

Programmering C eksamen:

SWAGWAY DEBUGGER

af
Mathias Dannesbo

9. maj 2012

1 Form1.cs

```
41
42     /* Kaldes ved afskutning: lukker seriel-porte */
43     private void Form1_FormClosing(object sender,
44         FormClosingEventArgs e)
45     {
46         if (serialPort.IsOpen) // Luk seriel-porten hvis
47             den er åben
48         {
49             serialPort.Close();
50         }
51     }
52
53     1 using System;
54     2 using System.Collections.Generic;
55     3 using System.ComponentModel;
56     4 using System.Data;
57     5 using System.Drawing;
58     6 using System.Linq;
59     7 using System.Text;
60     8 using System.Windows.Forms;
61     9 using System.IO.Ports; // Seriel-porte
62    10 using System.Text.RegularExpressions; // Regular expressions
63    11 using System.Globalization; // Forskeld på , og .
64    12 using System.Windows.Forms.DataVisualization; // Grafer
65
66    13
67    14 namespace Rolling_graph
68    15 {
69    16         public partial class Form1 : Form
70    17         {
71    18             string readFromUART; // Holder data læst direkte fra
72                UART
73    19             string rxStringBuffer; // Holder rest af sidste pakke
74    20             List<string> rxListBuffer = new List<string>(); //
75                Holder pakker som liste
76
77    21
78    22             int packageCount = 0;
79    23             int oldPackageCount = 0;
80
81    24
82    25             public Form1()
83    26             {
84    27                 InitializeComponent();
85    28             }
```

Swagway Debugger

```
29
30          /*******/
31          /* Indstillinger */
32          /******'*****/
33
34          /* Kaldes ved opstart: finder seriel-porte */
35          private void Form1_Load(object sender, EventArgs e)
36          {
37              LoadSerialPorts(); // Opdater listen med
38              serialporte
39              cbSpeed.SelectedIndex = 10; // Sætter 115200 baud
40              som standardindstilling
41
42          }
43
44          /* Kaldes ved afskutning: lukker seriel-porte */
45          private void Form1_FormClosing(object sender,
46              FormClosingEventArgs e)
47          {
48              if (serialPort.IsOpen) // Luk seriel-porten hvis
49              den er åben
50              {
51                  serialPort.Close();
52              }
53          }
54
55          /* Finder systemets seriel-porte */
56          private void LoadSerialPorts()
57          {
58              cbComPort.Items.Clear(); // Ryd listen
59
60              foreach (string s in SerialPort.GetPortNames()) //
61              For alle serielporte i systemet
62              {
63                  cbComPort.Items.Add(s); // Føj til listen
64              }
65
66              if (cbComPort.Items.Count > 0) // Hvis der er
67              fundet serielporte
68              {
69                  cbComPort.SelectedIndex = 0; // Sætter den fø
70                  rste serielport som standardindstilling
71              }
72              else
73              {
74                  lbConnectionStatus.Text = "No COM-ports found";
75                  // Ellers advar i statuslinjen
76              }
77          }
78
79          /* Opdater listen med serielporte */
80          private void btReload_Click(object sender, EventArgs e)
81          {
82              LoadSerialPorts();
83          }
```

Swagway Debugger

```
75     }
76
77     /* Forbind og afbryd til serielporten */
78     private void btConnect_Click(object sender, EventArgs e
79     )
80     {
81         if (btConnect.Text == "Connect") // Hvis der skal
82             forbindes
83         {
84             if (!serialPort.IsOpen) // Hvis der ikke er
85                 forbundet
86             {
87                 serialPort.PortName = cbComPort.
88                     SelectedItem.ToString(); // Find navnet
89                     på porten
90                 serialPort.BaudRate = int.Parse(cbSpeed.
91                     SelectedItem.ToString()); // Find
92                     hastigheden
93                 serialPort.Open(); // Forbind
94             }
95
96             if (serialPort.IsOpen) // Hvis forbindelsen
97                 lykkedes
98             {
99                 lbConnectionStatus.Text = "Connected to: "
100                     + serialPort.PortName; // Skriv i
101                     statuslinjen
102                 btConnect.Text = "Disconnect"; // Lav
103                     knappen om til afbryd
104             }
105         }
106     else // Hvis der skal afbrydes
107     {
108         if (serialPort.IsOpen) // Hvis der er
109             forbindelse
110         {
111             serialPort.Close(); // Afbryd
112         }
113
114         lbConnectionStatus.Text = "Disconnected"; //
115             Skriv i status linjen
116         btConnect.Text = "Connect"; // Lav knappen om
117             til forbind
118     }
119 }
120
121 /* Længden af X-aksen på grafen */
122 private void udPackages_ValueChanged(object sender,
123     EventArgs e) // Bliver kaldt hver gang tallet æ
124     ndres
125 {
126     if (udPackages.Value <= 1000) // Hvis værdien er
127         under eller ligemed 1000 skal den ændres med
```

