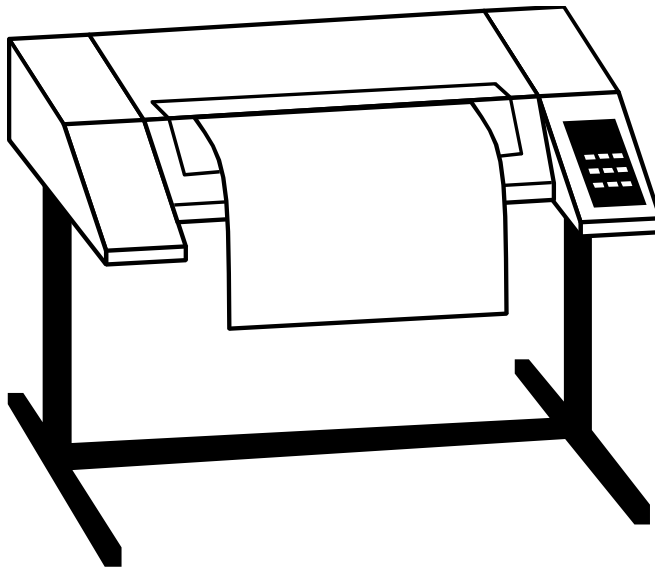

Übersichten und Notizen zu

ftPlot85CS

Ulrich Müller



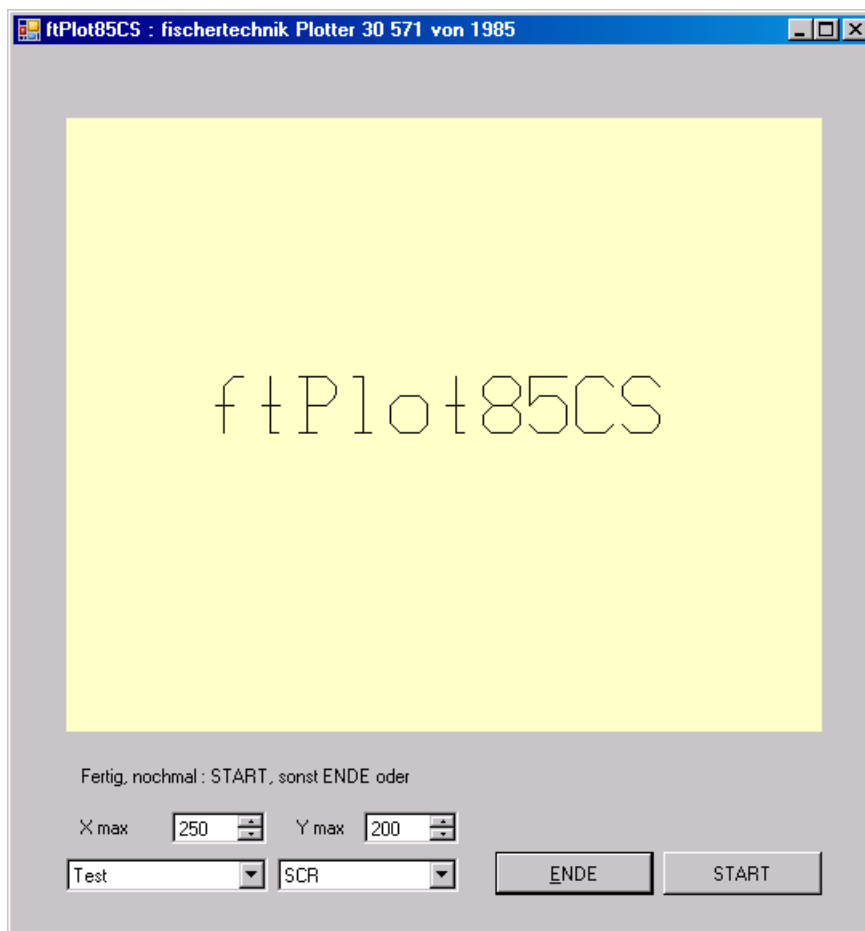
Inhaltsverzeichnis

Das Programm ftPlot85CS	3
Allgemeines	3
Referenz	4
Begriffe	4
Allgemeines	4
Anschluß des Plotters	4
Klasse FishPlot : FishStep : FishFace	5
Konstruktor	5
Eigenschaften	5
Methoden	5

