

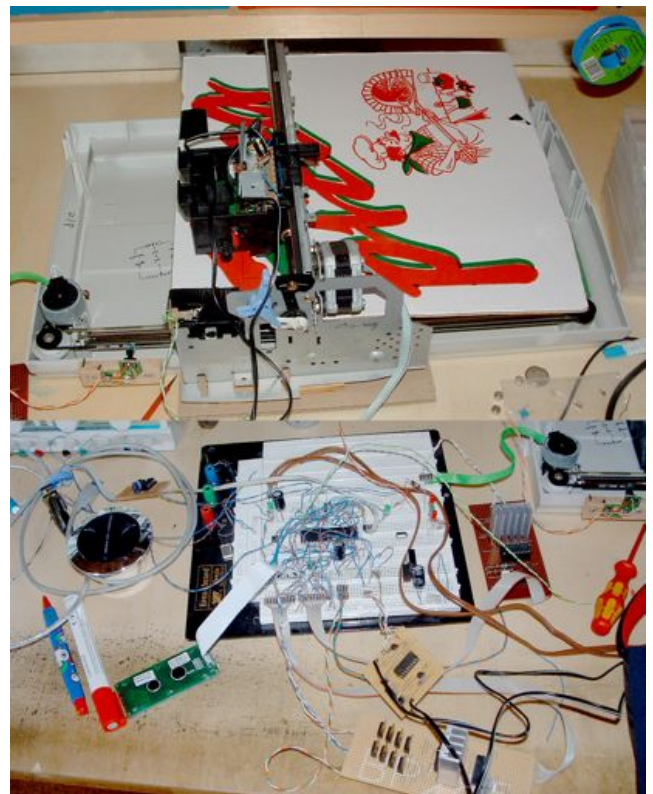
Plotter

Ein Projekt von Benedikt Seidl
Benedikt.Seidl@gmx.de

Inhaltsverzeichnis

1 Aufbau	1
1.1 X-Achse (Scanner)	2
1.1.1 Endschalter	2
1.2 Y-Achse (Drucker)	2
1.2.1 Endschalter	3
1.3 Z-Achse (Diskettenlaufwerk)	3
2 Elektronik	3
2.1 Schrittmotorensteuerung L297/L298	3
2.2 Schrittmotorensteuerung L293	4
2.3 Pegelwandler mit MAX232N	4
2.4 Prozessor und Verbindungen	5
3 Software	5
3.1 Protokoll	5
3.2 PIC	5
3.3 Mac	5
3.3.1 1ps.rev	6
3.3.2 2zeichen.rev	6
3.3.3 3cnc_sof.rev	6
4 Ergebnisse	6
4.1 Fazit	7
5 Kontakt	7

1 Aufbau



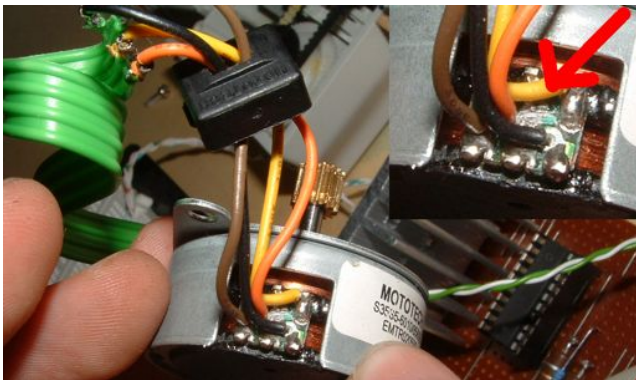
Ein lang gehegter Wunsch von mir war es einen Plotter zu bauen, schon vor Jahren versuchte ich mithilfe von DC-Motoren und Gewindestangen einen Linearantrieb für einen Plotter zu bauen. Da ich jedoch mit der altersbedingten Ungenauigkeit arbeitete, verwarf ich die Idee wieder. Dennoch konnte ich mich von der Idee nicht losbringen. Als dann mein Vater seinen alten Computer Schrott wegschmeißen wollte, kam mir die Idee: Warum aufwändig die Mechanik bauen, wenn man sie auch einfach aus einem Drucker bzw. Scanner ausbauen kann? Also schraubte ich vorsichtig den alten Scanner und Drucker auf. Dort wo früher die Tintenpatronen waren ist nun der Stift angebracht, Die Belichtungseinheit des Scanners wurde entfernt (schöne kleine Neonröhre, die mit 12V bei 0,5A hell macht) und stattdessen dort, wo früher der CCD-Sensor war, eine Platte montiert,

