549:         Label15->Caption="0x"+IntToHex(StrToInt(JvSegmentedLEDDisplay2->Text[2]),2);
550:         Label12->Caption="0x"+IntToHex(StrToInt(JvSegmentedLEDDisplay2->Text[1]),2);
551:         Label11->Caption="0x"+IntToHex(0x55,2);
552:      }
553: }
554:
555:
556: //---------------------------------------------------------------------------
557:
558: // Short description:
559:
560: // When the AC Tabsheet radiogroup is entered,
561: // a couple of values are reinitialisized:
562: //    - The AC JvDialbutton is set to the zero position
563: //    - The Labels coding for the characters to be send are set to 0x
564:
565:
566: // Pre:  None
567:
568: // Post: If radiogroup is entered, labels and dialbutton are reinitialized
569:
570: //---------------------------------------------------------------------------
571: void __fastcall TForm1::RadioGroup1Enter(TObject *Sender)
572: {
573:    JvDialButton2->Position=0;
574:    JvSegmentedLEDDisplay2->Text="";
575:    Label19->Caption="0x";
576:    Label16->Caption="0x";
577:    Label15->Caption="0x";
578:    Label12->Caption="0x";
579:    Label11->Caption="0x";
580: }
581:
582:
583: //---------------------------------------------------------------------------
584:
585: // Short description:
586:
587: // When the "Send AC" Button2 is clicked,
588: // the characters coding for the AC freqency are been send to the HCS12
589: // NB each character codes for either go ac or for a 0..9 digit.
590: // 4 characters make up the frequency : (xyz,s) x,y,z,s = 0..9
591:
592: // Pre:  ConnectionTrue has to be true to send the characters
593: //          if not pressing Button2 has no effect
594:
595: // Post: one stop character has been send,
596: //       or 5 characters coding for the HCS12 AC frequency have been send.
597:
598: //---------------------------------------------------------------------------
599: void __fastcall TForm1::Button2Click(TObject *Sender)
600: {
```

```
601:    unsigned char Temp;
602:    unsigned char* PTemp=&Temp;
603:
604:    if (ConnectionTrue==True)
605:     {
606:       if (RadioGroup1->ItemIndex==0)
607:         {
608:           Temp=0x00;
609:           Delay();
610:           ComPort1->Write(PTemp,1);
611:         }
612:       else
613:         {
614:           Temp=ACFirstChar;
615:           Delay();
616:           ComPort1->Write(PTemp,1);
617:           Temp=ACSeccondChar;
618:           Delay();
619:           ComPort1->Write(PTemp,1);
620:           Temp=ACThirdChar;
621:           Delay();
622:           ComPort1->Write(PTemp,1);
623:           Temp=ACForthChar;
624:           Delay();
625:           ComPort1->Write(PTemp,1);
626:           Delay();
627:           Temp=ACFifthChar;
628:           ComPort1->Write(PTemp,1);
629:         }
630:    }
631: }
632:
633:
634: //---------------------------------------------------------------------------
635:
636: // Short description:
637:
638: // Thisfunction reinitializes some objects when the user switches
639: // between The tabsheets reinitialisized:
640: //    - The AC or the DC JvDialbutton is set to the zero position
641: //    - a stop character (0x00) is send to the HCS12
642:
643:
644: // Pre:  Other tabsheet active
645:
646: // Post: user switches tabsheet:
647: //        - If new Tabsheet = AC: AC JvDialButton is set to zero position
648: //        - If new Tabsheet = DC: DC JvDialButton is set to middle position
649: //        - In all 3 cases: 0x00 (stop character) is send by Comport to HCS12
650:
651: //---------------------------------------------------------------------------
652: void __fastcall TForm1::PageControl1Change(TObject *Sender)
653: {
654:    unsigned char Temp;
655:    unsigned char* PTemp=&Temp;
656:
657:    if (PageControl1->ActivePage==TabSheet1)
658:     {
659:       JvDialButton1->Position=300;
660:       if (ConnectionTrue==True)
```

```
661:        {
662:          Temp =0x00;
663:          ComPort1->Write(PTemp,1);
664:        }
665:     }
666:    else if (PageControl1->ActivePage==TabSheet2)
667:     {
668:       JvDialButton1->Position=0;
669:       if (ConnectionTrue==True)
670:        {
671:          Temp =0x00;
672:          ComPort1->Write(PTemp,1);
673:        }
674:     }
675:    else if (PageControl1->ActivePage==TabSheet3)
676:     {
677:       if (ConnectionTrue==True)
678:        {
679:          Temp =0x00;
680:          ComPort1->Write(PTemp,1);
681:        }
682:     }
683: }
684:
685:
686: //---------------------------------------------------------------------------
687:
688: // Short description:
689:
690: // This function is opens an executable named Terminal.exe
691: // Before opening DCFlow.exe, if connected this function,
692: // sends a stop character (0x00) to the HCS12,
693: // and closes the comport connection.
694: // NB: DCFlow.exe is a program that
695: //       fluidly updates the DC PWM values every 0.15s
696:
697:
698: // Pre:  Comport is either connected or not
699:
700: // Post: If connected:
701: //         -  0x00 stop char is send and Comport1 is closed.
702: //         -  DCFlow.exe is been started
703: //       If not connected, just DCFlow.exe is been started
704:
705: //---------------------------------------------------------------------------
706: void __fastcall TForm1::StartDCFlow1Click(TObject *Sender)
707: {
708:    unsigned char Temp;
709:    unsigned char* PTemp=&Temp;
710:
711:    if (ConnectionTrue==true)
712:     {
713:       Temp =0x00;
714:       Delay();
715:       ComPort1->Write(PTemp,1);
716:       ConnectionTrue=false;
717:       Connect1->Checked = false;
718:       ComPort1->Close();
719:       JvCreateProcess2->Run();
720:     }
```

