

Inhaltsverzeichnis:

	Seite:
1. Allgemein	3
2. Mein Projekt	4
3. Konzept	5
3.1 Mechanik	5
3.2 Elektronik	5
3.3 Software	7
4. Mechanik im Detail	8
4.1 Allgemein	8
4.2 X-Achse	8
4.3 Y-Achse	9
4.4 Z-Achse	10
4.5 Kalibrierung	10
5. Elektronik im Detail	11
5.1 Allgemein	11
5.2 Schrittmotorkarte	11
5.2.1 Die IC-Kombination L297/ L298	11
5.2.1.1 L297	12
o Halfstep/Fullstep	12
o Wahl der Betriebsart	13
o Stromchopper	14
o Boost/Sleep- Signal	14
o Shaping-Mode	14
5.2.1.2 L298	15
o Phase Chopping	16
o Inhibit Chopping	16
o Auswahl des	17
Choppverfahren	17
o Auswahl der FR-Dioden	17
5.2.2 Temperaturüberlastschutz	17
5.2.3 Referenzspannungsaufbereitung	18
5.2.4 Bohrer u. Absaugsignal	18
5.2.5 Spannungsversorgung	18
5.3 PC- Interface	19
5.4 Plotter- Ausgabe	19
5.5 Busplatine	20
5.6 Zusatzplatine	20
5.7 Computer- Netzteil	20
5.8 42V- Netzteil	20
6. Messungen an der Schrittmotorkarte	22
6.1 Schrittgeschwindigkeit	22
6.2 Messung derReferenzspannung	22
6.3 Messung desTemperaturüberlastschutzes	22
6.4 Messung der Brückenspannung	22
7. Software im Detail	23
7.1 Allgemein	23
7.2 Meine Software BPSoft1	23
7.3 Programmiersprache C++ Builder	24
7.4 Parallel Komponent	25
7.5 Grundlagen zur Parallelen Schnittstelle	25
8. Zeitplan	28
9. Kostenrechnung	30
10. Schlusswort	31
Notizen:	32

Anhang 1:

Schrittmotorkarte:	Schaltplan	36
	Berechnung	40
	Bestückungsplan	43
	Layout	44
	Stückliste	46
PC- Interface	Schaltplan	47
	Berechnung	48
	Bestückungsplan	49
	Layout	50
	Stückliste	52
Plotter- Ausgabe	Schaltplan	53
	Berechnung	54
	Bestückungsplan	55
	Layout	56
	Stückliste	58
Backplane	Schaltplan	59
	Bestückungsplan	60
Zusatzplatine	Layout	61
	Schaltplan	63
Pinbelegung	Parallele Schnittstelle	64
	Centronics Schnittstelle	65
	Bussystem	66
	Multiplex- Statusanzeige	67
Messungen des Phasenstroms Vom Schrittmotor im Chopper- Betrieb	1kHz/42V	68
	500Hz/30V	69
	1kHz/30V	70
	2,7kHz/30V	71
	100Hz/12V	72
	1kHz/12V	73
Vergleich des Stromanstiegs bei einer Brückenspannung von 12V und von 42V		74
Mechanik- Zeichnungen	Linearführung MLF1 u. Wellenschlitten	75
	Profilaufnahme	76
	Printbohrmaschinenaufnahme	77
	Lagerblock	78
	Motorlagerblock	79
	Gewindemutterblock	80
	Rechteckprofil	81
Softwarelisting	Programm	82
	Strukturgramm Sortieren	93
	Strukturgramm Fehlerkorrektur	95
Anhang 2:	Datenblätter:	
	L297	
	L298	
	The L297 Stepper Motor Controller (Application Note)	
	Stepper Motor Considerations Common Problems & Solutions	
	Stepper Motor Driving	
	BYV27/100	
	74LS244	
	74LS125	
	74LS08	
	74LS02	
	74LS14	
	MAX232	
BC337		

1. Allgemeines:

Projektname: Printbohrplotter

Schüler: Mayr Bernhard

Klasse: 5bhe

Betreuer: Herr Dipl.Ing. Höfer

Zeitraum: Ausgabe: 18. September 2000

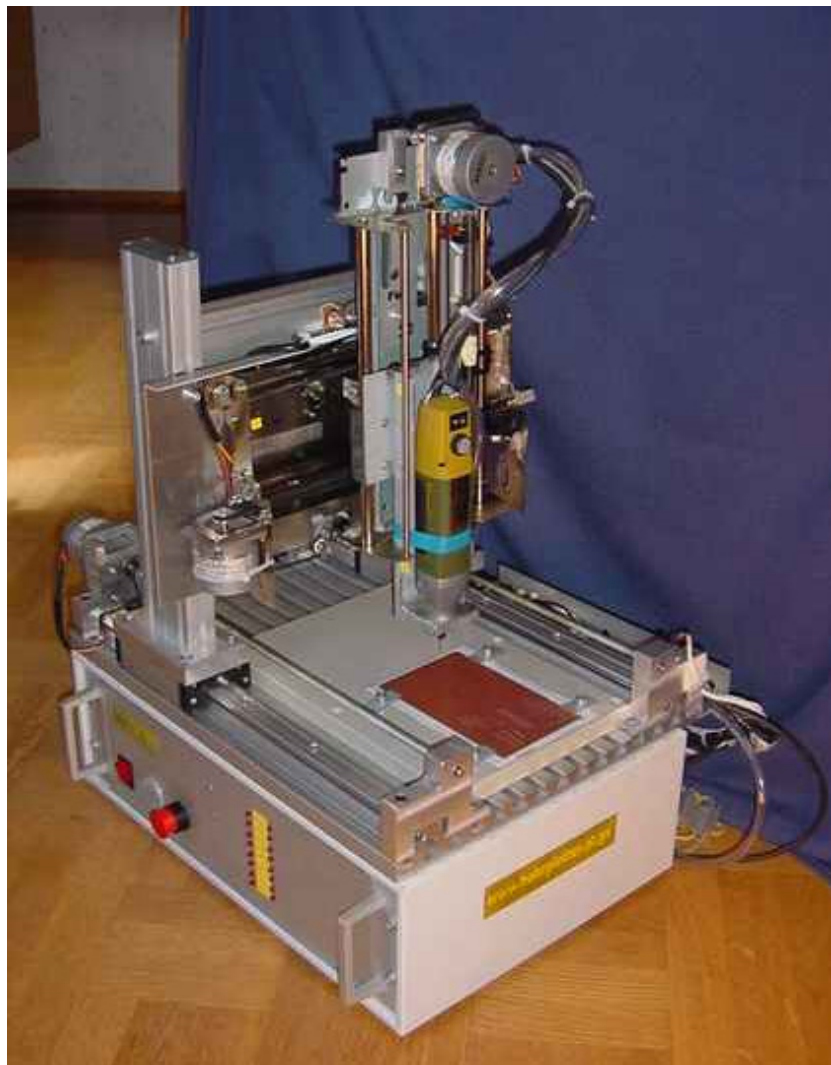
Abgabe: 7. Mai 2000

Insgesamt 28x 8 UE

Aufgabenstellung:

Der Schüler hat die Aufgabe, einen Bohrplotter selbstständig zu dimensionieren und zu fertigen. Dabei ist die **Mechanik**, die komplette **Elektronik** und die dafür notwendige **Software** zu erstellen.

Der Bohrplotter soll eine Platine, deren Layout im Eagle erstellt wurde, automatisch bohren. Die nötige ULP-Datei zum Auslesen der Koordinaten ist selber zu schreiben. Weiters ist eine Software, die die Bohrdatei einliest und die Koordinaten nach Bohrdurchmesser und nach Abstand sortiert, zu erstellen. Die Software ist zusätzlich für die Rampensteuerung und die Ausgabe über die Parallele Schnittstelle verantwortlich. Zur Ansteuerung der 3 Achsen ist eine Schrittmotorkarte für 3 Bipolar- Schrittmotoren zu dimensionieren. Für die Mechanik sind keine Vorgaben vorgesehen.



2. Mein Projekt:

Wieso einen Bohrplotter?

Mein Maturaprojekt ist eigentlich für unseren Schulzweig ein wenig unpassend. Nachrichtentechnik lässt sich schwer mit einem Bohrplotter assoziieren. Bei diesem Projekt ist nicht nur eine Elektronik, sondern auch eine Software zu schreiben, sowie eine aufwendige Mechanik zu konstruieren. Wer soll die zeitraubende Mechanik in den 28 Labortagen fertigen? Zu Schulbeginn war mein Abteilungsvorstand alles andere als zuversichtlich, als ich ihm mein Projektthema vorstellte. In der Schule wurde bereits vor drei Jahren ein Bohrplotter als Projektthema ausgegeben. Allerdings wurde in diesem Fall vom Betreuer die komplette Mechanik und Elektronik zur Verfügung gestellt. Die zwei Schüler mussten lediglich die Software zur Ansteuerung programmieren. Dabei stellte sich heraus, dass die Genauigkeit des Plotters aufgrund von Software- oder Hardwarefehlern viel zu ungenau sei. Ich ließ mich jedoch von meiner Idee, einen Bohrplotter als Maturaprojekt zu bauen, nicht abhalten. Grundsätzlich wäre laut Maturaverordnung auch die Möglichkeit gegeben gewesen, den Bohrplotter im Rahmen einer Diplomarbeit zu führen. Der Vorteil darin wäre eine Ersparnis der Projektwoche bei der Matura gewesen. Herr Gidl, mein Abteilungsvorstand hielt mich jedoch vor der Diplomarbeit ab, weil es zu riskant sei. (Ein nicht- funktionierender Bohrplotter hätte die Note 5 im Maturazeugnis bedeutet!) Jedoch bekam ich die Möglichkeit, mein Wunschprojekt als Projektarbeit zu machen. Da die anderen Projektlehrer schon mit ihren eigenen Projektthemen belegt waren, stellte sich Herr Höfer als mein Projektbetreuer zur Verfügung. (Normalerweise werden die Projektthemen vom Lehrer ausgegeben. Ich und Berninger Peter durften jedoch unsere eigenen Wunschthemen wählen.) An dieser Stelle möchte ich erwähnen, dass Herr Höfer zu Schulanfang der einzige Lehrer war, der mir optimistisch gegenüberstand. Der große Anreiz einen Bohrplotter selber zu bauen, war aber für mich trotzdem gegeben. Jetzt erst recht wollte ich meinem AV beweisen, dass ich in der Lage bin einen Bohrplotter zu bauen.

Printbohrplotter selber bauen? Ja aber warum?

Das Interesse für dieses Projekt entwickelte ich in den Sommerferien 1999. Der große Reiz, einen Printbohrplotter selber zu bauen, bestand für mich darin, dass ein gekaufter fast unbezahlbar wäre! Typische Preise für einen Bohrplotter (siehe ELV Katalog) sind 60.000ATS bis ca. 250.000ATS!!

Ein weiterer Grund, der mich zum Selbstbau veranlasste, ist die universale Anwendbarkeit eines Bohr- Fräsplotters. Zum Beispiel lässt sich der Plotter als kleine CNC - Fräsmaschine verwenden. Die eignet sich z.B. bei einem Gehäusebau für Schalter- Ausnehmungen. Eine weitere Anwendung wäre das Gravieren oder das Beschriften von Platinen (Siebdruckersatz) Mit der nötigen Genauigkeit würden sich genauso die Platinen nicht nur bohren sondern auch fräsen lassen. (Dadurch würde man sich den Ätzvorgang ersparen) Das Fräsen von Platinen liegt jedoch nicht in meinem Aufgabenbereich. (zu schwierig)

Ich kann nur jedem, der Interesse an einem Printbohrplotter hat, empfehlen, einen selber zu bauen, weil dieses Projekt so abwechslungsreich ist. Die Arbeiten reichen von der Mechanik bis zur Elektronik und Programmierung. Obwohl man für das komplette Gerät rund 10000,- im Eigenbau rechnen kann, ist es im Vergleich zu einem gekauften Bohrplotter sehr billig.

Obwohl dieses Projekt sehr aufwendig ist (speziell die Mechanik), habe ich versucht, den Plotter im Zuge der Labor5- Projektarbeit zu bauen. Außerdem ist die Schulwerkstätte meine einzige Möglichkeit, Metallstücke zu drehen oder zu fräsen.

3. Konzept:

Wie ich schon erwähnt habe, haben meine ersten Überlegungen bereits im Sommer 1999 begonnen. Um ein Konzept zu erstellen, ist es notwendig, sich über Bohrplotter genauestens zu informieren. Dabei stellte das Internet eine große Hilfe. Über das Internet konnte ich von anderen Personen, die auch einen Bohrplotter im Eigenbau machten, wichtige Informationen gewinnen. Allerdings blieb es mir nicht erspart, sich mit der Schrittmotortechnik auseinanderzusetzen. Nachdem ich passende Schrittmotorbausteine gefunden hatte, studierte ich mühselig die Applications- Notes von den Bausteinen. Nach einiger Zeit hatte ich mir einen Überblick über Bohrplotter verschaffen, und konnte somit ein Grobkonzept aufstellen. (Dieses war vor Maturaprojekt- Start vorhanden) Bereits aber nach den erstes Projektwochen kristallisierte sich ein klares Konzept heraus. Da sich dieses Projekt in drei Hauptbereiche, nämlich die Mechanik, die Elektronik und die Software strukturieren lässt, möchte ich anschließend auch mein Projektkonzept nach diesen 3 Punkten gliedern.

3.1. Mechanik:

Die X- Achse besteht aus zwei Linearführungen von der Firma Isel. Auf den Linearführungen sorgen zwei Wellenschlitten für guten Lauf. Der Antrieb der X- Achse wird als synchroner Tandemantrieb mit zwei Gewindestangen realisiert. Die Verbindung der Gewindestangen erfolgt über einen Zahnriemen.