auf der dann das zu bearbeitende Objekt liegt.

1.1 X-Achse (Scanner)

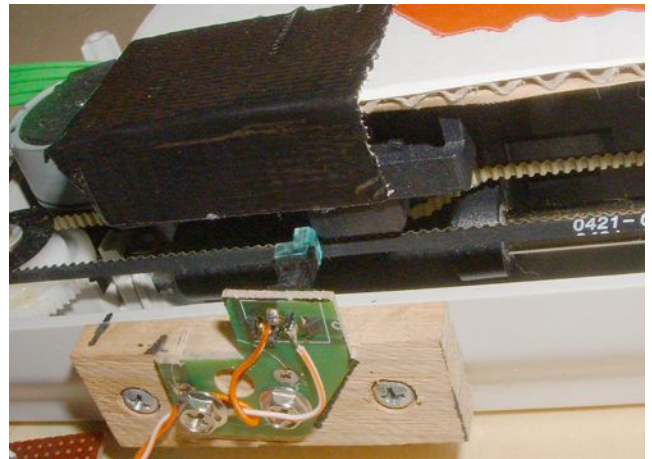


Wie bereits erwähnt, bewegt die Scanner Mechanik nicht den Stift, sondern das Papier unter dem Stift. Deshalb waren hier fast keine Anpassungen nötig. Das große Problem, wie auch beim Drucker-Schrittmotor war herauszufinden welche Leitung zu welcher Spule gehört. Dafür empfiehlt es sich den Motor auszubauen, da man, wegen der hohen Übersetzung des Motors, nicht erkennen kann ob er rund läuft.



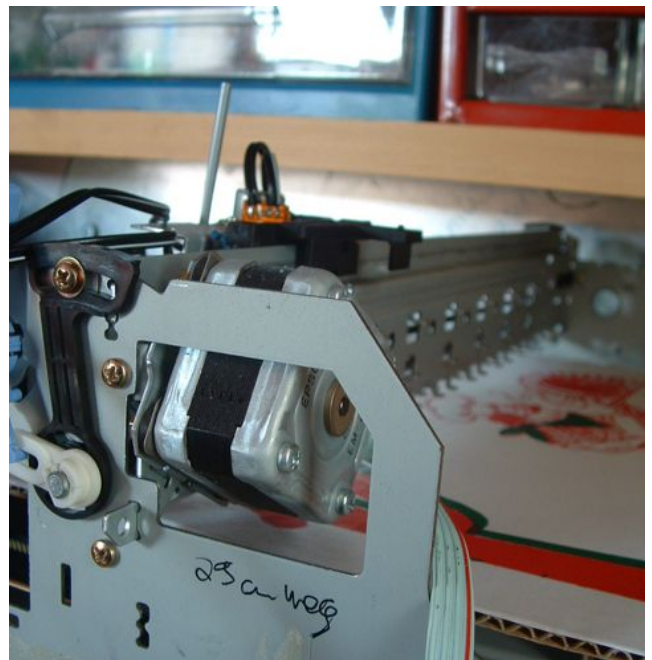
Ursprünglich war der Motor des Scanners ein unipolar Motor, mit nur einen Common Leitung. Glücklicherweise konnte man eine Gummi-Abdeckung abnehmen, unter der eine kleine Platine zu erkennen war, die auf jeder Seite 3 Löt pads hatte, welche mit dünnen Kupferleitungen auf der einen Seite und auf der anderen mit den Kabeln zur Ansteuerung des Motors auf der anderen Seite verbunden waren. Die mittleren beiden Löt pads waren miteinander und mit der Common Leitung verbunden. Die Verbindung untereinander trennte ich indem ich die Leiterbahn mit einem Schraubenzieher unterbrachte. Die Common Leitung wurde entfernt.

