

Software code

Terminal.exe

.h header files

Unit1.h

```

1:
2: /**
3: /**
4: /** ----- /**
5: /** Filename: Unit1.cpp /**
6: /** Part of: Terminal.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 Elektronik Laden /**
19: /** (Hbridge with PWM on PP0 & PP1, SCI1 input,IIC Output) /**
20: /** ----- /**
21: /** /**
22: /**
23:
24:
25: #ifndef Unit1H
26: #define Unit1H
27: /**-----/**
28: #include <Classes.hpp>
29: #include <Controls.hpp>
30: #include <StdCtrls.hpp>
31: #include <Forms.hpp>
32: #include "CPort.hpp"
33: #include <dstring.h>
34: #include <idtrivialftpbases.hpp>
35: /**-----/**
36: class TForm1 : public TForm
37: {
38: __published: // IDE-managed Components
39: TEdit *SendCharEdit;
40: TButton *Sended_Clear;
41: TButton *Connect;
42: TButton *Recieved_Clear;
43: TButton *Com_Dialog;
44: TComPort *ComPort;
45: TButton *SendChar;
46: TMemo *Sended_Memo;
47: TMemo *Received_Memo;
48: TEdit *CBaudRate;
49: TButton *Button1;
50: TLabel *CBaudMessage;
51: TLabel *CBaudSet;
52: TLabel *CBaudBitSec;
53: TButton *Button2;
54: TScrollBar *ScrollBar1;
55: TLabel *Label1;
56: void __fastcall ConnectClick(TObject *Sender);
57: void __fastcall Sended_ClearClick(TObject *Sender);
58: void __fastcall Recieved_ClearClick(TObject *Sender);
59: void __fastcall ComPortRxChar(TObject *Sender, int Count);
60: void __fastcall SendCharClick(TObject *Sender);

```

```
61:         void __fastcall Com_DialogClick(TObject *Sender);
62:         void __fastcall Button1Click(TObject *Sender);
63:         void __fastcall FormCreate(TObject *Sender);
64:         void __fastcall ScrollBar1Change(TObject *Sender);
65:         void __fastcall Button2Click(TObject *Sender);
66: private:    // User declarations
67: public:     // User declarations
68:         __fastcall TForm1(TComponent* Owner);
69: };
70: //-----
71: extern PACKAGE TForm1 *Form1;
72: //-----
73: #endif
74:
75: //-----
76: // end of file Unit1.h
77:
```

Software code

Terminal.exe

.cpp code files

Terminal.cpp
Unit1.cpp

```

1: //-----
2:
3: #include <vc1.h>
4: #pragma hdrstop
5: //-----
6: USEFORM("Unit1.cpp", Form1);
7: //-----
8: WINAPI WinMain(HINSTANCE, HINSTANCE, LPSTR, int)
9: {
10:     try
11:     {
12:         Application->Initialize();
13:         Application->CreateForm(__classid(TForm1), &Form1);
14:         Application->Run();
15:     }
16:     catch (Exception &exception)
17:     {
18:         Application->ShowException(&exception);
19:     }
20:     catch (...)
21:     {
22:         try
23:         {
24:             throw Exception("");
25:         }
26:         catch (Exception &exception)
27:         {
28:             Application->ShowException(&exception);
29:         }
30:     }
31:     return 0;
32: }
33: //-----
34:

```

```

1:
2: //*****
3: //*
4: //* -----*
5: //*  Filename: Unit1.cpp
6: //*  Part of:  Terminal.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 Elektronik Laden
19: //*    (Hbridge with PWM on PP0 & PP1, SCI1 input,IIC Output)
20: //* -----*
21: //*
22: //*****
23:
24:
25: #include <vcl.h>
26: #pragma hdrstop
27:
28: #include "Unit1.h"
29: //-----
30: #pragma package(smart_init)
31: #pragma link "CPort"
32: #pragma resource "*.dfm"
33:
34: bool Connected=false;
35:
36: TForm1 *Form1;
37:
38: //-----
39: __fastcall TForm1::TForm1(TComponent* Owner)
40:     : TForm(Owner)
41: {
42: }
43:
44:
45: //-----
46:
47: // Short description:
48:
49: // This menu option if checked opens a connection with the Comport VCL
50:
51: // Pre:  If Connect1 = Checked Comport is open
52: //       If Connect1 = not Checked Comport is closed
53:
54: // Post: If Connect1 was Checked Comport is now closed
55: //       If Connect1 was not Checked Comport is now open
56:
57: //-----
58: void __fastcall TForm1::ConnectClick(TObject *Sender)
59: {
60:     if (ComPort->Connected)

```

```

61:  {
62:      ComPort->Close();
63:      Connect->Caption="Disconnected";
64:      Connected=false;
65:  }
66:  else if (ComPort->Connected==0)
67:  {
68:      ComPort->Open();
69:      Connect->Caption="Connected";
70:      Connected=true;
71:  }
72: }
73:
74:
75: //-----
76:
77: // Short description:
78:
79: // If this button is pressed, the Send Memo field is cleared
80:
81: // Pre: Any Send Memo field content
82:
83: // Post: Send Memo field is cleared
84:
85: //-----
86: void __fastcall TForm1::Sended_ClearClick(TObject *Sender)
87: {
88:     AnsiString Str2="";
89:     Sended_Memo->Text = Str2;
90: }
91:
92:
93: //-----
94:
95: // Short description:
96:
97: // If this button is pressed, the Received Memo field is cleared
98:
99: // Pre: Any Received Memo field content
100:
101: // Post: Received Memo field is cleared
102:
103: //-----
104: void __fastcall TForm1::Recieved_ClearClick(TObject *Sender)
105: {
106:     AnsiString Str2="";
107:     Received_Memo->Text = Str2;
108: }
109:
110:
111: //-----
112:
113: // Short description:
114:
115: // If a character is received by Comport VCL, it will be displayed by this
116: // function as a hexadecimal value in the Received memo field
117:
118: // Pre: Any received character
119:
120: // Post: The received character is displayed as a hexadecimal value in