Die Y und Z- Achse wurden aus alten Druckerschlitten gebaut. Der Antrieb erfolgt bei beiden Achsen über einen Zahnriemen. D.h. kleinere Auflösung! Im nachhinein würde ich die Y u. Z- Achse auch mit gekauften Linearführungen realisieren. Diese sind eindeutig stabiler, genauer und unter Belastung leichtgängiger! Der Grund für die Verwendung von Druckerschlitten war eine reine Sparmaßnahme.

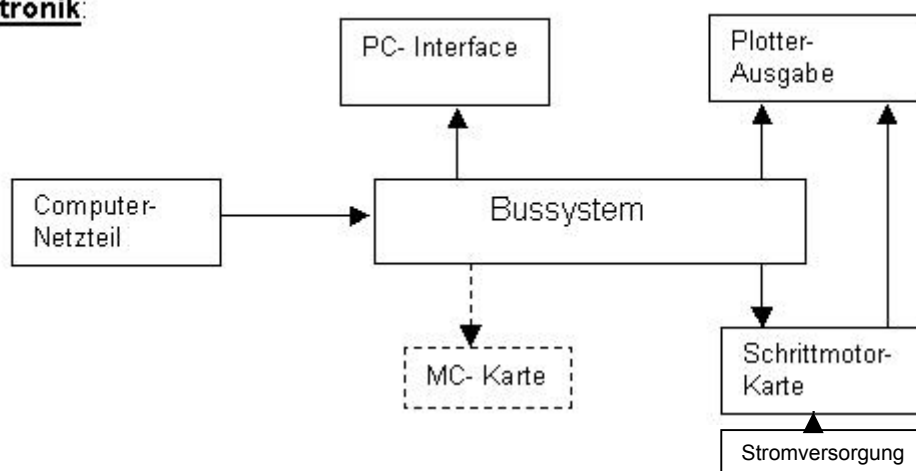
Als Schrittmotoren werden starke bipolar- Motoren von alten Siemens- Druckern verwendet.

Daten: $1,8^\circ / I_{\text{Phase}} = 1,5\text{A}$

3.2 Elektronik:

Überblick:

Elektronik:



Computer Netzteil:

Von einem alten AT- Gehäuse; Leistung: 18A/ +5V für Logik, 6A/ +12V für Brückentreiber und Lüfter.

Dieser Netzteil wird zur kompletten Versorgung des Bohrplotters verwendet; Einspeisung ins Bussystem; Deaktivierung mit Netzschalter oder NOT AUS!

Um höhere Taktgeschwindigkeiten zu erreichen, ist nachträglich ein 42V-Netzteil (externe Stromversorgung für Schrittmotorkarte) eingebaut worden.

Bussystem:

Als Bussystem wird eine 1:1 Backplane verwendet. Leiterplattenverbindungen werden mit VG-Stecker (Din 41612) realisiert. Zur Anbindung des PC- Interfaces wird ein Pfostenstecker verwendet.

Warum Bussystem? -> Bohrplotter soll beliebig erweitert werden können; z.B.: mit einer Microcontroller- Karte zur gesamten Verwaltung (Lichtschranken, Schrittmotorkarte, LCD-Anzeige u.s.w.) des Systems. Ein wichtiger Vorteil des Einsatzes einer MC- Karte ist die Unabhängigkeit der Rechnerleistung. Weiteres wird der Bohrablauf nicht mehr durch starten eines anderen Programms unterbrochen.

PC- Interface:

Das PC- Interface lässt sich über einem Schalter einstellen, ob die serielle Übertragung (für MC) od. die parallele Übertragung für Echtzeitansteuerung aktiviert ist. Bei meiner Software wird zu Zeit für die Übertragung die Parallele Schnittstelle verwendet. Die Serielle Schnittstelle wäre bei Einbindung eines Mikrocontrollers nötig.

Die Platine beinhaltet zusätzlich die nötigen Pull-Up/Down Widerstände sowie die Bustreiber für die Ein/Ausgangssignale.

Plotter- Ausgabe:

Die Plotter- Ausgabe ist eine kleine Platine, auf der die Schrittmotorsignale sowie die Gabellichtschranken über einen passenden Stecker nach außen bzw. nach innen geführt werden. Zusätzlich beinhaltet die Platine die Anbindung der Gabellichtschranken an die Busplatine.

Schrittmotorkarte:

Grundsätzlich ist diese Karte mit der Baustein- Kombination L297 u. L298 von SGS- Thomson aufgebaut.

Features:

- ☀ Karte für drei Achsen
- ☀ für Bipolar- Schrittmotoren mit max. 2A pro Phase und einer Motorspannung bis max. 42V (Strom- Copper)
- ☀ Halb- od. Vollschritt
- ☀ komplette Referenzspannungsaufbereitung für Stromregler (Boost/ Sleep- Signale separat für jede Achse vorhanden)
- ☀ Motorstrom über Spindeltrimmer für jeden Kanal einstellbar
- ☀ Shaping- Funktion: sorgt für optimal gleichmäßiges Drehmoment bei Halbschritt; zusätzliche Resonanzunterdrückung
- ☀ Relais- Schaltkreis für Bohrmaschine und Absaugung
- ☀ PTC- Temperaturüberlastschutz
- ☀ Strom- Copper- Betrieb sorgt für hohes Drehmoment auch bei schnellen Drehzahlen

Jede Achse wird mit einem Taktsignal (pro Takt ein Schritt) und einem Richtungssignal angesteuert. Zum Beschleunigen eignet sich der Boost/Sleep- Betrieb (Referenzspannung entspricht dem Soll- Phasenstrom) Bei der Acceleration und Deceleration wird der Phasenstrom durch das Boost/Sleep- Signal erhöht.

!!!Die Treiberkarte eignet sich nur für Bipolar- Motoren, ansonsten müsste statt des Brückentreibers L298 L6222 verwendet werden! Die Schrittmotorkarte ist auf einer Euro-Platine als 19"- Einschub für mein Bussystem dimensioniert.

3.3 Software:

Über eine Software werden die Bohrkoordinaten vom Eagle- Layouteditor ausgelesen und anschließend nach Bohrdurchmesser und nach Abstand sortiert. Als nächster Schritt werden per Software die Rampen für die Beschleunigung b.z.w. für die Verzögerung errechnet. Letzte Aufgabe der Software ist die direkte Ausgabe der Daten über die parallele Schnittstelle. Die Daten (Takt und Richtung für jede Achse sowie Bohrer, Absaugungssignal, Boost/Sleep-Signale, Referenzsignale von Gabellichtschranken und andere Statussignale) werden direkt von der Schrittmotorkarte verarbeitet.

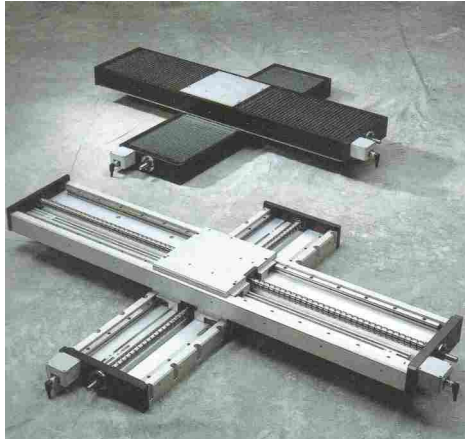
Die Software wird Im C++ Builder 4.0 programmiert. Ein passendes Komponent für die parallele Schnittstelle wurde nach langer Suche im Internet endlich gefunden.

Mit diesem Konzept entwickelte ich die einzelnen Platinen, zeichnete die einzelnen Alukörper und schrieb größtenteils die Software zur Ansteuerung. An dieser Stelle möchte ich mich bei meinem Freund Christian Scarlet bedanken, der mich beim C++ Builder – Lernen kräftig unterstützt hat und mir auch Teile von meiner Software geschrieben hat.

Anschließend werde ich die einzelnen 3 Punkte im Detail behandeln. Die genauen Funktionsbeschreibungen werden in den anschließenden Kapiteln zu finden sein. Auf Werkstückspläne, Schaltpläne, Berechnung, Layout, Bestückungsplan sowie Stückliste wird in den nächsten Kapiteln auf eine Seitenzahl verwiesen.

4. Mechanik im Detail:

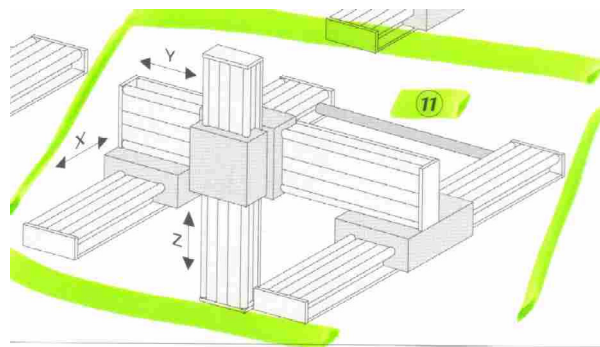
4.1 Allgemein: Wenn man die Mechanik eines Bohrplotters betrachtet, bemerkt man, dass einige Ansprüche auf Genauigkeit, Leichtgängigkeit, Trägheit gesetzt werden. Diese Ansprüche sprechen eigentlich alle gegen einen Eigenbau der Mechanik für einen Bohrplotter. Da außerdem die Fertigung von genauen Linearführungen sehr schwierig ist, informierte ich mich ausführlich über fertige Linearführungen und Flachbetтанlagen. Ich ließ mir von den verschiedensten Firmen wie Bosch- Automatisierung, IEF-Werner, THK, Step4 und Isel Informationsmaterial zuschicken.



Das linke Bild zeigt zum Beispiel eine Lineareinheit der Firma THK. Diese Lineareinheiten sind frei kombinierbar und natürlich hochpräzise. Jetzt stellte sich nur noch die Frage, wie der Aufbau gewählt werden sollte. Entweder man positioniert die Bohrspindel, wobei das Werkstück in Ruhelage bleibt oder man positioniert das Werkstück und die Bohrspindel bleibt in Ruhelage. Letztere Methode (stellt auch das linke Bild dar) erscheint eigentlich sinnvoller, weil im Normalfall eine Bohrspindel schwerer als ein Werkstück (z.B.: Platine)

ist. Der Grund, wieso ich mich andersrum entschied, ist der eingeschränkte Arbeitsbereich bei

einer gegebenen Fläche. Somit entschied ich mich für einen Aufbau, wie er im rechten Bild zu sehen ist (Positionierung der Bohrmaschine). Für diesen Aufbau wären wiederum Linearführungseinheiten der Firma IEF- Werner geeignet. Diese hochgenauen Lineareinheiten müsste man nur mit Verbindungsstücken zusammenschrauben und die Flachbetтанlage wäre fertig. Doch leider



wurde mir bei meiner ersten Preisanfrage gesagt, dass ein Achsenmodul mit Gewindespindeltrieb rund 13000 ATS kostet. Somit war mir klar, dass ich diese teure Industrietechnik aus Kostengründen nicht verwenden kann. Auf



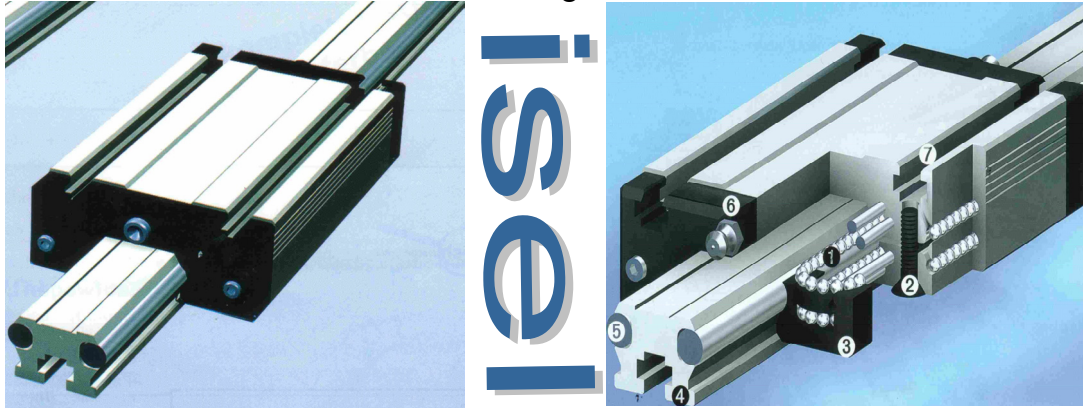
der Suche nach einer Alternative stieß ich auf die Modellbaufirma Step-four. Ein Fertigerät mit Software und Elektronik kostet dort 36000 ATS. Die Flachbetтанlage alleine kostet jedoch noch immer 20000ATS. Da ich mein Investitionslimit auf 10000ATS gesetzt habe, suchte ich weiter. Eine passende Lösung fand ich bei der Firma Isel. Dort bekommt man preisgünstige Linearführungen auch ohne Antrieb. (Der Antrieb mit einer Kugelgewindespindel und den nötigen Lagern ist meist das teuerste). Aus

Kostengründen entschied ich mich, nur die X-Achse aus gekauften Führungen zu bauen und Y- sowie Z- Achse aus den Linearführungen von alten Nadeldruckern zu bauen.

4.2 X-Achse: Nachdem ich aus dem Katalog passende Linearführungen für die X-Achse gewählt hatte, bestellte ich beim Distributor (KSE Kontaktsysteme) von der Firma Isel zwei Linearführungen. Weiteres bestellte ich noch eine T-Nut-Platte mit den Abmessungen 375*500mm als Grundplatte und ein T-Nut-Rechteckprofil als Aufbau. Aus dem Rechteckprofil wurde ein H-Aufbau gebaut, der stirnseitig über einen ALU-Block mit den

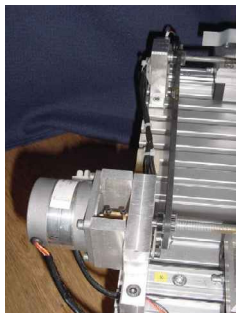
Wellenschlitten verschraubt wurde. In die Linearführungen MLF1 wurden Befestigungslöcher mit einem Zapfensenker gebohrt, sodass die Linearführungen mit Zylinderschrauben und T-Nutsteine auf der Grundplatte befestigt werden können.