1.1.1 Endschalter



Der Endschalter ist aus einem alten CD-Player, den mir freundlicherweise der Hifi-Laden an der Ecke geschenkt hat. Der Endschalter wurde dann mit der Laubsäge (es gibt für die Laubsäge auch Sägeblätter speziell für Eisen, mit denen geht das ganz gut) so ausgeschnitten, dass ich gleich die passende Halterung habe. Danach musste das ganze noch auch einen kleinen Holz-sockel montiert werden, da sonst der Schalter an dem Zahnriemen schleifen würde.

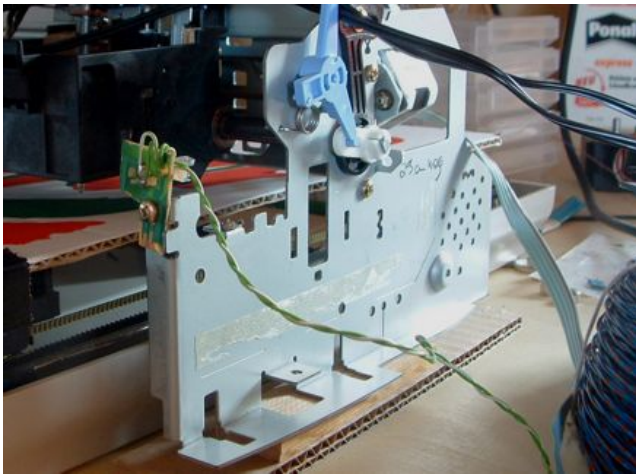
1.2 Y-Achse (Drucker)



Großer Vorteil bei der Drucker Mechanik war, dass es sich die Lineareinheit mit den passenden Füßen ausbauen ließ. Man hätte es direkt mit zwei Abstandhaltern auf eine Platte schrauben

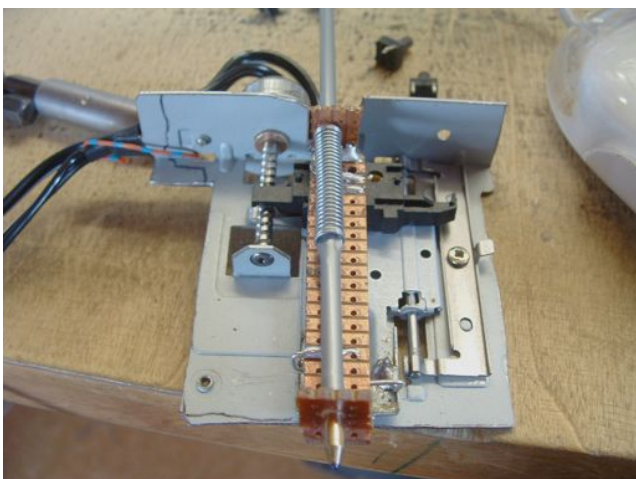
können, und schön wäre die Y-Achse montiert gewesen.

1.2.1 Endschalter



Auch bei der X-Einheit wurde nicht der originale Endschalter in Form einer Gabellichtschranke verwendet, sondern ein Endschalter aus einem CD-Player verwendet. Das Problem im fliegenden Aufbau ist jedoch, dass, wenn sich der Schrittmotor dreht, die gesamte Konstruktion vibriert, dadurch wird der Endschalter zu früh ausgelöst. Da jedoch der Endschalter nur am Anfang nach einem Reset benötigt wird, kann man einfach den Reset noch mal ausführen.