Swagway Debugger

```
112         100
113     {
114         udPackages.Increment = 100;
115     }
116     if (udPackages.Value > 1000) // Hvis værdien er
117         over 1000 skal den ændres med 1000
118     {
119         udPackages.Increment = 1000;
120     }
121     if (udPackages.Value == 1100) // Er værdien 1100
122         skal den være 2000
123     {
124         udPackages.Value = 2000;
125     }
126     chart1.ChartAreas.First().AxisX.Maximum = (double)
127         udPackages.Value; // Set X-aksens maksimum væ
128         rdi
129     }
130     /******
131     /* Seriel: modtag & monitor */
132     /******
133     /* Læs inkomende data til buffer og kald ReadToMonitor
134     () */
135     private void serialPort_DataReceived(object sender,
136         SerialDataReceivedEventArgs e)
137     {
138         readFromUART = serialPort.ReadExisting();
139         this.BeginInvoke(new EventHandler(ReadToMonitor));
140     }
141     /* Skriver rå data til monitor, ryderop i monitor og
142     kalder CleanData()*/
143     private void ReadToMonitor(object sender, EventArgs e)
144     {
145         tbMonitor.AppendText(readFromUART); // Tilføjer til
146         monitor
147
148         if (tbMonitor.TextLength > 50000) // Hvis der mere
149             end ca. 2000 pakker
150         {
151             tbMonitor.Lines = tbMonitor.Lines.Skip(20).
152                 ToArray(); // Slet 20 linjer
153         }
154         CleanData(readFromUART); // Kalder CleanData()
```

Swagway Debugger

```
155     rxStringBuffer += input; // Tilføjer nyt rå data
      fra UART
156     rxListBuffer = rxStringBuffer.Split('<').ToList();
      // Splitter ved hver begyndelse af ny pakke '<'
157     rxStringBuffer = rxListBuffer[rxListBuffer.Count()
      - 1]; // Ligger den sidste, ikke fuldstændige,
      pakke tilbage i buffer
158     rxListBuffer.Remove(rxListBuffer.Last()); //
      Sletter den ikke fuldstendige pakke fra listen
159
160     /* For alle nye pakker */
161     for (int i = 0; i < rxListBuffer.Count(); i++)
162     {
163         /* Slet alt efter slutningen af pakken '>' */
164         int j = rxListBuffer[i].IndexOf('>'); // Finder
            indexet af '>'
165         if (j > 0)
166         {
167             rxListBuffer[i] = rxListBuffer[i].Substring
                (0, j); // Klipper fra begyndelsen til
                indexet af '>'
168         }
169         /* End : Slet efter slutning*/
170
171         /* God pakke? Så plot den */
172         if (Regex.Match(rxListBuffer[i], @"^((-?\d
            {1,3}[.]\d\d,){3,3}\d{4,})$").Success) //
            Pakken undersøges om den passer ind.
173         {
174             string[] stringPackage = rxListBuffer[i].
                Split(',').ToArray(); // Pakken
                splittes til et string-array
175             float[] floatPackage = new float[4];
176
177             for (int k = 0; k < stringPackage.Length; k
                ++)) // String-arrayet parses til et
                float-array
178             {
179                 float.TryParse(stringPackage[k],
                    NumberStyles.Float, CultureInfo.
                    InvariantCulture, out floatPackage[
                    k]);
180                 /* TryParse prøver at parse, hvis det
                    ikke er muligt bliver pakken
                    droppet.
181                 * CultureInfo er nødvendigt for at
                    kunne bruge "." som decimaltegn */
182             }
183
184             packageCount++;
185             plotPackage(floatPackage); // Send float-
                arrayet til grafen
186         }
187         /* End : God pakke */
```

Swagway Debugger

```
188     }
189     /* End : For alle nye */
190 }
191
192     /******
193     /* Seriel: send */
194     /******
195
196 /* Sender ud-buffer til seriel-porten */
197 private void SendToSerial()
198 {
199     if (serialPort.IsOpen)
200     {
201         serialPort.Write(tbSend.Text); // Send ud-
202             bufferen
203         tbSend.Clear(); // Tøm ud-bufferen
204     }
205 }
206
207 /* Kalder SendToSerial hvis "Send"-knappen trykkes ned
208 */
209 private void btSendMon_Click(object sender, EventArgs e
210 )
211 {
212     SendToSerial();
213 }
214
215 /* Kalder SendToSerial hvis der trykkes enter i ud-
216 bufferen */
217 private void tbSend_KeyDown(object sender, KeyEventArgs
218 e) // Kaldes ved hver ny tastetryk
219 {
220     if (e.KeyCode == Keys.Enter) // Er tastetrykket
221         Enter?
222     {
223         SendToSerial();
224     }
225 }
226
227     /******
228     /* Graf */
229     /******
230
231 /* Tilføjer pakke til graf og rydder op */
232 private void plotPackage(float[] package)
233 {
234     for (int i = 0; i < package.Length-1; i++) // Plot
235         alle punkterne i pakken til hver sin serie
236 {
237     chart1.Series[i].Points.AddY(package[i]);
238 }
239
240 if (packageCount > udPackages.Value) // Slet alle
241     pakker over grænseværdi
242 {
```

Swagway Debugger

```
234         for (int i = 0; i < package.Length-1; i++)
235         {
236             chart1.Series[i].Points.RemoveAt(0);
237         }
238     }
239 }
240
241         /******
242         /* Statusbar */
243         /******
244
245         /* Udregn antal pakker per sekund */
246         private void timerSample_Tick(object sender, EventArgs
247         e) // Kaldes hvert sekund
248         {
249             int diff = packageCount - oldPackageCount; //
250                 Forskellen i antallet pakker siden sidste
251                 sekund
252             lbSamplesPerSec.Text = diff.ToString() + " Samples/
253                 sec"; // Skriv det på status linjen
254
255             if (diff>pbSamplesPerSec.Maximum) // Hvis
256                 forskellen er større en maksimum af statusbjæ
257                 lken
258             {
259                 pbSamplesPerSec.Maximum = diff; // Forstør
260                 makimum
261             }
262
263             pbSamplesPerSec.Value = diff; // Set statusbjælken
264                 til antallet af pakker per sekund
265
266             oldPackageCount = packageCount;
267         }
268     }
269 }
```