Die nächsten zwei Bilder zeigen die Linearführung MLF1 und den dazugehörigen Wellenschlitten WS1 der Firma Isel. Bei den Wellenschlitten sorgen 4 Kugelumläufe für guten Lauf. Außerdem sind die Wellenschlitten wartungsfrei.



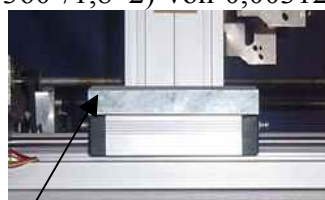
Der Antrieb der X-Achse wurde als Tandemantrieb realisiert. Jeder der beiden Wellenschlitten wird über eine M8-Gewindestange angetrieben. Die Gewindestange ist auf beiden Seiten gelagert. Die Lager sind jeweils in einem Alu-Lager-Block eingebaut. Die Lagerblockpläne sowie den Gewidemutterblockplan finden sie im [Anhang auf Seite 78 & 79](#). Bei einer Seite ist

die Gewindestange direkt mit dem Schrittmotor gekoppelt. Die Alu-Blöcke zwischen Motor und Lagerblock dienen für eine gute Wärmeableitung. Mit einem Zahnriemen wurde die Kraftübertragung auf die zweite Gewindestange realisiert, sodass sich beide Gewindestangen immer synchron bewegen. Auf beiden Wellenschlitten wurde ein Gewinde-Mutterblock montiert. Dadurch wird durch eine Rotationsbewegung der Gewindestangen eine Linearbewegung der Wellenschlitten durchgeführt. Die Steigung einer



Gewidemutterblock

M8-Gewindestange beträgt 1,25mm, sodass sich eine theoretische Auflösung bei einem 1,8°- Schrittmotor im Halbschrittmode (400 Schritte pro Umdrehung = $360^\circ/1,8 \cdot 2$) von 0,003125mm/Schritt ergibt. Da alle anderen mechanischen Komponenten um



Schlittenblock

einiges ungenauer sind, hat die Auflösung keineswegs mit der tatsächlichen Genauigkeit zu tun. Am linken Bild ist gut die Montage des Rechteckprofils zu sehen. Das Rechteckprofil wurde auf der Unterseite des Schlittenblocks stirnseitig mit zwei M8-Schrauben verschraubt. Der Schlittenblock selbst ist an den 4 Ecken mit T-Nutsteinen und Zylinderschrauben am Wellenschlitten montiert. Den Plan für den Schlittenblock

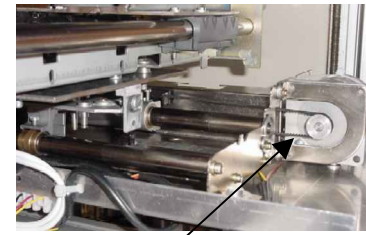
finden sie im [Anhang auf Seite 76](#).

4.3 Y-Achse: Die Y-Achse wurde aus der Mechanik eines alten Mannesmann- Druckers gefertigt. Obwohl die zwei Stahlführungen einen Durchmesser von 12mm haben, bilden diese keine starre Führung. Die Blechkonstruktion der Y-Achse ist für den schweren Z-Achsenaufbau überlastet. Dies bringt einerseits Ungenauigkeit und andererseits Schwergängigkeit der Linearführung. (Die alte Druckerkopfaufnahme läuft auf den Wellen nur mit Gleitlagern jedoch nicht mit Kugellagern. Die Gleitlager sind für solche Belastungen, wie sie durch die Z-Achse



Y-Achse

zum Einsatz kommen nicht geeignet.) Daher kann ich nur jedem empfehlen, das Geld für gekaufte Linearführungen (z.B.: von der Firma Isel) auch für Y- und Z-Achse aufzuwenden. Der Antrieb erfolgt bei der Y-Achse über einen Zahnriemen. Da meine Schrittmotorkarte nur für bipolare Motoren geeignet ist, musste der originale unipolare Motor ersetzt werden. Als Ersatz für den Originalmotor wurde der gleiche Motor wie bei der X-Achse eingebaut. (1,8° / 1,5A pro Phase, bipolar) Die Auflösung errechnet sich aus dem Durchmesser des Antriebrades. (Durchmesser: 16,2mm * P = 51mm / 400 Schritt pro Umdrehung = 0,127 mm/Schritt) Wenn man dieses Ergebnis mit dem der X-Achse vergleicht, stellt man einen Faktor von rund 40 als Unterschied fest. Bei einer gekauften Linearführung würde ich auf jeden Fall wie bei der X-Achse einen Gewindestangen-Antrieb einbauen.



Y- Antrieb

4.4 Z-Achse: Die Z-Achse wurde aus einem alten Epson-Drucker ausgebaut. Zur besseren



Führung der Bohrmaschine wurde noch eine dritte Welle montiert. Die Bohrmaschinenhalterung selbst läuft jedoch über zwei Gleitlager nur auf zwei Wellen. Da die Abwärtsbewegung durch die Erdbeschleunigung wesentlich leichter als die Aufwärtsbewegung der Z-Achse geht, wurde eine lange Feder als Ausgleich zwischen Motor und Bohrmaschinen-Halterung gespannt. Die Halterung für die Bohrmaschine wurde aus einem 2mm- Stahlblech gebogen und ist daher sehr stabil. Die Bohrmaschine selbst ist nur am unteren Ende der

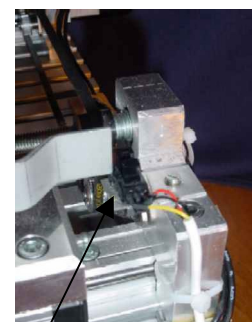


Bohrmaschinenblock

Bohrmaschinenhalterung über einen Bohrmaschinenblock mit einem Zylinderschrauben eingespannt. Den Plan für den Bohrmaschinenblock finden sie im [Anhang auf Seite 77](#). Der an der Bohrmaschine heruntergeführte PVC-Schlauch dient als Spanabsaugung. Die komplette Z-Achse ist über eine quadratische Stahlplatte mit der ehemaligen Druckerkopfhalterung der Y-Achse verbunden. Durch den langen Hebelarm der Bohrmaschinenhalterung ist der komplette Aufbau einwenig unstabil, daher würde ich auch für die Z-Achse gekaufte Linearführungen empfehlen. Der Antrieb ist wie bei der Y-Achse über einen Zahnriemen. (Besser wäre auch hier ein Gewindestangenantrieb) Der originale Motor wurde inklusive Motorflansch durch einen eigenen ersetzt. Auch hier ist

wiederum der Motor der gleiche Bipolar- Motor wie bei den anderen zwei Achsen. Die Auflösung von der Z-Achse ist gleich wie bei der Y-Achse. (0,127mm/Schritt) Als Bohrmaschine wird die Proxxon Micromot 220/E, regelbar von 5000- 20000 Umdrehungen/ min, 100W/ 220V und mit Stahlspannzangenfutter für extreme Rundlaufgenauigkeit verwendet.

4.5 Kalibrierung: Vor jedem Bohrdurchlauf werden alle 3 Achsen über Gabellichtschranken kalibriert. Jede Achse ist mit einem Referenzgabellichtschranken sowie mit einem Endgabellichtschranken ausgestattet, wobei die Endgabellichtschranken nicht achsenspezifisch abgefragt werden können. Nachdem jede Achse ihren Endpunkt erkannt hat, bewegt sich diese mit einem bestimmten Offset zum Aufspannnagel. Dort hat der Benutzer noch die Möglichkeit, eine Augenkontrolle durchzuführen, sowie die Bohrtiefe manuell einzustellen. Sobald der Bohrablauf gestartet wurde, bewegt sich die Bohrmaschine zum absoluten Nullpunkt. Der genaue Kalibriervorgang ist in der Softwarebeschreibung zu finden.

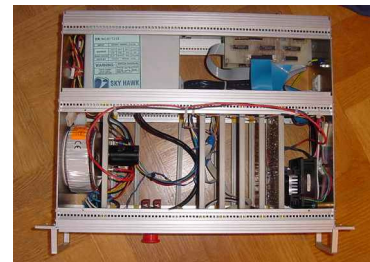


Gabellichtschranke

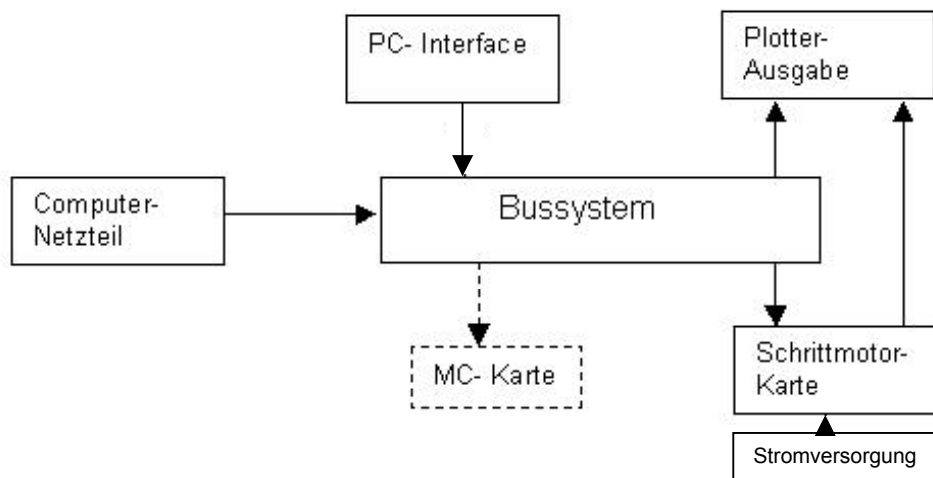
5. Elektronik im Detail:

5.1 Allgemein:

Ein wichtiger Punkt für den Entwurf der Elektronik war für mich die Erweiterbarkeit sowie ein modularer Aufbau. Dies setzt ein Bussystem voraus. Die erste Erweiterung soll eine Mikrocontrollerkarte sein, die die Schrittmotorkarte ansteuert und für die Peripherieverwaltung (LCD-Anzeige, Taster) zuständig ist. Daher habe ich beim PC-Interface nicht nur die Parallele Schnittstelle sondern auch die Serielle Schnittstelle, welche für die Microcontrollertechnik entscheidend ist, integriert. Somit stehen die Signale Rx/Tx am Bussystem für eine zukünftige MC-Karte schon zur Verfügung.

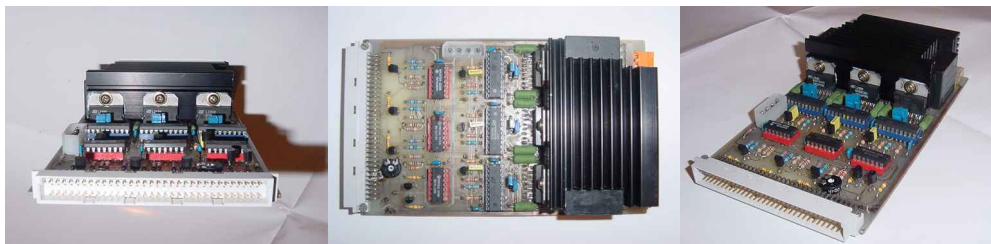


Eine grundlegend wichtige Entscheidung ist auch die Auswahl der Schrittmotor-ICs. Meine ersten Erfahrungen mit Schrittmotoren machte ich mit dem Schrittmotor-IC SAA1027 von der Firma Phillips. Jedoch merkte ich bald, dass dieser IC für meine Anwendung nicht ausreichend ist (maximale Phasenströme bis zu 500mA, für unipolare Motoren). Da ich auf jeden Fall bipolare Schrittmotoren einsetzen wollte, (Bei gleicher Motorgröße mehr Phasenstrom möglich, weil nur eine Spule pro Phase verwendet wird.) musste ich auf jeden Fall nach einem neuen IC suchen. Nach längerer Suche im Internet traf ich auf die IC-Kombination L297/L298 von SG-Thomson. Diese Kombination ermöglicht eine Schrittmotoransteuerung mit wenigen zusätzlichen Bauteilen. Unter den Aspekten Erweiterbarkeit, Modularität und mit der IC-Kombination L297/ L298 versuchte ich ein Konzept für die Elektronik zu erstellen:



5.2 Schrittmotorkarte:

Die Schrittmotorkarte wurde für 3 bipolare Schrittmotoren für Phasenströme von 0,6 bis 2A ausgelegt.