1.3 Z-Achse (Diskettenlaufwerk)



Die Z-Achse, dient eigentlich nur dazu den Stift auf und ab zu setzen, d.h. sie kann nicht stufenlos angesteuert werden. (Wäre aber technisch schon möglich). Als mechanische Grundlage dient hier die Mechanik, die in einem Diskettenlaufwerk den Schreib/Lese-Kopf bewegt hat.

Darauf wurde, mit den original Schrauben, ein Stück Leiterplatte geschraubt, welche weiter unten auch noch an die Mechanik des Laufwerks festgelötet wurde. Auf die Leiterplatte wurde dann wiederum im 90° Winkel weitere kleine Stücke Leiterplatte gelötet, welche aber zuvor angebohrt wurden. Dadurch ergibt sich eine mehr oder weniger genaue Führung für die Kugelschreibermine, welche mit einer Feder nach unten gedrückt wird.

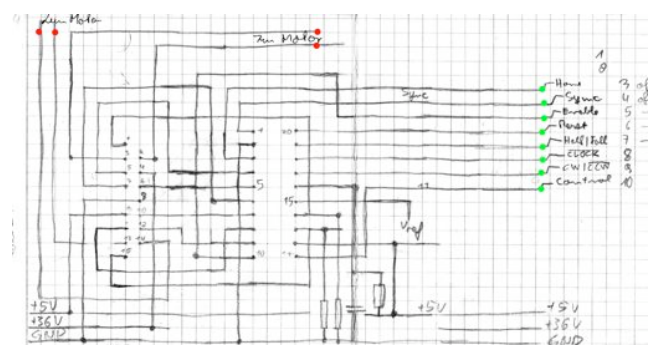
Leider ergibt sich durch die Kugelschreibermine ein kleines Problem, da diese öfter streift, besonders, wenn der Plotter einige Tage gestanden ist.

2 Elektronik

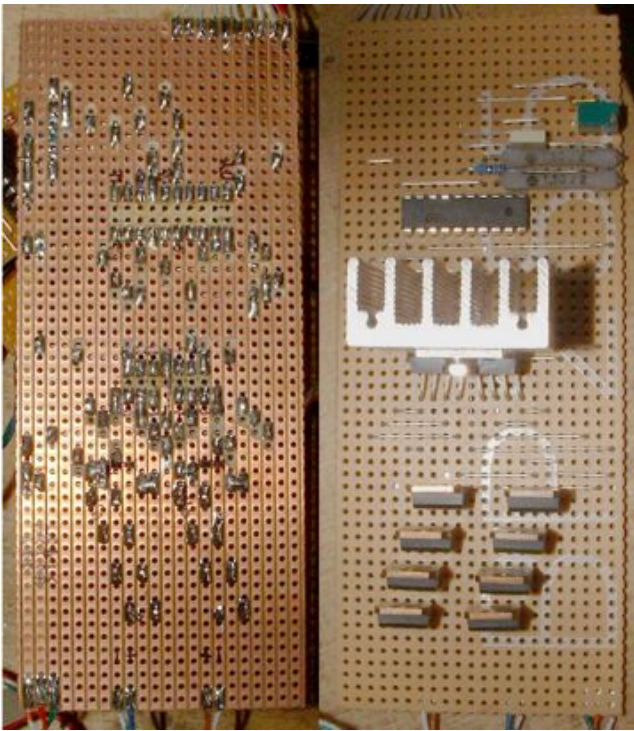
Die Elektronik ist komplett auf Streifenrasterplatinen aufgebaut. Ätzen ist mir zu aufwändig (vor allem fehlt da auch die Software am Mac). Zwar dauert das löten mit Streifenrasterplatinen auch relativ lange, aber bis jetzt hatte ich noch keine Probleme damit.

2.1 Schrittmotorensteuerung L297/L298

Die klassische Schrittmotorensteuerung kommt auch bei mir zum Einsatz: (Eingänge sind grün, die Ausgänge rot markiert.)