Copyright Ulrich Müller. Dokumentname : ftPlot85CS.doc. Druckdatum : 27.10.2007

Das Programm ftPlot85CS

Allgemeines



Das Programm ftPlot85 ist eine Anwendung zum Betrieb des fischertechnik Plotters 30 571 von 1985. Der Plotter ist recht leistungsfähig und erfreut sich auch heute noch großer Beliebtheit. Es orientiert sich an dem von fischertechnik mitgelieferten GW Basic Programm, es wurde 1999 bereits auf VB6 umgestellt (www.ftcomputing.de/plot85.htm) Die vorliegende Fassung wurde in C# 2005 erstellt und beherzigt dessen objektorientiertes Konzept mit besonderem Vergnügen. Zentraler Punkt ist die Klasse FishPlot, die ihrerseits von der Klasse FishStep (Betrieb von Schrittmotoren) erbt. FishStep seinerseits beerbt die Klasse FishFace (Betrieb der fischertechnik Interface ROBO und Intelligent).

Die MainForm ermöglicht den Betrieb des Plotters und die Auswahl von vorliegenden Beispielanwendungen. Für Übersichtszwecke können die Graphiken zunächst auf der Form in einer PictureBox angezeigt werden (Auswahl SCR). Bei Auswahl USB bzw. COMx wird dann geplottet. Die Anzeige in der PictureBox läuft zur Orientierung mit. Die Größe der nutzbaren Zeichenfläche orientiert sich an den Möglichkeiten des eingesetzten Plotters. Sie kann über die UpDown-Boxen eingestellt werden, die PictureBox skaliert dann ihre Anzeige entsprechend.

Referenz

Begriffe

Blatt

Kontrollanzeige in einer PictureBox (picBlatt)

Nullpunkt links/oben,

Maßeinheit (float) Pixel ab Nullpunkt

Plotter

Bewegungsraum des Plotters

Nullpunkt Taster X/Y true (links/unten),

Maßeinheit (float) Steps ab Nullpunkt

max. **Fahrweg** wird vorgegeben MotCntl[0].maxPos und MotCntl[1].maxPos.

Zeichnung

Die zu erstellende Zeichnung, Sicht der Anwendung, Abbildung auf Plotter und Blatt

Nullpunkt : wählbar, **Maßeinheit** wählbar (**float**)

XNull, YNull, XLength, YLength, fakBlatt, fakPlotter

Allgemeines

Anschluß des Plotters

Anschlußbelegung

M1 Schrittmotor X-Achse (rot / schwarz) und Schrittmotor Y-Achse (grün / grau)

M2 Schrittmotor X-Achse (grün / grau)

M3 Schrittmotor Y-Achse (rot / schwarz)

M4 Stiftmagnet

I1 Endtaster X-Achse

I5 Endtaster Y-Achse

Details

Schrittmotor $7,5^\circ = 48$ Schritte pro Umdrehung.

Schnecke mit 4,75 mm Steigung.

Auflösung : Da beim Schrittmotor immer 4 Schritte zu einer Einheit zusammengefaßt werden. 0.4 mm

Klasse FishPlot : FishStep : FishFace

Konstruktor

FishPlot(int[,] MotList)

MotList : Liste der eingesetzten Schrittmotoren. Standard X-Achse an M1 (Endtaster an I1) und M2, Y-Achse an M3(Endtaster an I5) und M1. Zusätzlich wird der max. Fahrweg in Steps ab Endtaster für jede Achse als zweiter Parameter angegeben. Der Stift-Magnet ist z.Zt. fest an M4.

Aufbau des Plotters entsprechend Plotter 85

Beispiel :

```
private FishPlot fp = new FishPlot(new int[,] {{1, 500}{3, 400}});  
fp.Openinterface(IfTypen.ftROBO_first_USB, 0);  
....  
fp.CloseInterface();
```

Stepmotor X-Achse an M1 und M2 (mit Endtaster an I1) und einem Fahrweg von 500 Steps ab Endtaster. Y-Achse an M3 / I5, 400 Steps. Anschließend muß ein OpenInterface erfolgen um die Verbindung zum Interface herzustellen. Hier dem ersten ROBO Intetrface an USB.

FishPlot(int[,] MotList, PictureBox Blatt)

Wie oben, zusätzlich erfolgt eine Ausgabe in die angegebenen PictureBox. Sinnvoll zur Kontrolle des Plotvorganges.

FishPlot(PictureBox Blatt)

Ausgabe nur in die angegebenen PictureBox. Maßeinheit : Pixel, Koordinaten-Null : linke, untere Ecke, Zeichenfläche : Die aktuelle PictureBox

Sinnvoll in der Testphase.