5.2.1 Die IC-Kombination L297/L298:

5.2.1.1 L297:

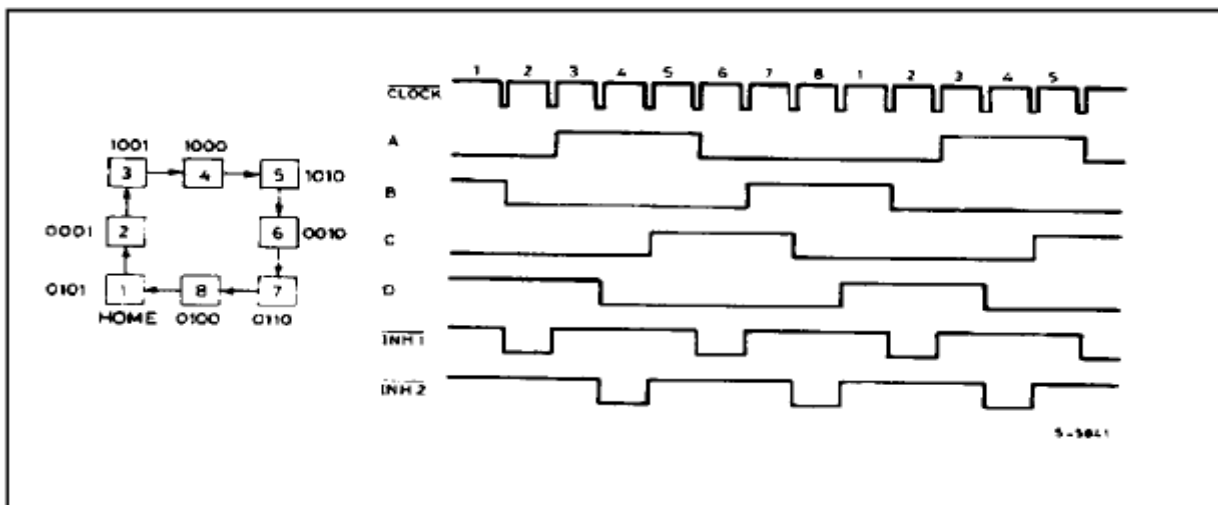
Der IC L297 von SG-Thomson generiert aus Takt und Richtungssignal das nötige Bitmuster zur Ansteuerung der Leistungstristorbrücke L298. Das generierte Bitmuster für

Half/Fullstep ist in den nächsten zwei Grafiken ersichtlich. Zu beachten ist, dass das Bitmuster ohne Berücksichtigung des Stromchopper dargestellt ist, sozusagen das „Sollbitmuster“

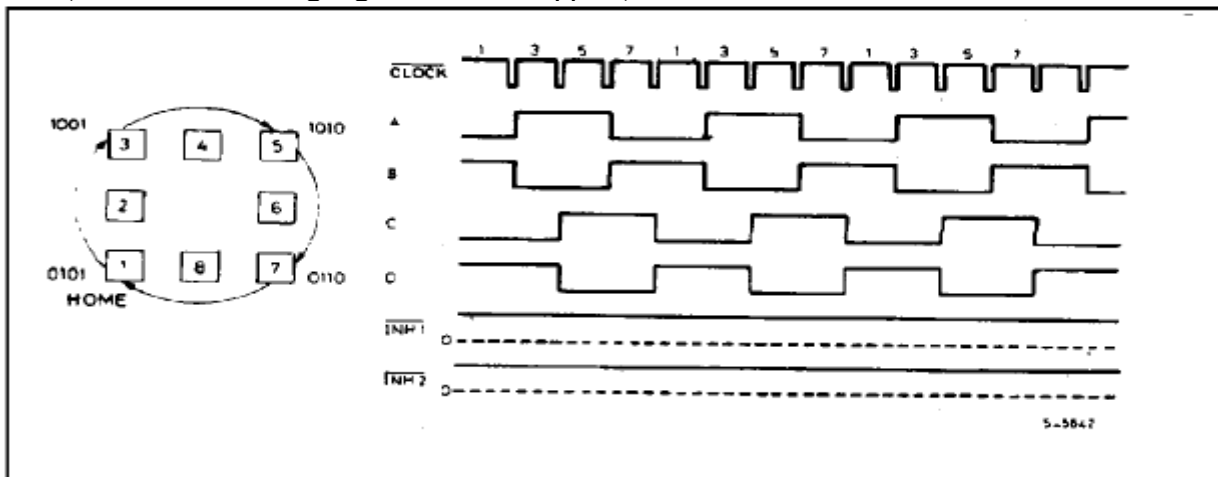
Kurze Beschreibung:

Die Phasen A und B sowie C und D repräsentieren die Spannungen an Spulenden zu Masse, wobei A und B die Phasen für Spule 1 sowie C und D die Phasen für Spule 2 sind. Inhibit1 repräsentiert den Phasenzustand für Spule 1 sowie Inhibit2 für Spule 2 zuständig. Half bzw. Fullstep wird mit dem Pin19 einstellen. Die Drehrichtung wird durch den Zustand von Pin17 (CW/CCW) bestimmt. Am Clock-Pin muss der Schritttakt angelegt werden. (CLK- Pin ist lowaktiv!)

Halfstep: Im Halfstepmode wird der physikalische Schrittwinkel halbiert. Daher ergibt sich bei meinen Schrittmotoren (1,8°) im Halfstepmode ein Schrittwinkel von 0,9°. Beim Analysieren des Bitmusters ist erkennbar, dass bei jedem zweiten Schritt nur eine Spule bestromt ist. Für diese Zeitspanne wird das jeweilige Inhibitsignal auf low gesetzt. Das Inhibit- Signal liefert für die später erklärte Shaping-Funktion die nötige Information.



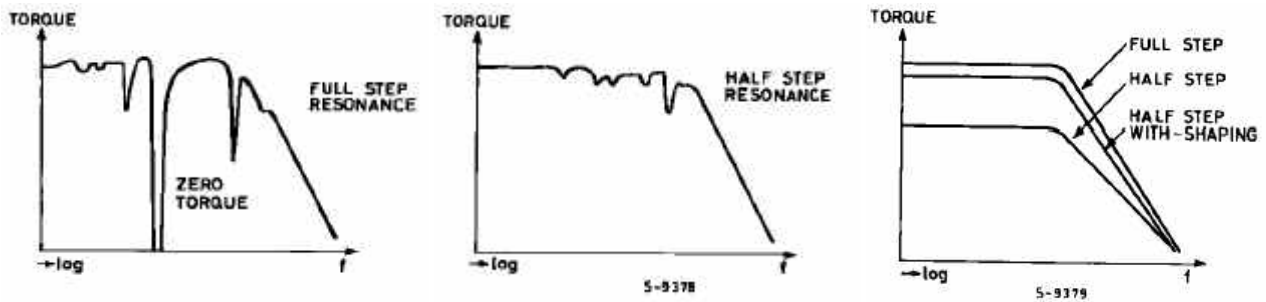
Fullstep: Der Fullstepmode wird auch „two-Phase-on-Mode“ genannt, weil zu jedem Zeitpunkt beide Spulen bestromt werden. Zu beachten ist, dass die Inhibit-Signale immer high sind (ohne Berücksichtigung des Stromchoppers)



Wahl der Betriebsart:

Der einzige Vorteil von Fullstep ist, dass der Motor um ein bisschen mehr Drehmoment als bei Halfstep mit Shaping (siehe Grafik) hat. Daher kann ich diesen Mode nur empfehlen, wenn keine hohe Anforderung auf die Schrittauflösung gestellt ist. Ansonsten würde ich auf jeden Fall den Halfstepmode verwenden (bei mir in Verwendung). Ein Nachteil bei Halfstep ist die

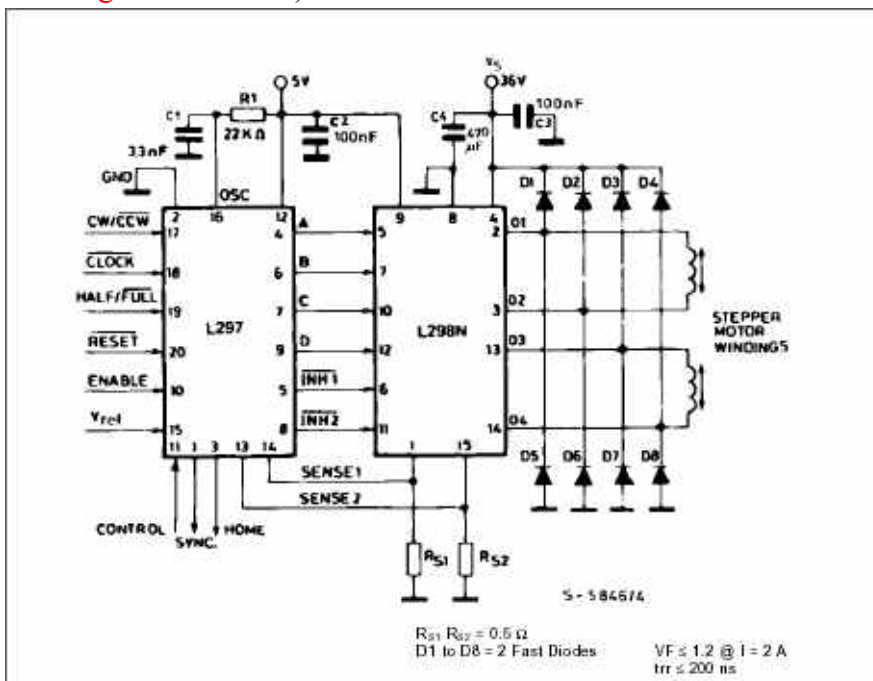
doppelt so hohe Taktfrequenz, welche für eine gleichschnelle Geschwindigkeit im Fullstep nötig wäre



Die wesentlichen Vorteile im Halbstepmode sind:

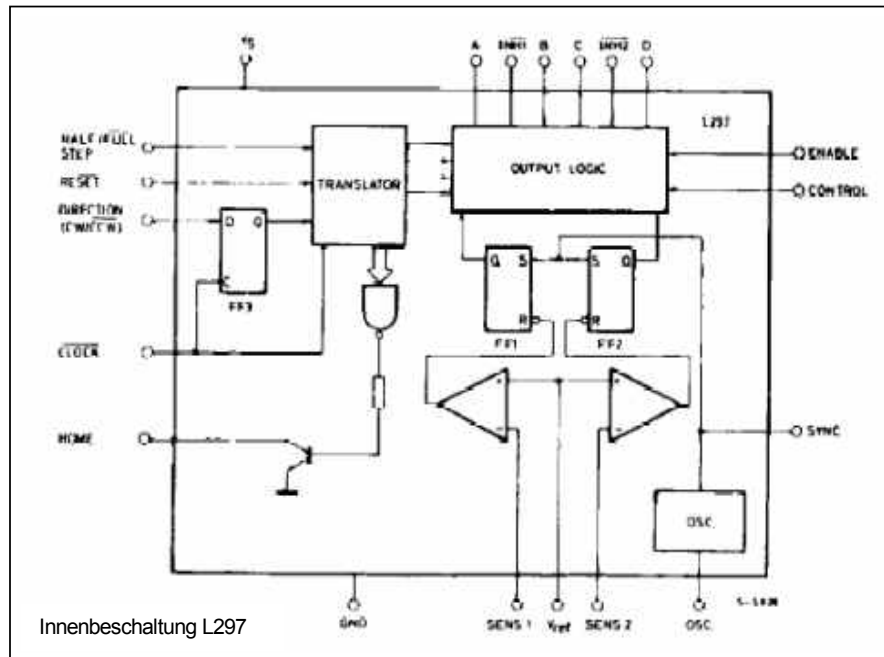
- o Geringere Resonanzprobleme und daher keinen frühzeitigen Zusammenbruch des Drehmomentes (Torque) . Dies würde zu extremen Schrittfehlern führen!
- o Doppelte Auflösung!

Stromchopper: Die Brückenversorgungsspannung ist um vieles größer, als die, welche für den nötigen Phasenstrom im Gleichstromfall erforderlich wäre (Motorimpedanz = 2,7 Ω / maximaler Motorphasenstrom = 2A). Dadurch steigt der Strom in der Spule schneller an (siehe Anhang S. 68 - 74) und erreicht daher schneller seinen Sollwert, wodurch höhere



Motorgeschwindigkeiten realisierbar sind. Damit der Strom seinen Sollwert nicht überschreitet, wird der Strom im Oszillatortakt mit dem Referenzwert verglichen. L297 besitzt einen internen Oszillator, der für den Stromchopper gedacht ist. Die Chopperfrequenz wird mit einem externen Kondensator und Widerstand eingestellt ($f = 1/(0,69 RC)$) und liegt bei einem typischen Wert von 25kHz (max. 40kHz). Diese

Chopperfrequenz ist die Frequenz mit der der über die Sensewiderstände gemessene Phasenstrom mit der Referenzspannung verglichen wird. Ist die Spannung am Sensewiderstand, welche den Phasenstrom repräsentiert größer als die Referenzspannung, so wird über das Inhibit- Signal diejenige Phase abgeschaltet. (genaueres siehe unter L298 Chopperarten) Wenn beim nächsten Taktzyklus die Spannung am Sensewiderstand schon kleiner als die Referenzspannung ist, wird diese Phase wieder eingeschaltet. Ansonsten bleibt die Phase so lange ausgeschaltet, bis die Sensespannung kleiner als die Referenzspannung ist. Ist diese Bedingung erfüllt, so wird die Phase wieder solange eingeschaltet, solange die Sensespannung kleiner als die Referenzspannung ist. Der IC L297 stellt auch ein Synchronisierungssignal zur Verfügung. Sobald mehr als ein L297 verwendet wird (bei meiner Schrittmotorkarte sind 3 in Verwendung), ist es zu empfehlen, die einzelnen Bausteine mit dem



SYNC-Pin (Pin1) zu verbinden.

Die Oszillatorbeschaltung darf dann nur bei einem Baustein angeschlossen werden. Durch die Synchronisierung werden Resonanzprobleme und Motorgeräusche reduziert.

Weitere Steuersignale:
Enable: Wenn Enable (Pin 10) low ist, werden beide Inhibit- Signale auf Low gesetzt, sodass beide Phasen ausgeschaltet sind.
Sense1/Sense2: Pin 13

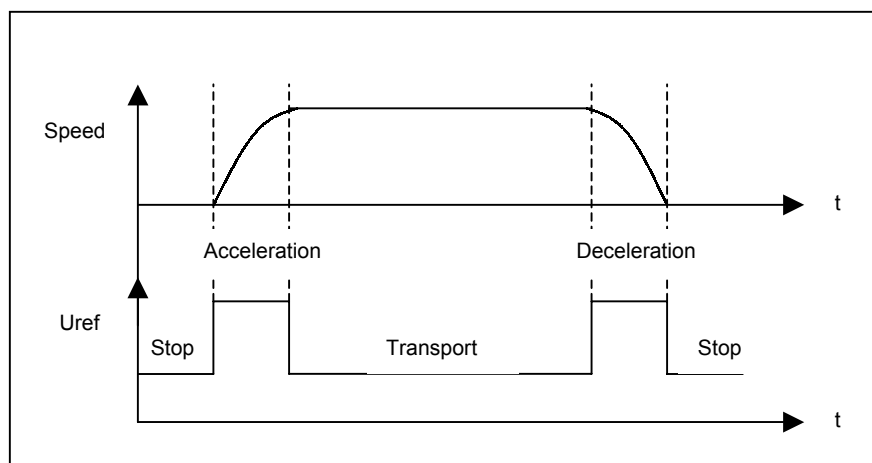
u. 14 sind die Eingänge für die Phasenstrommessung. **Home:** Dient als Ausgang und signalisiert, wenn der Bitmuster-generator im State 0 ist.