Das ganze gelötet sieht so aus:



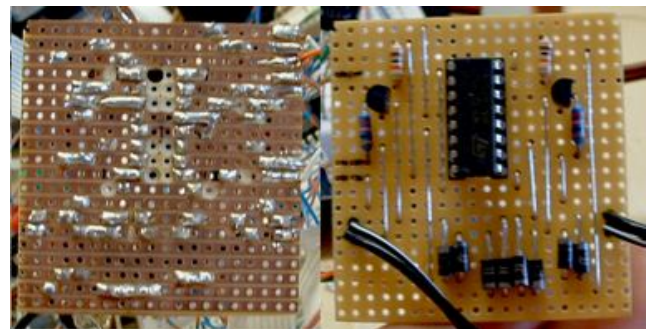
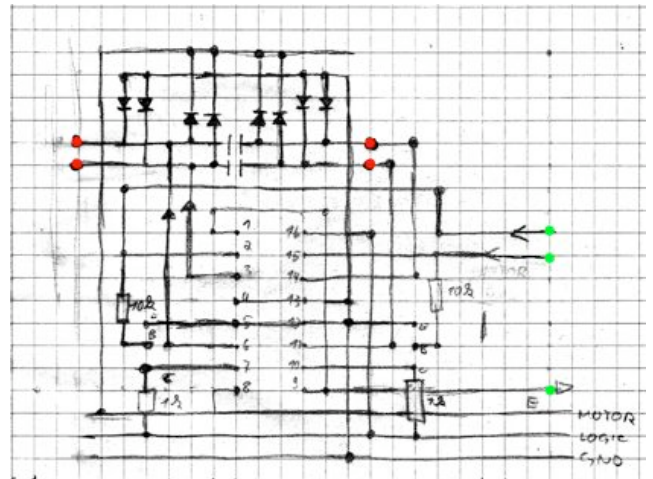
Der Kühlkörper ist aus einer alten unterbrechungsfreien Stromversorgung, welches das Gymnasium, an dem ich war, entsorgt hat.

2.2 Schrittmotorensteuerung L293

Da der Aufwand die L227/L298 Schaltung zu bauen rel. groß ist, entschied ich mich, den Schrittmotor des Diskettenlaufwerkes mit einer einfacheren Schaltung aufzubauen. Ich verwendete einen L293, da der noch zu Hause rumflog. Zum Ansteuern eines bipolaren Schrittmotors muss man so vorgehen:

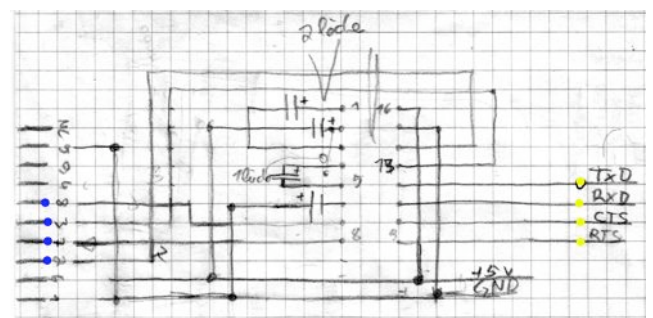
	A1	A2	B1	B2
Schritt 1	+	-	+	-
Schritt 2	+	-	-	+
Schritt 3	-	+	-	+
Schritt 4	-	+	+	-

Wie man sieht, sind die Spulen A und B immer an, und werden nur verpolt. Deshalb habe ich mithilfe von BC338 Transistoren einen Inverter aufgebaut, sodass ich nur 3 Ports (zwei für die Inverter, der dritte für die Enable Eingang) für die Ansteuerung des Schrittmotors brauche. Allerdings ist dafür ein größerer Software Aufwand nötig als bei der L227/L298 Schaltung, weil der PIC jeweils wissen muss welche Spule wie rum gepolt wird.