Eigenschaften

StiftMotor

Bestimmen des Stiftmotors (default = 4)

Methoden

OpenInterface(IFTypen PortName)

Erstellen einer Verbindung zum Interface.
Erbteil der Basisklasse FishFace siehe dort.

CloseInterface()

Schließen der Verbindung zum Interface (Erbteil Basisklasse FishFace)

DriveHome()

Anfahren des Plotter Nullpunktes mit gehobener Feder. Das heißt anfahren der zugehörigen Endtaster und "Freifahren" in Gegenrichtung um zwei Steps.

DriveOrigin()

Anfahren des Blatt Nullpunktes mit gehobener Feder. Standardmäßig ist das die linke untere Ecke des Plotters, also die gleiche Position wie die durch DriveHome erreichte.

Die Origin kann durch SetOrigin, SetScale und DrawAxis verändert werden.

SetOrigin(PointF NewPos)

Setzen eines neuen Koordinaten-Nullpunktes bezogen auf den aktuellen Nullpunkt bei unverändertem Maßstab der Zeichnung.

SetScale(PointF min, PointF max)

Setzen von Maßstab und Koordinaten Nullpunkt der Zeichnung. min enthält die Werte der linken unteren Ecke, max die der oberen rechten Ecke, der Nullpunkt liegt entsprechend.

DrawAxis(PointF min, PointF max)

Setzen von Maßstab und Koordinaten-Nullpunkt wie bei SetScale. Zusätzlich : Zeichnen eines Koordinatensystems (z.Zt. Koordinatenkreuzes).

DrawBox

DrawBox(PointF minEcke, PointF maxEcke)

Zeichnen eines Rechteckes. Koordinaten bzogen auf den Nullpunkt.

DrawBox(PointF minEcke, float width, float height)

Zeichnen eines Rechteckes, Koordinaten linke untere Ecke bezogen auf den Nullpunkt.

DrawBoxDelta(PointF minEcke, float width, float height)

Zeichnen eines Rechteckes, Koordinaten linke untere Ecke relativ.

DrawCircle

DrawCircle(PointF Center, PointF Radius, float anfWinkel, float endWinkel)

Zeichnen einer Ellipse mit Mittelpunkt, bezogen auf den Nullpunkt, bei Center und den Radien in X und Y (wenn sie gleich sind wird ein Kreis gezeichnet). Volle Figur bei Winkel 0, 360 Grad.

DrawCircle(PointF Center, float Radius)

Zeichnen eines Kreises mit Mittelpunkt Center und Radius.

DrawLine

DrawLine(PointF NewPos) alternativ **DrawLine**(float X, float Y)

Zeichnen einer Linie ab aktueller Position zu den angegebenen Koordinaten

DrawLineDelta

DrawLineDelta(PointF DeltaPos) alternativ **DrawLineDelta**(float X, float Y)

Zeichnen einer Linie ab aktueller Position um Increment

DrawLines

DrawLines(PointF[] punktListe)

Zeichnen mehrerer an einander anschließender Linien durch Vorgabe ihrer Koordinaten bezogen auf ihren Nullpunkt.

DrawLinesDelta

DrawLinesDelta(PointF[] punktListe)

Zeichnen mehrere an einander anschließender Linien durch Vorgabe ihrer relativen Koordinaten

PositionTo

PositionTo(PointF NewPos) alternativ **PositionTo**(float X, float Y)

Fahren mit gehobener Feder zu der angegebenen Koodinate

PositionDelta

PositionDelta(PointF DeltaPos) alternativ **PositionDelta**(float X, float Y)

Verfahren mit gehobener Feder um Increment

DrawText

DrawText(PointF startPos, string textString, int Richtung, int Groesse)

Zeichnen eines textStrings, beginnend ab startPos (linke, untere Ecke des ersten Zeichens) bezogen auf den Nullpunkt. Ausrichtung 0 : waagrecht, 1 : senkrecht, 2 und 3 : gespiegelt. Skaliert in relativer Groesse von 1 – 5.

DrawText(PointF startPos, string textString)

Zeichnen eines textString ab startPos, waagrecht, Groesse 2

DrawTextDelta

DrawText(PointF startPos, string textString, int Richtung, int Grosse)

Zeichnen eines textStrings wie DrawText aber mit rel. Angabe der Startposition

DrawText(PoitnF startPost, string textString)

Zeichnen eines textStrings abe startPos (rel.) waagrecht, Groesse 2