Reset: setzt den Bitmuster-generator auf State 0 zurück (lowaktiv). Beide Funktionen werden bei meiner Schrittmotorkarte nicht verwendet!

Uref: Pin 15 dient als Eingang für die Vergleichsspannung für den Stromchopper. Allerdings ist diese Referenzspannung bei meiner Schrittmotorkarte **abhängig vom:**

o **Boost/Sleep-Signal:**

Idee: Die Referenzspannung, welche den Sollwert des Phasenstroms repräsentiert, soll für jede Achse per Software in zwei Stufen einstellbar sein. Ist das Boost/Sleep-Signal high, stellt sich die Referenzspannung auf ihr Maximum. Der Zustand Boost ist für die Acceleration und Deceleration

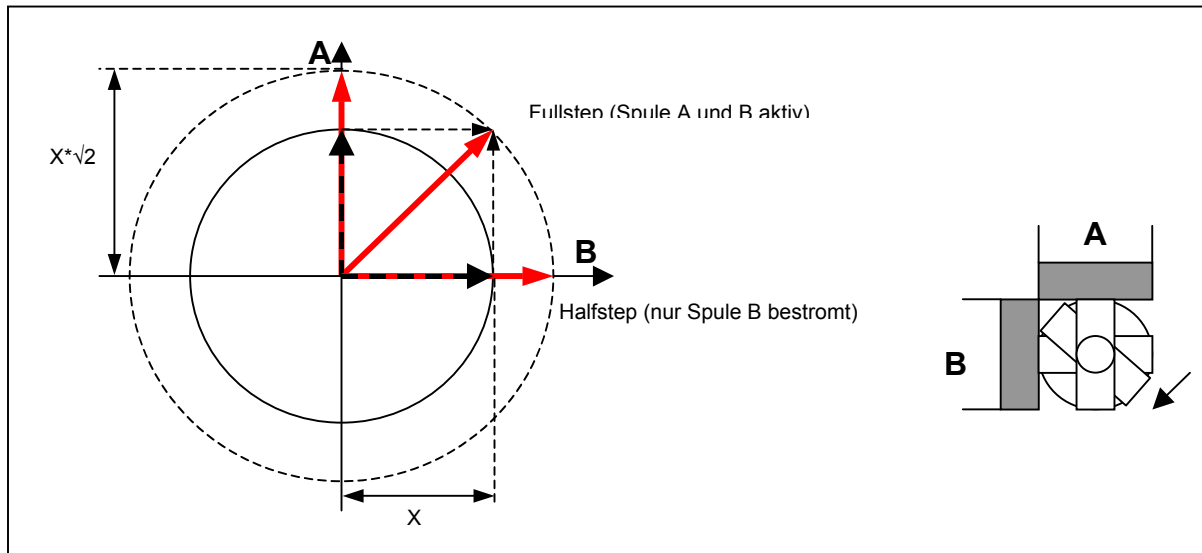


der einzelnen Achsen, wo ein hohes Drehmoment gebraucht wird, gedacht. Im Sleep-Mode (Boost/Sleep-Signal low) stellt sich die Referenzspannung auf ihr Minimum. Dieser Mode soll für die Zeitdauer zwischen Beschleunigung und Abbremsung der Achsen sowie im Ruhezustand

(damit sich die Motoren nicht unnötig erwärmen) verwendet werden. Ist der Bohrplotter eingeschalten, aber nicht in Arbeit, so ist es sinnvoll, mit dem Enable-Signal (bei meiner Platine über das Busy-Signal) die komplette Transistorbrücke abzuschalten.

o **Shaping-Mode:**

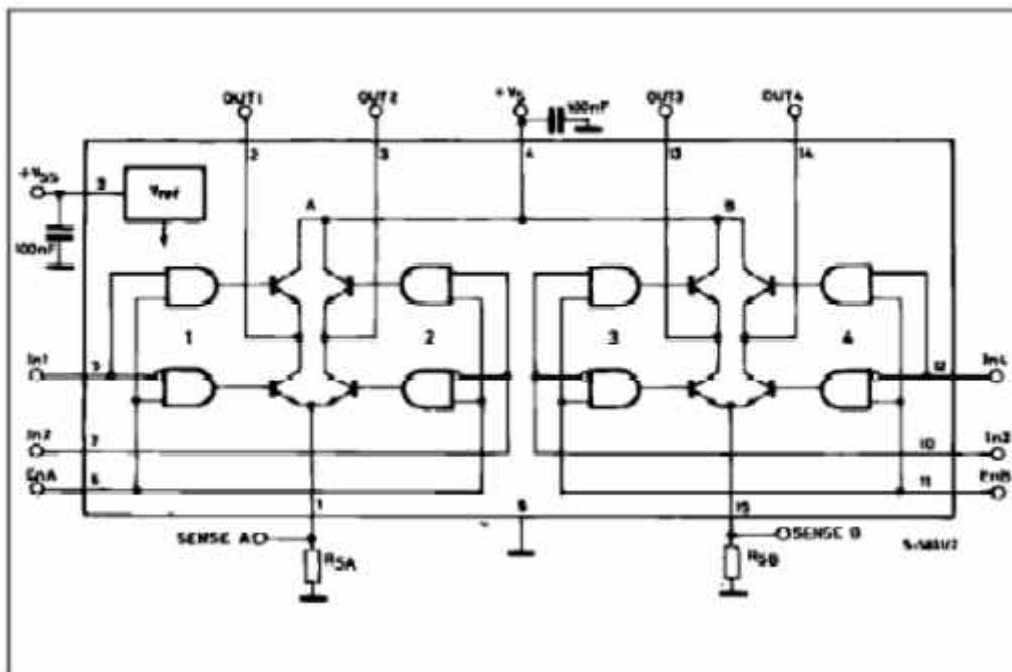
Dieser Mode ist im Halfstep-Mode von großer Bedeutung. Aus dem Bitmuster im Halfstep-Betrieb ist ersichtlich, dass bei jedem zweiten Schritt nur eine Spule bestromt ist (Im Vergleich zum Full-Step-Mode, wo zu jedem Zeitpunkt zwei Spulen bestromt sind). Wenn im Zustand,



in dem nur eine Spule bestromt ist, die Referenzspannung gleich bliebe, würde der resultierende Kraftvektor kleiner sein, als in einem Zustand, wo beide Spulen bestromt sind. Dies würde Resonanzprobleme, unangenehme Motorgeräusche und Drehmomentverlust zur Folge haben. Im Shapingmode wird mit Hilfe von den Inhibit- Signalen die Referenzspannung in den Zuständen, wo nur eine Phase aktiv ist, um den Faktor $\sqrt{2} = 1,41$ erhöht, wodurch auch der Phasenstrom um diesen Faktor ansteigt. Schaltungstechnisch wird dies mit einem Spannungsteiler und einem Transistor realisiert. Der Transistor wird über das &- Gatter, welches INH1 und INH2 als Eingang hat, geschaltet. Nur wenn beide Inhibit high (beide Spulen) aktiv sind, wird der Transistor leitend, sodass die Referenzspannung um den Faktor $\sqrt{2}$ sinkt.

5.2.1.2 L298:

L298 ist ein Dual- Full- Bridge- Driver designed für TTL- Pegel zum Schalten von Induktivitäten. Daher eignet sich dieser IC auch sehr gut zum Ansteuern von Schrittmotoren.



Die maximale Brückenversorgungsspannung beträgt 42V (46V Maximum Ratings) Diese Brückenspannung ermöglicht schon Taktgeschwindigkeiten bis zu 5kHz bei 2A pro Phase. Der Zusammenhang

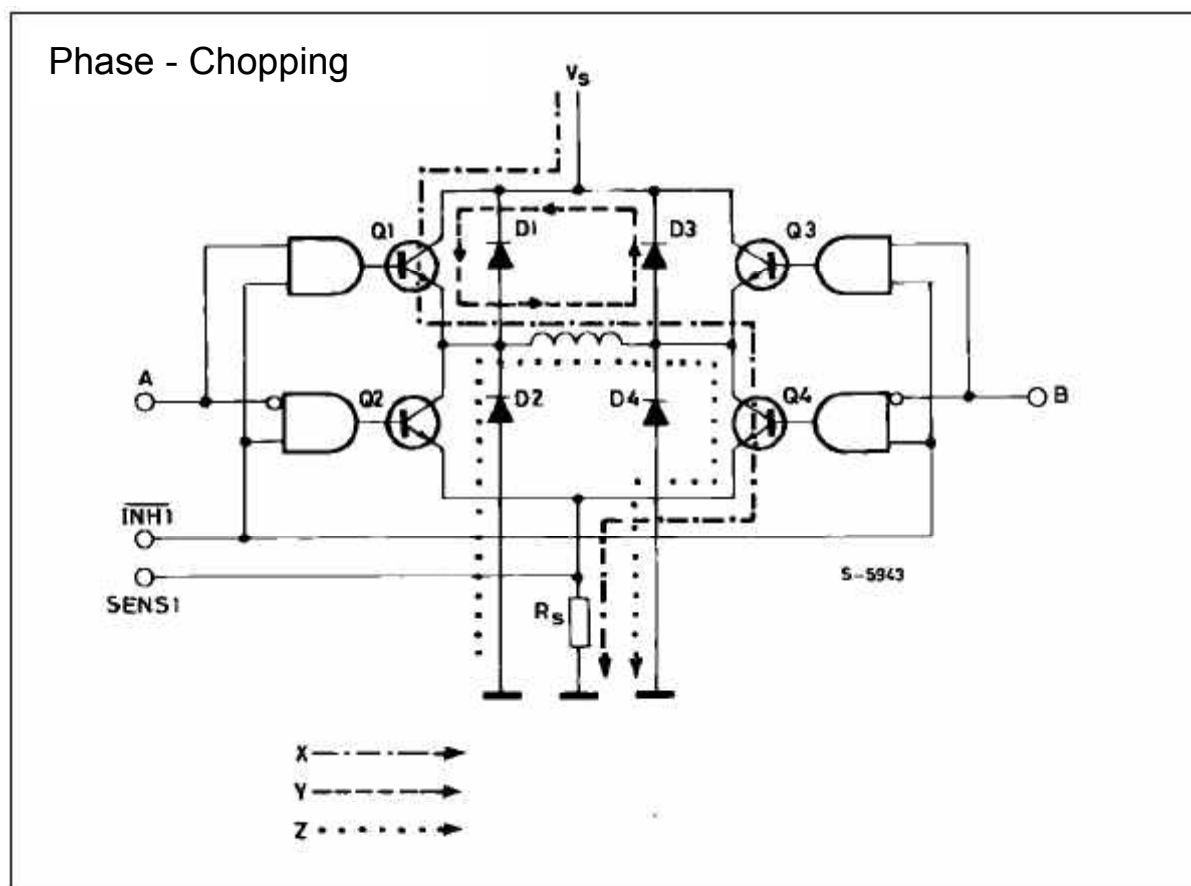
Brückenspannung und maximale Taktgeschwindigkeit ist an der Steigung der e-Kurve ersichtlich. (siehe Unterschied zwischen 10V u. 42V im [Anhang auf Seite 74](#)) Für den Betrieb

ist auch eine vergleichbar große Brückenspannung zu empfehlen. Bei kleinen Betriebsspannungen ist die hohe Sättigungsspannung der Leistungstransistoren des L298 (bei 2A rund 1,7V für den Transistor gegen Masse und 2V für den Transistor gegen V_s) nicht zu vernachlässigen.