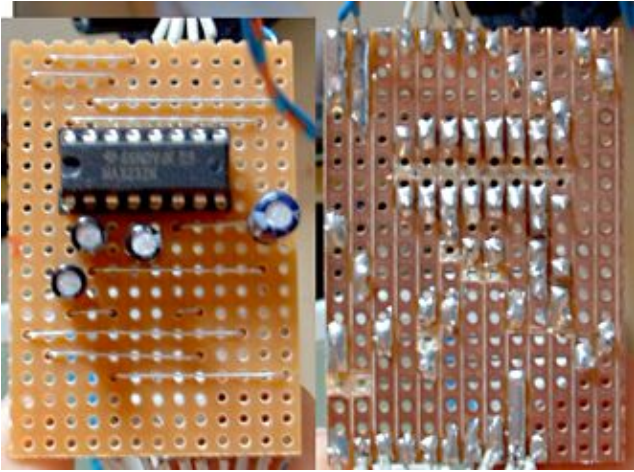


2.3 Pegelwandler mit MAX232

Da der normale Serielle Anschluss am Computer mit $\pm 12V$ läuft, der PIC aber nur 0 oder 5 V versteht, muss ein Pegelwandler zwischen PC und PIC. Die Standard-Schaltung hierbei ist mit einem MAX232 aufgebaut. Die ELKOs sind $1\mu F$, dass der eine größer ist als die anderen 3 hat keinen bestimmten Grund, ich hatte einfach keinen anderen mehr zur Hand.

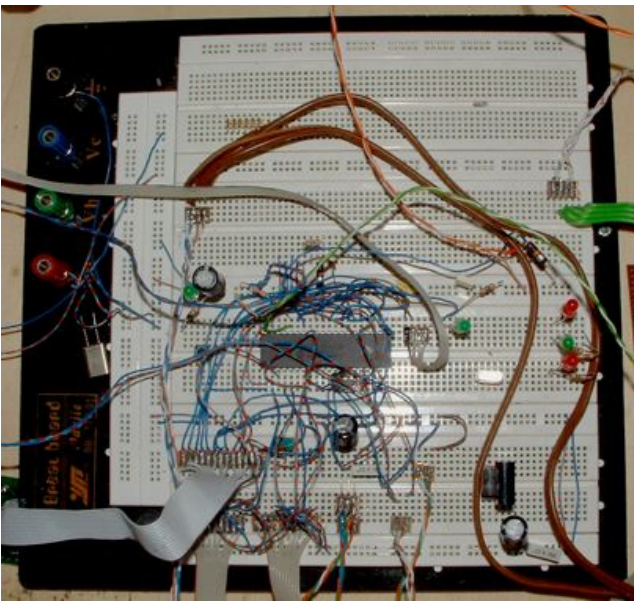


Auf dem „Schaltplan“ sind die Anschlüsse in Richtung PC blau, und in Richtung PIC gelb markiert.

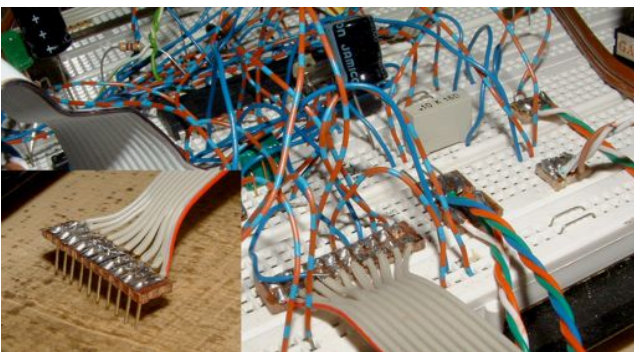


2.4 Prozessor und Verbindungen

Dieser Teil der Schaltung wurde nicht gelötet, sondern nur auf einem Steckbrett gesteckt. Dies hat den Vorteil, dass die Schaltung ohne Probleme geändert werden kann.



Die Verbindungen sind von mit selbst gebastelte Steckerleisten.



Ein kleines Stück Streifenrasterplatine, auf der dicke Silberdraht Stücke gelötet wurden. So ist ein

sicherer Kontakt gewährleistet. Einzige das „ausstecken“ ist etwas schwieriger, da werden meistens ein bis zwei Pins verbogen, welche sich aber leicht wieder gerade richten lassen.

