

Programmering C eksamen:

SWAGWAY DEBUGGER

af
Mathias Dannesbo

10. maj 2012

1 Indledning

Formålet med opgaven er at lave et stykke software der kan bruges sammen med forfatterens eksamenprojekt i Teknikfag A: El. Formålet med teknikfags projektet er at bygge en Segway klon, en motoriseret selvbalancerende tohjulet transportenhed. Nawnet på Segway klonen er Swagway. Under udviklingen af Swagway blev der behov for et stykke software til at teste et filter.

Kort beskrivelse af Swagway funktion

For at Swagway kan holde sig lodret skal den kende vinklen den er fra lodret. Det udregner den ved at læse data fra to sensorer: et accelerometer og et gyroskob.

Et gyroskob kan måle vinkelhastigheder. Det vil sige ændringer i vinklen. Hvis denne vinkelhastighed integreres over tid finder man vinklen gyroskopet har flyttet sig. Problemet med at integrer gyroskopdata er, at pga. måleusikkerheder vil det målte nulpunkt drive væk fra det fysiske nulpunkt.

Et accelerometer kan måle accelerationer. Man kan med hjælp fra tanges og data fra to akser udregne den vinkel accelerometeret står i forhold til jordens tyngdekraft. Problemet med dette er at accelerometer måler alle accelerationer, ikke kun jordens tyngdekraft. Når Swagwayen fx accelerer, bremser eller køre over en sten bliver den udregnet vinkel forkeret.

For at komme begge problemer til livs bliver begge sensorer brugt og data fra begge to bliver samlet i et såkaldt Kalman-filtret. Kalman-filtret kan tilnærmelses ses som et high-pass og low-pass filter. Det taget de hurtige ændringer (high-pass) fra gyroskop-dataene og bruger accelerometer-dataene til at holde det målte og det fysiske nulpunkt over ens.

Swagway Debugger

For at teste om Kalman-filtret er implementeret og tunet korrekt er det nødvendigt at sammenligne de tre vinkler fra hhv. gyroskopet, accelerometeret og kalmanfiltret. En måde at gøre dette på er ved at sende disse data fra elektronikken på Swagwayen til en PC og vise den grafisk. Det er her Swagway Debugger-softwaren kommer ind i billedet.

Swagway Debugger modtager data fra en Arduino mikrocontroller på Swagwayen over en seriel-port og skriver dataene på en graf. Som der står i produktbeskrivelsen:

Opgaven er at lave software der kan modtage data fra en Arduino over en serielport og visualiserer det. Softwaren er til test af filtereringen af data fra et gyroskop og accelerometer. Softwaren viser bl.a. et rullende plot, samt nogle viserinstrumenter.

2 Udførelse

Swagway Debugger er skrevet på Windows i C# som en Windows Form Application. Der er ingen eksterne afhængigheder; det bruger kun Microsoft biblioteker:

Fixme: Er dette biblioteker?

```
9 using System.IO.Ports; // Seriel-porte
10 using System.Text.RegularExpressions; // Regular expressions
11 using System.Globalization; // Forskel på , og .
12 using System.Windows.Forms.DataVisualization; // Grafer
```

Der er fem globale variabler:

```
18 string readFromUART; // Holder data læst direkte fra UART
19 rxStringBuffer; // Holder rest af sidste pakke
20 List<string> rxListBuffer = new List<string>(); // Holder
    pakker som liste
21
22 int packageCount = 0;
23 int oldPackageCount = 0;
```

De tre første er buffere der bliver brugt til midlertidig at gemme modtaget data i, de to sidste er tællere over antallet af pakker.

2.1 Indstillinger

Ved opstart af applikationen indlæses alle tilgængelige serial-porte og der sættes en standardindstilling for valget af seriel hastighed:

Fixme: Indsæt billed af "settings-tab"

```
34 /* Kaldes ved opstart: finder seriel-porte */
35 private void Form1_Load(object sender, EventArgs e)
36 {
37     LoadSerialPorts(); // Opdater listen med serialporte
38     cbSpeed.SelectedIndex = 10; // Sætter 115200 baud som
    standardindstilling
39 }
```

Ved afslutning af applikationen lukkes en eventuel åben seriel-port:

```
41 /* Kaldes ved afskutning: lukker seriel-porte */
42 private void Form1_FormClosing(object sender,
    FormClosingEventArgs e)
43 {
44     if (serialPort.IsOpen) // Luk seriel-porten hvis den er å
        ben
45     {
46         serialPort.Close();
47     }
48 }
```