L298 benötigt zusätzlich zur Brückenversorgungsspannung 5V für die interne Logik. Weiters besitzt der Baustein 6 Eingänge. A,B,C,D sind die 4 Phasen welche vom L297 für Vollschritt bzw. Halbschritt generiert werden. Die Inhibit-Signale (EnA u. EnB) sind für den Stromchopper notwendig. Für den Stromchopper gibt es zwei Möglichkeiten, welche sich beim L297 am Pin Control einstellen lassen:

Phase Chopping:

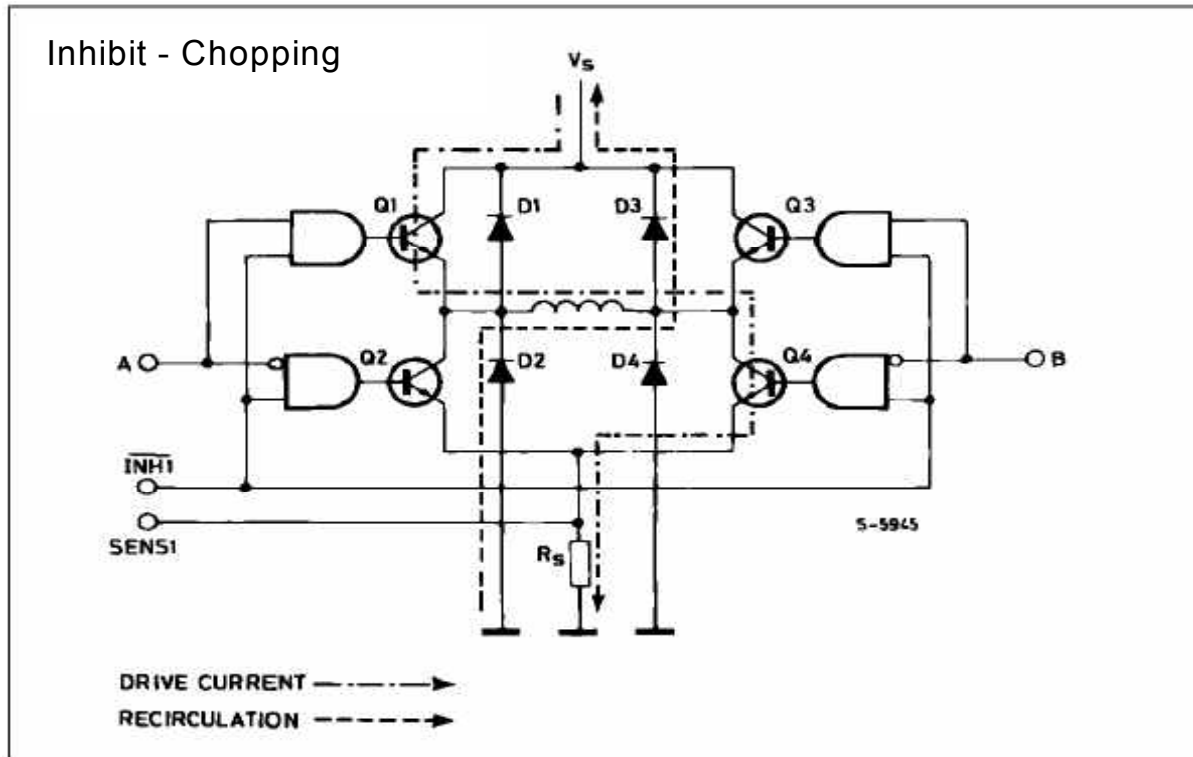
Spule wird nicht über das jeweilige Inhibit sondern direkt durch die Phase (A,B,C,D) abgeschaltet. Die nächste Abbildung zeigt den Stromverlauf, wenn A high und B low ist. Der Spulenstrom fließt laut Kurve X. Wenn die Spannung am Sensewiderstand die der



Referenzspannung übersteigt, wird der Schalttransistor zwischen Spule und Sensewiderstand (Q4) gesperrt. Die gespeicherte Energie der Spule baut sich durch einen zirkulierenden Strom durch Q1 und D3 ab (Kurve Y). Im ersten Moment würde es logischer erscheinen, einfach A auf low zu setzen, um die Spule abzuschalten. Der Vorteil wenn man zum Abschalten B high setzt besteht darin, dass sich die Energie der Spule nicht über den Sense-Resistor, was zusätzliche Verlustleistung für den Widerstand bedeuten würde abbaut (Kurve Z).

Inhibit Chopping:

Der Stromchopper wird in diesem Mode mit den Inhibit-Signalen realisiert. Dabei behalten die Phasen den Zustand laut Bitmuster entsprechend Halbschritt/ Vollschritt. Sobald die Spannung am Sense-Resistor die Referenzspannung übersteigt, wird das Inhibit Signal auf low gesetzt, wodurch alle 4 Transistoren sperren. In der Grafik ist ersichtlich, wenn die Spule über Q1 u. Q4 bestromt wurde, fließt der Recirculationsstrom über D2 und D3.



○ **Auswahl des Chopperverfahrens:**

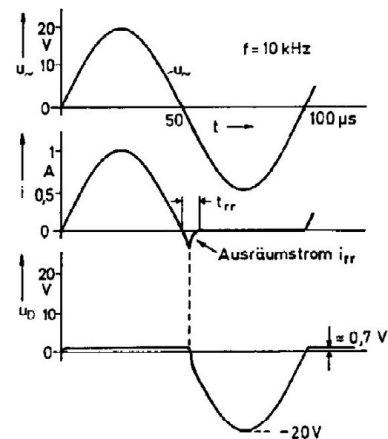
Die Auswahl des Chopperverfahrens ist abhängig vom Motortyp. Bei **Bipolar- Motoren** (solche verwende ich!) ist das Verfahren generell beliebig zu wählen. Vorteilhaft ist die Phase-Choppermethode, weil dadurch die Verlustleistung von R_s vermindert wird. Bei **Unipolaren Motoren** darf ausschließlich Inhibit-Chopping verwendet werden.

Auswahl der Fast Recovery- Dioden für L298:

Die Spezifikationen laut Datenblatt von L298 sind:

- $U_f < 1,2V$ bei $2A$ I_f
- t_{rr} (Reverse Recovery-Time) $< 200ns$

Für diese Anwendung leicht ausreichend ist die Diode BYV27/55. Die Reverse Recovery-Time für $I_f = 0,5A$ und $I_r = 1A$ beträgt maximal 25ns. Die Forward- Voltage liegt auch innerhalb des erlaubten Bereichs. (max 1,07V bei 3A) Genauere Informationen siehe Datenblatt!



5.2.2 Temperaturüberlastschutz der Schrittmotorkarte:

Der Temperaturüberlastschutz wurde mit einem PTC realisiert. Der Grund, wieso ich mich für einen PTC und keinen NTC entschieden habe, ist der, dass damit keine Temperatur gemessen sondern nur bei zu hoher Temperatur abgeschaltet werden soll. Ein Kaltleiter eignet sich für einen Zweipunktregler besser als ein Heißleiter, da ein PTC bei seiner Sprungtemperatur eine starke Widerstandsänderung hat. Als Sprungtemperatur wurde 100°C gewählt (maximale T_j von L298 beträgt 150°C) Wird der Überlastschutz aktiv, bleibt die Schrittmotorkarte solange ausgeschaltet, bis die Temperatur des Kühlkörpers wieder 60°C erreicht hat. Währenddessen wird der Kühlkörper zusätzlich



durch einen Lüfter gekühlt. Die Ein-Ausschalt-Hysterese wird durch den nachgeschalteten TTL- Schmitttrigger 74LS14 erreicht. Eine zu hohe Temperatur ist allerdings nicht die einzige Möglichkeit, die SM-Brücke abzuschalten. Mit dem Busy-Signal, welches mit dem Temperaturüberlastschutzsignal verodert ist, ist es möglich, z.B. über einen MC die SM-Brücke ein bzw. auszuschalten.

5.2.3 Referenzspannungsaufbereitung:

Wie schon erwähnt ist für den Stromchopper eine Referenzspannung nötig. Die Funktion und die Aufgaben der Referenzspannungsaufbereitung wurden bereits im Punkt 2 unter Boost/Sleep-Signal behandelt. Den dazugehörigen Schaltplan finden sie auf [Seite 39](#). Der Phasenstrom lässt sich getrennt für jede Achse mit einem Spindeltrimmer einstellen. Der Einstellbereich liegt im Boostfall (Transistor leitet) von 0,6 – 2A. Im Sleepmode liegt der Bereich von 0,47- 1A.

5.2.4 Bohrer und Absaugrelais:

Für Bohrmaschine und Absaugung ist jeweils ein Relais (max. 10A) zum automatischen Ein/Ausschalten vorgesehen. Beide Relais sind per Software steuerbar. Angeschlossen werden die zu schaltenden Leitungen an der Steckerleiste MT6-4Y. Eine weitere Anschlussmöglichkeit für das Bohrspindelrelais besteht bei der Schraubklemme AK505. Zu beachten ist, dass das Relais mit der Brückenversorgungsspannung versorgt wird. Die Betriebsspannung der Relais muss daher an die Spannung bei Veränderung der Brückenspannung angepasst werden. (regulär 12V-Relais)

5.2.5 Spannungsversorgung:

Da ich mir die Arbeit, einen Netzteil zu dimensionieren, ersparen wollte, setzte ich in der ersten Planungsphase ein Computernetzteil ein. Dieses lieferte mir sowohl für die Logik die nötigen 5V als auch für die Schrittmotorbrücke den nötigen Strom von 6A, jedoch nur eine Spannung von 12V. Ich habe die Erfahrung gemacht, dass sich mit der Brückenversorgungsspannung von 12V nur sehr kleine Schrittgeschwindigkeiten (max. 500Hz) realisieren lassen ohne mit Schrittverlusten rechnen zu müssen. Dies wirkt bei der X- Achse (Antrieb über Gewindestange) sehr störend (langsam). Daher habe ich im Nachhinein eine 42V Stromversorgung (begrenzt durch den Brückentreiber L298) in mein 19-Zoll Rack eingebaut. Ich kann jedenfalls bestätigen, dass sich die aufzuwendenden Kosten von rund 600 ATS gelohnt haben. Mit dieser Versorgung sind jetzt auf der X- Achse Taktgeschwindigkeiten von bis zu 3kHz möglich. Der Computernetzteil dient noch immer zur Versorgung der Logik (5V) und zur Versorgung der Lüfter (12V).

Wichtig: Wird die Schrittmotorkarte nicht mehr über das Bussystem (12V), sondern über eine externe Spannungsversorgung (42V) versorgt, so muss die Leiterbahn der 12V-Zuführung beim VG64-Stecker unterbrochen werden, da ansonsten ein Kurzschluss vorhanden wäre.

Die Logik-Versorgungsspannung ist mittels Z-Diode gegen Spannungsspitzen geschützt. Beide Versorgungsnetze sind mit Stützkondensatoren laut Application Notes gestützt.



42V-Netzteil (30V-Trafo)

Schaltpläne, Berechnung, Layout, Bestückungsplan und Stückliste zur Schrittmotorkarte finden sie von Seite 36 - 46!

5.3 PC-Interface:

Das PC-Interface hat die Aufgabe, die einzelnen Schnittstellensignale je nach Wahl der Schnittstelle zu schalten und zu regenerieren. Zusätzlich werden die einzelnen Signale über Pull-Up bzw. Pull-Down-Widerstände auf ihre Grundeinstellung gebracht. Als Schnittstellen zwischen PC und Bohrplotter sind die Parallele und die Serielle Schnittstelle wählbar. Die Umschaltung erfolgt mit einem Schalter, der über das Latch 74LS75 die tristatefähigen Bustreiber 74LS244 schaltet. Der aktuelle Zustand wird jeweils durch eine leuchtende Led neben dem Schnittstellenstecker signalisiert. Allerdings wurden die beiden Leds auf der Platine falsch angeordnet, sodass diese ausgekreuzt gehören.

Die Parallele Schnittstelle dient für eine direkte Ansteuerung des Plotters. Dabei wurden allen Daten- und Steuerleitungen der Centronics -Schnittstelle bestimmte Funktionen zugewiesen. Z.B. Data 0 -> Richtung X, Data1 -> Takt X u.s.w. Die einzelnen Steckerbelegungen und Pinzuweisungen finden Sie auf [Seite 64 & 65](#).

Die Serielle Schnittstelle ist für eine zukünftige Mikrocontrollerplatine vorgesehen. Die Bohrdaten würden dann über die Serielle Schnittstelle ins RAM übertragen werden. Der MC hätte dann die Aufgabe, die Bohrdaten auszuwerten und die Schrittmotorkarte dementsprechend anzusteuern. Weiters würde der MC auch zuständig für die Gabellichtschranken, Statusleds, Funktionstasten und eventuell eine LCD-Anzeige sein. Der Vorteil einer MC-Steuerung wäre die Unabhängigkeit der Rechnergeschwindigkeit in Bezug auf die Ansteuerung.

Als Linetransceiver für die RS232 wird der Baustein MAX232 verwendet. Die Kondensatoren für die Ladungspumpe des MAX232 sind laut Datenblatt gewählt.

Als Verbindung zwischen Busplatine und Interface-Platine dient ein 26pol. Flachbandkabel. Für die Stützkondensatoren der ICs wurden 100nF MK-Folienkondensatoren gewählt.

Die Interface-Platine ist an der Rückseite des 19-Zoll-Gehäuses oberhalb der Plotter-Ausgabe-Platine montiert.



Interface- Platine

[Schaltpläne, Berechnung, Layout, Bestückungsplan und Stückliste zum PC-Interface finden sie von Seite 47 - 53!](#)

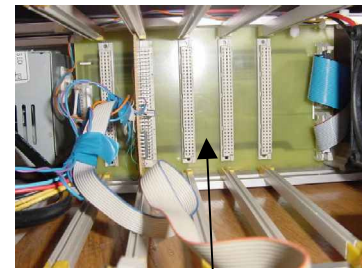
5.4 Plotter- Ausgabe:

Die Plotter-Ausgabe-Platine ist zuständig für die Adaption der Stecker nach außen. Als Stecker für die Verbindung zwischen Schrittmotorkarte und Plotter-Ausgabe-Platine dient ein Schraubklemmenstecker. Von der Rückplatte zu den Schrittmotoren wird ein 48pol. DIN 41612-Stecker Typ F (Hochstromausführung bis 6A der DIN-Reihe) verwendet. Auf der Platine ist auch die nötige Elektronik für die Gabellichtschranken untergebracht. Für jede Achse ist ein Referenzgabellichtschranken und ein Endgabellichtschranken vorgesehen. Jeder Gabellichtschranken wird mittels 3 Leitungen angeschlossen. (Anode von Led, Kollektor von Phototransistor und Kathode von Led verbunden mit Emitter von Phototransistor angeschlossen auf Masse) Die Vorwiderstände der Leds bzw. die Kollektor-Widerstände der Phototransistoren sind als Widerstandsnetzwerke ausgeführt. Als Verbindung der Gabellichtschrankensignale an die Busplatine dient ein 16pol. Flachbandkabel.