3 Software

3.1 Protokoll

Der PIC erhält vom Computer insgesamt 5 bytes je Datensatz, wobei das erste byte den Befehl, und jeweils 2 bytes die neue X bzw. Y-Position darstellt. Soll der Z motor auf bewegt werden, wird als Befehl b'00000001' gesendet, soll er sich ab bewegen wird b'00000011' gesendet. Bei einer Positionsänderung wird als Befehl b'00000000' verwendet.

3.2 PIC

Wie man bereits am Protokoll erkennen kann, ist die Auflösung der Motoren so hoch, dass mit 16bit also 2 8bit Registern gearbeitet werden muss. Dies ist mit einem 8bit PIC aufwändig und umständlich. Nicht desto trotz ist es mir gelungen, den Bresenham-Algorithmus zum Rendern von Linien in den PIC zu packen.

Des weiteren muss der PIC noch ein Zweites Problem meistern: Ein Schritt bei der Y-Achse entspricht von der Länge her nicht einem Schritt bei der X-Achse. Dies wird durch die Software auf dem PIC kompensiert.

Die Aktuelle Position, sowie irgendwelche Fehler bei der seriellen Kommunikation werden auf einem kleinen 2 Zeiligen Display angezeigt.

Das Hauptprogramm heißt cnc.asm. lcd.asm wird vom Hauptprogramm eingebunden.

3.3 Mac

Die Software am Mac wurde komplett mit Runtime Revolution geschrieben. Sie ist darauf ausgelegt, ein PostScript, welches mit Freehand und dem „als Postscript sichern unter“-Druckerdialog von Mac OSX erstellt wurde in Befehle umzuwandeln, welche dann an den PIC geschickt werden können. Zum benutzen der Programme muss

man Runtime Revolution besitzen, oder eine zeitlich begrenzte Trail herunterladen.

3.3.1 1ps.rev

Mit diesem Programm wird aus dem PostScript eine Grafik gemacht, welche von Runtime Revolution angezeigt werden kann. Dabei werden die Bezierkuven, welche in der PostScript enthalten sind in Punkte bzw. Linien „gerastert“

3.3.2 2zeichen.rev

Die Grafik wird aus dem ersten Stack kopiert und in den Stack „2zeichen.rev“ eingefügt. Nach dem Klicken auf den „cnc“-Button wird die Grafik in Informationen umgerechnet, die das dritte Programm dann zum PIC schickt. Dafür werden die Informationen in das rechte Text Feld vom Stack „3cnc_soft.rev“ kopiert.

3.3.3 3cnc_sof.rev

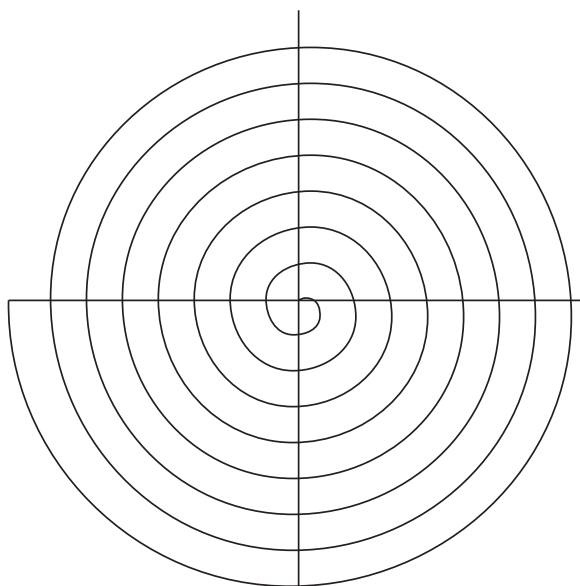


Der Stack „3cnc_soft.rev“ basiert auf dem Stack „SerialTest.rev“ von Sarah Reichelt.