```
721:   else
722:    {
723:       JvCreateProcess2->Run();
724:    }
725: }
726:
727:
728: //---------------------------------------------------------------------------
729: // Short description:
730:
731: // This function sends a speed character for the RealTime Interrupt frequency
732: // if Radiogroup2 RadioButton Slow speed is selected, RTI=15,625 Hz
733: // if Radiogroup2 RadioButton High speed is selected, RTI=15,625 KHz
734:
735: // Pre:  Comport is either connected or not
736:
737: // Post: If connected:
738: //        -  if Radiogroup2[0] selected: send 0x05 to HCS12 (RTI=15,625 Hz)
739: //        -  f Radiogroup2[1] selected: send 0x10 to HCS12 (RTI=15,625 KHz)
740: //       If not connected: no action
741:
742: //---------------------------------------------------------------------------
743: void __fastcall TForm1::Button3Click(TObject *Sender)
744: {
745:    if (ConnectionTrue==True)
746:     {
747:        unsigned char Temp;
748:        unsigned char* PTemp=&Temp;
749:        if (RadioGroup2->ItemIndex==0)
750:         {
751:            Temp=0x05;
752:            ComPort1->Write(PTemp,1);
753:         }
754:        else if (RadioGroup2->ItemIndex==1)
755:         {
756:            Temp=0x10;
757:            ComPort1->Write(PTemp,1);
758:         }
759:     }
760: }
761:
762: //---------------------------------------------------------------------------
763:
764: // Short description:
765:
766: // If a character is received by Comport VCL, it will be displayed by this
767: // function as an hexadecimal value in the Received memo field
768:
769: // Pre:  Any received character
770:
771: // Post: The received character is displayed as an hexadecimal value in
772: //       the Received Memo field
773:
774: //---------------------------------------------------------------------------
775: void __fastcall TForm1::ComPort1RxChar(TObject *Sender, int Count)
776: {
777: AnsiString Str2;
778:   unsigned char a;
779:   unsigned char* Pa=&a;
780:   int *PointSum=&Sum;
```

```
781:   int *PointTeller=&Teller;
782:
783:   ComPort1->Read(Pa,Count);
784:   *PointSum+=a;
785:   *PointTeller+=1;
786:   Form4->Memo1->Text = Form4->Memo1->Text + "  " + IntToHex(a,2);
787:   Form4->Label3->Caption=*PointSum;
788:   Form4->Label4->Caption=*PointTeller;
789: }
790: //---------------------------------------------------------------------------
791: // end of file Unit1.cpp
792: void __fastcall TForm1::Button4Click(TObject *Sender)
793: {
794:   if (ConnectionTrue==True)
795:     {
796:        unsigned char Temp;
797:        unsigned char* PTemp=&Temp;
798:        if (RadioGroup3->ItemIndex==0)
799:         {
800:            Temp=0x25;
801:            ComPort1->Write(PTemp,1);
802:         }
803:        else if (RadioGroup3->ItemIndex==1)
804:         {
805:            Temp=0x20;
806:            ComPort1->Write(PTemp,1);
807:         }
808:     }
809: }
810:
811:
812: //---------------------------------------------------------------------------
813:
814: void __fastcall TForm1::ShowReceiveField1Click(TObject *Sender)
815: {
816:     Form4->Show();
817: }
818: //---------------------------------------------------------------------------
819:
820:
```