[Schaltplan, Berechnung, Layout, Bestückungsplan und Stückliste zur Plotter-Ausgabe finden sie von Seite 53 - 58!](#)

5.5 Busplatine:

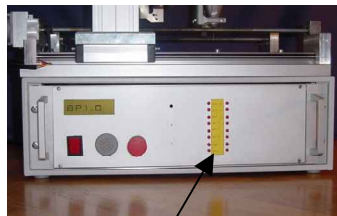
Die Backplane besitzt 5 Steckplätze, welche 1:1 verbunden sind. (d.h. dass Pin 1 von Steckplatz1 mit Pin1 von den anderen Steckplätzen verbunden sind) Als Stecker sind DIN 41612-TypeC Stecker in Verwendung. An die Busplatine ist die Interfaceplatine, Plotterausgabeplatine und der Computernetzteil angeschlossen. Auf der Backplane stehen die Spannungen 5V, -5V, 12V, -12V zur Verfügung. Bis auf 8 Pins sind alle 64 Pins vordefiniert. Die 8 Pins können für andere Funktionen verwendet werden, die genaue Pinvergabe finden sie auf Seite 66. **Schaltplan, Layout und Bestückungsplan der Busplatine finden sie von Seite 59 - 61!**



Backplane:

5.6 Zusatzplatine:

Da zwei kleine Funktionen (Piepser und Gabellichtschrankverknüpfung) auf der Schrittmotorkarte nicht mehr Platz hatten, musste eine kleine Zusatz-Lochrasterplatine gelötet werden. Der Warnpiepser ist per Software über ein vorgesehene Bussignal schaltbar. Auf der Zusatzplatine befindet sich der nötige Schalttransistor. Wie erwähnt sind auf der Busplatine alle 6 Gabellichtschrankensignale vorhanden. Da die Parallele Schnittstelle nur eine begrenzte



Ledleiste:

Anzahl von Eingängen hat, werden die 3 Endschalersignale verodert zu einem OOS (Out of Space)-Signal zusammengefasst. Auf der Frontseite des 19-Zoll Gehäuses befindet sich eine Statusleiste. Die Statusleiste ist als Multiplexplatine aufgebaut und stammt von einem Post-Messgerät. Der Multiplexbetrieb ist ideal für einen Mikrocontroller. Da zur Zeit noch keine MC- Platine vorhanden ist, wurden auf der Multiplexanzeige die Enable-Signale der Latches auf high gesetzt, sodass die Latches transparent sind.

Die Bussignale, welche auf der Statusleiste ersichtlich sein sollen, wie z.B. die Boost/Sleep-Signale, Gabellichtschrank u.s.w., wurden von der Zusatzplatine auf die Ledleistenplatine geführt.

Schaltplan der Zusatzplatine und Pinbelegung der Ledleiste finden sie auf [Seite 67](#).

5.7 Computer-Netzteil:

Das Computer-Netzteil ist für die komplette Logikversorgung und die Versorgung der Lüfter zuständig.

Technische Daten:

- Primär: 230V; Leistung:200W
- sekundär:12V/6A; -12V/0,5A; 5V/18A; -5V/0,5A

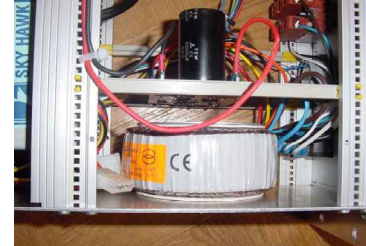
Alle Spannungen werden ins Bussystem eingespeist. Der Netzteil lässt sich durch den Hauptschalter oder durch den NOT AUS-Schalter abschalten.

5.8 42V-Netzteil für Schrittmotorbrücke:

Der Netzteil wurde schon kurz unter dem Punkt Schrittmotorkarte/Stromversorgung erwähnt. Jetzt folgt die technische Betrachtung des Netzteils. Die Anforderung an den Netzteil ist, eine gut geglättete, jedoch nicht geregelte Ausgangsspannung zu liefern. Die konstante Ausgangsspannung ist wenig von Bedeutung, weil es lediglich die Steigung der e-Kurve beim Stromchopper beeinflusst. Es gilt nur das Maximum von 46V, welches durch die Brückentreiber begrenzt ist, nicht zu überschreiten. Benutzt man einen 30V-Trafo ergeben sich nach dem Brückengleichrichter 41V Gleichspannung ($\Rightarrow 30 \cdot \sqrt{2} = 42,4V - 2 \cdot 0,7V = 41V$). Bei einer Netzschwankung von 10% (Uprim. max. 253V) würde dies eine Spannung von 45V Gleichspannung bedeuten und daher immer noch im akzeptablen Bereich liegen. Die

Voraussetzung dafür ist, dass sich der Trafo nie im Leerlauf befindet, da ansonsten die Ausgangsspannung um rund 5% höher liegen würde und daher zu hoch wäre. Dadurch, dass sich der Trafo beim Einschalten durch das Laden der Glättungskondensatoren unter Vollast befindet, stellt der Leerlauf kein Problem dar. Spannungsspitzen am Netz können durch einen primärseitigen Varistor eliminiert werden. Wird jedoch der Bohrplotter innerhalb kürzester Zeit ein- und wieder ausgeschaltet, so besteht die Gefahr für Überspannung, weil die Glättungskondensatoren noch geladen sind. Daher empfiehlt es sich, auch sekundärseitig einen Varistor oder eine Z-Diode parallel zu schalten.

Um die nötige Leistung des Trafos zu bestimmen zu können muss die Stromaufnahme des Trafos bekannt sein. Rechnerisch ist dies ziemlich komplex, wenn man bedenkt, dass durch den Stromchopper und der Betriebsart ein sehr zeitabhängiger Strom fließt. Daher versuchte ich die Stromaufnahme durch eine primärseitige Messung beim Computernetzteil zu ermitteln. (Schrittmotorbrücke an 12V angeschlossen) Durch sekundärseitige Rückrechnung und unter Vernachlässigung der 5V und des Wirkungsgrads des Schaltnetzteils wurde ein Strom von 7A ermittelt. $30V \cdot 7A = 210VA$ Gewählt wurde letztendlich der nächstgrößere Typ mit 225VA.



Glättungskondensatoren: Je nach Stromaufnahme der Motoren sollte man dem Netzteil mindestens $10000\mu F$ gönnen. Als Brückengleichrichter wird KBPC801 verwendet. (für 8A)
Die Platine wurde wie die Zusatzplatine auf Lochraster aufgebaut.

6. Messungen an der Schrittmotorkarte:

6.1 Schrittgeschwindigkeiten:

Die interessanteste Messung ist natürlich die maximale Geschwindigkeit der Schrittmotoren. Diese ist abhängig von:

- Der Brückenversorgungsspannung: Der extreme Unterschied ist im Vergleich der 2 exponentiellen Kurven für 12V und 42V der Stromzunahme der Motorspulen auf **Seite 74** gut ersichtlich.
- Der Motorbelastung: Der Drehmoment ist im Ruhezustand (Haltedrehmoment), wo sich noch der komplette Spulenstrom bilden kann am größten und nimmt mit steigender Geschwindigkeit ab. Zusätzlich wird die Drehmomentkennlinie durch Motorresonanzeinknicke geprägt. (Bei der X-Achse ist die Motorbelastung durch die Gewindestange wesentlich kleiner als bei der Y- und Z-Achse)

Der Stromverlauf in den Motorspulen lässt sich am besten über die Sensewiderstände des L298 messen. Zum Triggern eignen sich die 4 Phasen vom L297 (A, B, C, D) sehr gut.

Die einzelnen Messungen wurden für verschiedenen Brückenspannungen und für verschiedene Phasenströme durchgeführt. Dabei ist gut zu sehen, dass sich bei zu hoher Geschwindigkeit der Strom nicht mehr zur Gänze ausbildet. Der Schaltimpuls schrumpft zusammen, bis der Motor zum Stehen kommt. Die Messungen wurden im Halbschrittmode durchgeführt. Daher ist durch die Shaping- Funktion in der Mitte die $\sqrt{2}$ - Erhöhung.

Die einzelnen Mess-Plots sind ab **Seite 68** zu sehen.

Achsgeschwindigkeiten: Die Messung der maximalen Achsgeschwindigkeit ist ein subjektives Ergebnis, da ich keine Grenzwerte für die Schrittmotortechnik weiß. Die Angaben sind die Geschwindigkeiten, bei denen sich die Impulsamplitude max. um 10% verkleinert hat.

Achse:	Max. Taktfrequenz	Max. Achsgeschwindigkeit
X-Achse:	3kHz	9,3mm/s
Y- und Z-Achse: ₁₎	500Hz	63mm/s

1) mit Rampe für Beschleunigung und Abbremsung!

6.2 Messung der Referenzspannung:

Die 3 Achsen wurden jeweils über den Spindeltrimmer auf 2A gestellt, sodass sich eine Referenzspannung von 940mV ergibt. Die Phasenstromangabe am Motor beträgt nur 1,4A/Phase. Diese Angabe bezieht sich jedoch auf den Gleichstromfall, der bei mir nicht der Fall ist. Daher ist es kein Problem, die Motoren mit 2A (im Boost-Mode) zu betreiben.

6.3 Messung des Temperaturüberlastschutzes:

Der Potentiometer wurde so eingestellt, dass sich die Schrittmotorbrücke bei einer Kühlkörpertemperatur von 100°C ausschaltet und bei 60° wieder einschaltet.

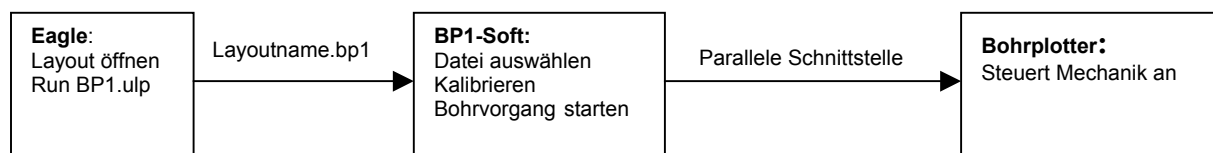
6.4 Messung der Brückenspannung:

Die gleichgerichtete Brückenspannung unter Belastung beträgt 42,2V (Sollte laut Berechnung nur 41V sein!). Dies ist auf die erhöhte Netzspannung von 234V zurückzuführen. Die gemessene Leerlaufspannung (nach Gleichrichtung) betrug 44,4V. Die Differenz beträgt daher zwischen U_0 und U_1 genau 5%!

7. Software im Detail:

7.1 Allgemein:

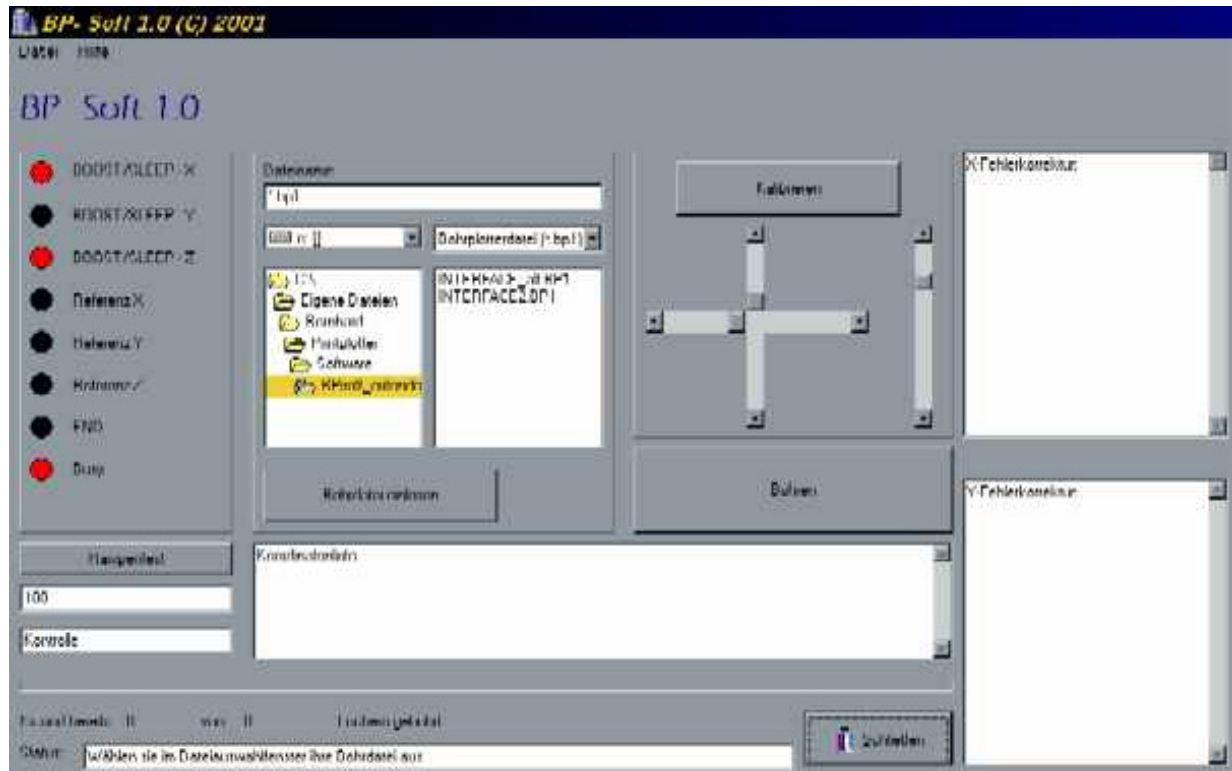
Die Software ist ein sehr wichtiges Element dieses Projekts. Zu beachten ist, dass sie das flexibelste Modul ist. Durch Programmierung der richtigen Software lässt sich der Bohr-Fräspotter für die verschiedensten Aufgabenstellungen anwenden. So ist es z.B. jetzt schon möglich, mit meinem Plotter über Kcam, einer Software (Freeware) aus dem Internet Teile zu fräsen. Mit diesem Programm lassen sich die verschiedensten Dateiformate (Dxf, HPGL...) zu einem G-Code umwandeln, welcher dann auch über die Parallele Schnittstelle ausgegeben werden kann. Die Software ist so flexibel, dass man selbst die einzelnen Funktionen den Pins der Parallelen Schnittstelle zuweisen kann. Der einzige Nachteil von diesem Programm ist, dass man nur eine einheitliche Taktfrequenz für alle 3 Achsen einstellen kann. Dies ist für meinen Aufbau nicht optimal, weil ich die X-Achse mit einer Gewindestange, jedoch die Y- und Z-Achse mit einem Zahnriemen antreibe, und daher stark unterschiedliche Taktraten nötig sind. (Nähere Infos über die Frässoftware Kcam sind auf meiner Homepage www.bohrplotter.at.gs zu finden!)