```

```

121: //      the Received Memo field
122:
123: //-----
124: void __fastcall TForm1::ComPortRxChar(TObject *Sender, int Count)
125: {
126:     AnsiString Str2;
127:     unsigned char a;
128:     unsigned char* Pa=&a;
129:
130:     ComPort->Read(Pa,Count);
131:     Received_Memo->Text = Received_Memo->Text + " " + IntToHex(a,2);
132: }
133:
134:
135: //-----
136:
137: // Short description:
138:
139: // If just one character is entered in the SendCharEdit field,
140: // it will be send by the ComPort and it will also show up in the
141: // Send Memo field
142:
143: // Pre: Any single character entered in the SendCharEdit field
144:
145: // Post: The single character will be send by the ComPort VCL and,
146: //      it is displayed as an hexadecimal value in the Send Memo field
147:
148: //-----
149: void __fastcall TForm1::SendCharClick(TObject *Sender)
150: {
151:     AnsiString Str;
152:     unsigned char Karakter;
153:     unsigned char* PKarakter=&Karakter;
154:     int Integer;
155:
156:     if (Connected)
157:     {
158:         if (SendCharEdit->Text.Length()==1)
159:         {
160:             Karakter = StrToWord(SendCharEdit->Text);
161:             ComPort->Write(PKarakter,1);
162:             //Integer = Karakter;
163:             Sended_Memo->Text = Sended_Memo->Text + " " + IntToHex(Karakter,2);
164:         }
165:         else
166:         {
167:             ShowMessage("      Just one char!      ");
168:         }
169:
170:     }
171:     else
172:     {
173:         ShowMessage("      Not Connected      ");
174:     }
175: }
176:
177:
178: //-----
179:
180: // Short description:

```



```

181:
182: // Choosing this menu option, a setup dialog for the Comport VCL is shown.
183: // In this dialog, you can change:
184: // - Port number
185: // - Baudrate
186: // - Data bits
187: // - Stop bits
188: // - Parity
189: // - Flow control
190:
191: // Pre: ComPort1
192:
193: // Post: setup dialog shown and changes effect ComPort1
194:
195: //-----
196: void __fastcall TForm1::Com_DialogClick(TObject *Sender)
197: {
198:     ComPort->ShowSetupDialog();
199: }
200:
201:
202: //-----
203:
204: // Short description:
205:
206: // Button to set CPort VCL Custom baudrate
207:
208: // Pre: - CBaudRate->Text contains a value between 0..500.000
209: //      - CustomBaudRate
210:
211: // Post: The string value is translated in a integervalue and
212: //        is used to set a new CustomBaudRate
213: //        if buadrate is not within specs a non valid message is shown
214:
215: //-----
216: void __fastcall TForm1::Button1Click(TObject *Sender)
217: {
218:     if ((StrToInt(CBaudRate->Text)>=1)&&(StrToInt(CBaudRate->Text)<=500000))
219:     {
220:         ComPort->CustomBaudRate=StrToInt(CBaudRate->Text);
221:         CBaudSet->Caption = ComPort->CustomBaudRate;
222:     }
223:     else ShowMessage("    Not valid input for BaudRate    ");
224: }
225:
226:
227: //-----
228:
229: // Short description:
230:
231: // When ever this program is started, this function is run first.
232: // At the moment it only sets the custom baudrate to 500.000
233:
234: // Pre: Program is not running
235: //
236: // Post: If program is started, CustomBaudRate is set to 500.000
237:
238: //-----
239: void __fastcall TForm1::FormCreate(TObject *Sender)
240: {

```

```

241:     ComPort->CustomBaudRate = 500000;
242: }
243:
244:
245:
246: //-----
247:
248: // Short description:
249:
250: // if the user manipulates the Scrollbar1, a label shows the translated
251: // scrollbar integer as a hexadecimal value (0..ff)
252:
253: // Pre:  previous scrollbar position
254: //
255: // Post: New scrollbar position integer,
256: //       and a new to hex translated integer shows up in Label1
257:
258: //-----
259: void __fastcall TForm1::ScrollBar1Change(TObject *Sender)
260: {
261:     Label1->Caption="0x "+IntToHex(ScrollBar1->Position,2);
262: }
263:
264:
265:
266:
267: //-----
268:
269: // Short description:
270:
271: // Scroll bar position is read out as a character.
272: // This character then will be send by the ComPort
273: // and it will also show up in the Send Memo field
274:
275: // Pre:  Scroll bar position
276:
277: // Post: The Scroll bar position is been translated in a single character
278: //       which will be send by the ComPort VCL and,
279: //       it is displayed as an hexadecimal value in the Send Memo field
280:
281: //-----
282: void __fastcall TForm1::Button2Click(TObject *Sender)
283: {
284:     AnsiString Str;
285:     unsigned char Karakter;
286:     unsigned char* PKarakter=&Karakter;
287:     int Integer;
288:
289:     if (Connected)
290:     {
291:         Karakter = (char)(ScrollBar1->Position);
292:         ComPort->Write(PKarakter,1);
293:         Sended_Memo->Text = Sended_Memo->Text + " " + IntToHex(Karakter,2);
294:     }
295:     else
296:     {
297:         ShowMessage("      Not Connected      ");
298:     }
299: }
300:

```