```
 1:
 2: //***********************************************************
 3: //*                                                         *
 4: //* ------------------------------------------------------- *
 5: //*  Filename: Unit2.cpp                                    *
 6: //*  Part of:  ACDCStepping.exe                             *
 7: //*                                                         *
 8: //*  Compiler: Borland C++ Builder 6 servicepack 4          *
 9: //*            CPort VCL , JVCL                             *
10: //*  Made by:  Eric Halmans                                 *
11: //*  For:      Fontys Highschool Eindhoven, Mechatronica    *
12: //*  Date:     April 2006                                   *
13: //*  Version:  1.0 beta test version                        *
14: //*                                                         *
15: //*  Description:                                           *
16: //*    This file is part a windows program,                 *
17: //*    which is used for universal AC & DC motor control,   *
18: //*    with a HCS12 TBoard from Elektronic Laden            *
19: //*    (Hbridge with PWM on PP0 & PP1, SCI1 input,IIC Output)  *
20: //* ------------------------------------------------------- *
21: //*                                                         *
22: //***********************************************************
23:
24: #include <vcl.h>
25: #pragma hdrstop
26:
27:
28: #ifndef Unit2H
29: #include "Unit2.h"
30: #endif
31:
32: //---------------------------------------------------------------------------
33: #pragma package(smart_init)
34: #pragma resource "*.dfm"
35: TForm2 *Form2;
36: //---------------------------------------------------------------------------
37: __fastcall TForm2::TForm2(TComponent* Owner)
38:         : TForm(Owner)
39: {
40: }
41:
42:
43: //---------------------------------------------------------------------------
44:
45: // Short description:
46:
47: // Button to close about window
48:
49: // Pre:  about window is shown
50:
51: // Post: about window is closed
52:
53: //---------------------------------------------------------------------------
54:
55: void __fastcall TForm2::Button1Click(TObject *Sender)
56: {
57:    Form2->Close();
58: }
59:
60: //---------------------------------------------------------------------------
```

```
61: // end of file Unit2.cpp
62:
```

```
 1:
 2: //***********************************************************
 3: //*                                                         *
 4: //* ------------------------------------------------------- *
 5: //*  Filename: Unit3.cpp                                    *
 6: //*  Part of:  ACDCStepping.exe                             *
 7: //*                                                         *
 8: //*  Compiler: Borland C++ Builder 6 servicepack 4          *
 9: //*            CPort VCL , JVCL                             *
10: //*  Made by:  Eric Halmans                                 *
11: //*  For:      Fontys Highschool Eindhoven, Mechatronica    *
12: //*  Date:     April 2006                                   *
13: //*  Version:  1.0 beta test version                        *
14: //*                                                         *
15: //*  Description:                                           *
16: //*    This file is part a windows program,                 *
17: //*    which is used for universal AC & DC motor control,   *
18: //*    with a HCS12 TBoard from Elektronic Laden            *
19: //*    (Hbridge with PWM on PP0 & PP1, SCI1 input,IIC Output)  *
20: //* ------------------------------------------------------- *
21: //*                                                         *
22: //***********************************************************
23:
24: #include <vcl.h>
25: #pragma hdrstop
26:
27: #ifndef Unit3H
28: #include "Unit3.h"
29: #endif
30:
31: #ifndef Unit1H
32: #include "Unit1.h"
33: #endif
34:
35:
36: //---------------------------------------------------------------------------
37: #pragma package(smart_init)
38: #pragma resource "*.dfm"
39: TForm3 *Form3;
40: //---------------------------------------------------------------------------
41: __fastcall TForm3::TForm3(TComponent* Owner)
42:         : TForm(Owner)
43: {
44: }
45:
46: //---------------------------------------------------------------------------
47:
48: // Short description:
49:
50: // Button to set CPort VCL Custom baudrate
51:
52: // Pre:  - CBaudRate->Text contains a value between 0..500.000
53: //       - CustomBaudRate
54:
55: // Post: The string value is translated in a integervalue and
56: //       is used to set a new CustomBaudRate
57: //       if buadrate is not within specs a non valid message is shown
58:
59: //---------------------------------------------------------------------------
60: void __fastcall TForm3::Button1Click(TObject *Sender)
```

```
61: {
62:  if ((StrToInt(Edit1->Text)>=1)&&(StrToInt(Edit1->Text)<=500000))
63:  {
64:     Form1->ComPort1->CustomBaudRate=StrToInt(Edit1->Text);
65:     Label2->Caption = Form1->ComPort1->CustomBaudRate;
66:  }
67:  else ShowMessage("    Not valid input for BaudRate     ");
68: }
69:
70: //---------------------------------------------------------------------------
71: // end of file Unit3.cpp
72:
```

```
 1: //---------------------------------------------------------------------------
 2:
 3: #include <vcl.h>
 4: #pragma hdrstop
 5:
 6: #include "Unit4.h"
 7: //---------------------------------------------------------------------------
 8: #pragma package(smart_init)
 9: #pragma resource "*.dfm"
10: TForm4 *Form4;
11: //---------------------------------------------------------------------------
12: __fastcall TForm4::TForm4(TComponent* Owner)
13:         : TForm(Owner)
14: {
15: }
16:
17: //---------------------------------------------------------------------------
18:
19: // Short description:
20:
21: // If this button is pressed, the Received Memo field is cleared
22:
23: // Pre:  Any Received Memo field content
24:
25: // Post: Received Memo field is cleared
26:
27: //---------------------------------------------------------------------------
28:
29: void __fastcall TForm4::Button1Click(TObject *Sender)
30: {
31:    AnsiString Str2="";
32:    int *PointSum=&Sum;
33:    int *PointTeller=&Teller;
34:    Memo1->Text = Str2;
35:    *PointSum=0;
36:    *PointTeller=0;
37:    Label3->Caption=*PointSum;
38:    Label4->Caption=*PointTeller;
39: }
40: //---------------------------------------------------------------------------
41:
```