Nach dem öffnen des Stacks muss zuerst ein Port ausgewählt werden. Danach muss die serielle Verbindung initialisiert werden, die geschieht mit einem Klick auf „Open Serial Port“. Mit den Feldern und Buttons rechts oben kann ein einzelner Befehl an den PIC geschickt werden. In das rechte Textfeld werden die Daten aus dem vorherigen Stack kopiert, und danach auf den unteren „Button“-Button geklickt. Im mittleren Textfeld wird die Kommunikation zwischen dem Mac und dem PIC angezeigt. Sobald der PIC Daten empfängt schickt er ein X-OFF, also ein 19 zum Computer.

Nachdem er den Befehl ausgeführt hat, schickt er eine 17 zum Computer. Daraufhin beginnt dieser mit dem nächsten Datensatz.

4 Ergebnisse



Oben ist die Freehand-Datei zu sehen, unten das vom Plotter gezeichnete. Bei den geraden Linien ist er erstaunlich schnell, bei den Rundungen, z.B. beim P oder R bremst die Kommunikation mit dem Computer den PIC aus, sodass es ziemlich langsam voran geht.

Noch einige Anmerkungen:

- die kleinen Striche beim inneren des Rs oder Ps sind nicht geplottet, sondern kommen durch das auf und absetzen des Stiftes zustande, der leicht schräg montiert ist
- die Linien des Fadenkreuzes stehen nicht genau im 90° Winkel aufeinander, das liegt daran, dass der Scanner und der Drucker nicht genau zueinander ausgerichtet ist
- die erkennbaren Stufen beim O könnten etwas herausgemacht werden, indem man die Software anders konfiguriert (die Bezier-Linien müssten mit einer höheren Auflösung gerastert werden)
- einige Ungenauigkeiten könnte man entfernen, wenn man eine andere (ebenere) Unterlage als die ehemalige Pizza-Schachtel verwenden würde.

4.1 Fazit

Eigentlich wollte ich diese „Dokumentation“ schreiben, dass ich auch noch nach dem Zerlegen weiß wie das früher mal war. Nachdem ich jetzt mal wieder gesehen habe, wie er plottet bin ich mir nicht mehr sicher, ob ich ihn wirklich zerlegen will, oder doch versuchen werde, einen Schneideplotter daraus zu bauen.

An dem Konzept aus vorhanden Drucker oder Scannern einen Plotter zu bauen gibt es folgende Punkte zu beachten

- Der Motor des Druckers ist zu schwach. Man kann ihn rel. einfach mit der Hand aufhalten. Dagegen ist der Scanner fast nicht

aufzuhalten, was wohl an dem Getriebe liegt.

- Man sollte sich zwei Baugleiche Scanner/Drucker besorgen. Dadurch ist zwar der mechanische Aufwand etwas größer, aber dafür muss man nicht die unterschiedlichen Schrittlängen anpassen
- Die Kommunikation über den COM-Port ist zu langsam. Mit RunRev kann ich max. 57,6 kbaud einstellen. Ich weiß es nicht genau, evtl. liegt es auch an meinem Programm, dass die Kommunikation so langsam ist. Eventuell könnte man sich auch überlegen die Daten zuerst in ein Eeprom zu speichern, und dann zu Plotten, Das würde zumindest vom Plott-Vorgang auf besser aussehen
- Ein Kugelschreiber ist kein idealer Plotterstift (keine sauberen Linien, zu viel Druck notwendig)
- Ein Kugelschreiber ist ein idealer Plotterstift (muss nicht verschlossen werden, lässt keine Tinte ausströmen, wenn er an einem Punkt steht, ist robust)

Dennoch finde ich, dass es ein schönes Spielzeug ist (in diesem Stadium ist es wohl wirklich nicht mehr), und es sehr viel Spaß macht ihm beim Plotten zuzusehen.

5 Kontakt

Benedikt Seidl
Lerchenstraße 15
91177 Thalmässing
Benedikt.Seidl@gmx.de
aim: BenediktSeidl
icq: #262474181
msn: benediktseidl@hotmail.com
jabber: benedikt.seidl@gmx.de