7.2 Meine Software BP-Soft1:

Mein Aufgabenbereich liegt jedoch „nur“ darin, eine Platine vollautomatisch zu bohren. Wenn eine Eagle- Schaltung fertig gelayoutet wurde, wird mit der ULP(user language programm)-Datei BP1.ulp eine Datei mit dem Namen Layoutname.bp1 generiert. (Eagle stellt ULP-Dateien für bestimmte Anwendungen wie z.B. Bohrkoordinaten auslesen, Pads verändern u.s.w. zur Verfügung.) Die Datei BP1.ulp ist die modifizierte Drill.ulp von Eagle. Dabei wurden die Maßeinheiten von inch auf mm sowie der Dateityp von .drh auf .bp1 geändert und die Anordnung der Bohrkoordinaten verändert. Die modifizierte Drill.ulp (BP1.ulp) ist auf [Seite 96](#) zu sehen. Die Erstellung der Bohrdatei erfolgt in der Befehlszeile von Eagle mit dem Befehl **run BP1.ulp** . Die ULP-Datei muss sich für diesen Vorgang im Eagleverzeichnis in dem Ordner ULP befinden. In der Bohrdatei sind listenförmig die X/Y-Koordinaten sowie der Bohrdurchmesser für jedes Pad angegeben. Diese Datei wird in meiner Software mit Hilfe des Dateiauswahlfensters ausgewählt. Nach Drücken des Buttons „Bohrdatei einlesen“ wird die Bohrdatei eingelesen und nach Koordinaten und Bohrdurchmesser sortiert. Der nächste Schritt ist die Kalibrierung der Achsen. Durch drücken des Buttons „Kalibrieren“ bewegt sich jede Achse bis ans Ende, welches durch den jeweiligen Gabellichtschranken erkannt wird und fährt dann mit dem nötigen Offset zum relativen Nullpunkt (entspricht Lötnagel, auf welchen die Platine aufgesteckt wird) Die Z-Achse bleibt 1cm über dem Lötnagel aufgrund der unterschiedlichen Bohrerlängen stehen und muss dann manuell kalibriert werden. Somit kann der Bohrvorgang schon gestartet werden. Durch einen Buttonclick auf die Schaltfläche „Bohren“ wird der Bohrvorgang gestartet. Der Bohrablauf erfolgt schleifenförmig in X-Richtung. Sobald alle Löcher von einem Bohrdurchmesser gebohrt wurden, wird der Bohrablauf gestoppt und falls nötig eine Softwaremeldung für Bohrerwechsel ausgegeben. Nach dem Bohrerwechsel wird durch einen Click auf die Message-Box der Bohrvorgang fortgesetzt. Die Z-Achse muss wieder manuell kalibriert werden. Durch Beendigung der manuellen Kalibrierung werden die nächsten Löcher gebohrt. Der Bohrstatus ist in der Progressbar ersichtlich. Die Anzahl der gebohrten Löcher ist aus der Progressbar zu entnehmen. Links ist die gleiche Statusbar wie auf der Frontplatte vom Bohrplotter. Unten ist eine Statusleiste, welche dem Benutzer immer wichtige Daten übermittelt. In den zwei rechten Textfeldern wird die Fehlerkorrektur der Y- und X-Achse protokolliert.

Bild von BPSoft1:



7.3 Die Programmiersprache C++ Builder:

Die Auswahl der Programmiersprache war für mich ziemlich leicht. Da ich nur in der 2. und 3. Klasse HTL Borland-C lernte, musste ich mein Programm in C schreiben und da ich kein DOS-Programm, sondern ein Programm mit Windowsoberfläche schreiben wollte, musste ich C++ Builder lernen. (Ein Dos-Programm wäre auch aus programmieretechnischen Gründen nicht möglich gewesen, da Win98 keinen direkten Zugriff, wie es mit dem C-Befehl `outportb` der Fall wäre zu! Versucht man unter Win98 in der Eingabeaufforderung mit `outportb` auf die Parallele Schnittstelle zu schreiben, wird dieser Schreibvorgang von Win98 ignoriert! Der selbe Vorgang glückt jedoch bei Verwendung eines älteren Rechners und Win3.11!) An dieser Stelle möchte ich mich nochmals herzlich bei Scarlet Christian für die Einführung in den C++ Builder und die Hilfsbereitschaft beim Programmieren bedanken. Nachdem ich meine ersten Testprogramme geschrieben hatte, machte ich mich auf die Suche nach einem Parallel-Interface Komponent. (Um vom C++ Builder auf die Parallele Schnittstelle schreiben zu können, ist ein Komponent notwendig. Dieses besitzt eigens definierte Befehle für die Ausgabe.) Erst nach mehrmaligen Suchversuchen hatte ich ein passendes Package gefunden. Während der einzelnen Suchen im Internet machte ich die Erfahrung, dass Komponenten für Delphi um einiges leichter zu bekommen sind. Die Komponente von meinem Package hatte nur den Nachteil, dass kein Pinzugriff möglich war. Das Maskieren der Pins blieb daher nicht erspart. Zufälligerweise fand ich einen Tag später ein Komponent, bei dem auch ein Pin-Zugriff möglich ist. Die Internetadresse lautet: <http://www.entechtaiwan.com/tools.htm> Damit wurde das Programmieren noch vereinfacht. Das Package ist für C++ Builder 1-4 erhältlich. (Ich programmiere mit C++ Builder 4)

Im C++ Builder wird das Programm nicht in einem Hauptprogramm, sondern in vielen einzelnen Routinen geschrieben. Jedes Objekt (Button, Textfeld) besitzt verschiedene Ereignisse. (z.B. Buttonclick) Somit werden für die verschiedenen Ereignisse eigene Routinen geschrieben. Die Programmiersprache C++ Builder ist eine objektorientierte Sprache. Eine

Eigenschaft von einem Objekt zu verändern, erfolgt folgendermaßen: *Objektname->Eigenschaft = Wert*; Stark unterschiedlich gegenüber dem normalen C ist auch die Deklaration von Unterprogrammen. Im C++ Builder besitzt jede Form (Programmoberfläche) eine eigene Headerdatei, in der alle Objekte der Form und alle Deklarationen der programmierten Ereignisse enthalten sind. Möchte man im C++ Builder ein Unterprogramm deklarieren, muss dies in der dafür vorgesehenen Headerdatei erfolgen.

7.4 Parallel- Komponent :

Der erste Schritt für die Verwendung des Komponents ist es, den Port mit einem Befehl zu öffnen:

```
IOPort->OpenDriver();
  if (!IOPort->ActiveHW)
  {
    MessageBeep(0);
    Application->MessageBox("Driver not found!", " Warning! ",IDOK);
    exit;
  }
```

Komponent- Befehle:

Mit dem Komponent TVIC-Port lassen sich die einzelnen Pins, sowie ein komplettes Register beschreiben. Weiters ist es möglich, Steuersignale wie z.B.: Strobe oder Busy mit einem eigenen Befehl anzusprechen. Anschließend sind nur von mir verwendete Befehle angeführt.

property Port[Index: Word]: Byte; (public, read, write);

Value of the Port array is of type Byte.

z.B.: auf LPT1 in das Data register (378h) den Wert 128 schreiben:

`IOPORT->Port[0x378] = 128 ;`

128 dezimal = 10000000 binär, d.h. dass bis auf D7 alle Datenleitungen low sind.

property Pin[Index:1..25] : Boolean; (public, read, write)

z.B: Setzen von Boost/Sleep X (Pin16): **`IOPort->Pin[14]=false;`**

Abfragen von Referenz X (Pin11): **`if (IOPort->Pin[15] == true) Anweisung;`**

7.5 Grundlagen zur Parallelen Schnittstelle:

Um auf eine Schnittstelle schreiben zu können, müssen die Adressen sowie der Aufbau der Register bekannt sein. Unabhängig von ihrer Lage im Adressraum, weisen alle parallelen Schnittstellen (Druckerports) ein einheitliches Register-Interface auf, das aus drei Ports besteht. Sie belegen die ersten drei Portadressen im Adressbereich der Karte, also beispielsweise 378h, 379h und 37Ah bei der ersten parallelen Schnittstelle. Die Bits in den Schnittstellenregistern stehen in direkter Verbindung mit den Übertragungsleitungen eines Centronics- Kabels. Wird in ein solches Register-Bit der Wert 1 geschrieben, wird dadurch sofort auch die korrespondierende Leitung unter Strom gesetzt (Pegeländerung von 0 =>1). Andersherum wird die Leitung wieder auf low gelegt, wenn das Bit auf 0 zurückgesetzt wird. Grundsätzlich gilt, dass die Leitung solange ihren Status behält, bis das jeweilige Bit per Software verändert wird. Zu beachten ist, dass einige Steuerleitungen der Parallelen Schnittstelle lowaktiv sind.

Die Parallele PC-Schnittstelle (Centronics- Schnittstelle) ist in 3 Kanälen aufgeteilt:

- **Statuskanal:** Adresse: 0379h (für LPT1)
Zugriff: nur Lesen

Bit:	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Pin:	Pin11	Pin10	Pin12	Pin13	Pin15	x	x	x
Verwendung:	Ref X	Busy	Ref Y	Ref Z	Endschalter			

x bedeutet : Dieses Bit wird nicht verwendet

Verwendung: Die Normsignale werden für die Bohrplotteranwendung bestimmte Funktionen zugewiesen.

- **Steuerkanal:** Adresse: 037Ah (für LPT1)
Zugriff: Lesen und Schreiben

Bit:	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Pin:	x	x	x	Inter.	Pin17	Pin16	Pin14	Pin1
Verwendung:					B/S Y	B/S X	Piepser	B/S Z

Inter.: Interrupt 7 freigeben (interne Anwendung)

Wird dieses Bit auf 1 gesetzt, so kann das Acknowledge –Signal von der peripheren Einheit den Interrupt 7 auslösen. Dieses Bit hat für die Bohrplotteranwendung keine Bedeutung.

- **Datenkanal:** Adresse: 0378h (für LPT1)
Zugriff: Schreiben und Lesen

Bit:	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Pin:	Pin9	Pin8	Pin7	Pin6	Pin5	Pin4	Pin3	Pin2
Verwendung:	Absaugung	Bohrer	CLK Z	DIR Z	CLK Y	DIR Y	CLK X	DIR X

Einen Überblick der Pindefinitionen finden sie auf [Seite 64](#) !

Bedeutung der Schnittstellensignale: (für Drucker)

- **Strobe** (lowaktiv): Pin1 (Ausgabe)
Dieses Signal wird auf '0' gesetzt, wenn die Daten von der peripheren Einheit übernommen werden sollen.
- **Data** Bit 0-7: Pin 2-9 (Ausgabe)
Dies sind die Daten, die ausgegeben werden sollen. Diese Daten können wieder eingelesen werden, jedoch dürfen keine Signale an diese Pins von außen angelegt werden.
- **Acknowledge:** (lowaktiv): Pin10 (Eingabe)
Mit diesem Signal quittiert die periphere Einheit die Übernahme der Daten.
- **Busy:** Pin11 (Eingabe)
Liegt diese Leitung auf High-Pegel, kann die periphere Einheit momentan keine Daten annehmen. (Unabhängig von meiner Busy-Funktion!)
- **PE**(Paper End): Pin12 (Eingabe)
Mit diesem Signal wird das Papierende angezeigt. Einige Druckertypen liefern das Signal nicht.
- **Select out:** Pin13 (Eingabe)
High-Pegel an diesem Pin bedeutet, dass eine periphere Einheit vorhanden ist, bzw. , dass die Einheit ONLINE ist.
- **Auto Feed** (lowaktiv): Pin14 (Ausgabe)

Low-Pegel veranlasst den Drucker, automatisch ein Line-Feed auszuführen. (Einige Drucker werten das Signal nicht aus.)

- **Error** (lowaktiv): Pin15 (Eingabe)
Dieser Pin wird auf 0 gezogen, wenn die periphere Einheit einen Fehler meldet.
- **Initialize Printer** (lowaktiv): Pin16 (Ausgabe)
Mit diesem Signal wird das angeschlossene Gerät zurückgesetzt. (entspricht Reset!)
- **Select In** (lowaktiv): Pin17 (Ausgabe)
Hiermit kann eine Bereitschaftsanforderung gestellt werden.
- **Ground**: Pin18-25: Masseleitungen

Überblick der Schnittstellensignale: Die Basisadresse ist für LPT1 0378hex und für LPT2 0278hex. (Basisadresse + 1 -> ein Register höher z.B.: 378h -> 379h!)

Computer Pin (25 Pol SubD)	Drucker Pin (Centronics)	Leitungsname	Bedeutung	Basisadr.	Bit
1 ==> 1		STROBE (INV)	Zeigt Übertragung an	Basisadr. + 2	0
2 ==> 2		D0	Datenleitung Bit 0	Basisadr.	0
3 ==> 3		D1	Datenleitung Bit 1	Basisadr.	1
4 ==> 4		D2	Datenleitung Bit 2	Basisadr.	2
5 ==> 5		D3	Datenleitung Bit 3	Basisadr.	3
6 ==> 6		D4	Datenleitung Bit 4	Basisadr.	4
7 ==> 7		D5	Datenleitung Bit 5	Basisadr.	5
8 ==> 8		D6	Datenleitung Bit 6	Basisadr.	6
9 ==> 9		D7	Datenleitung Bit 7	Basisadr.	7
10 <== 10		ACK	Letztes Zeichen Angenommen	Basisadr. + 1	6
11 <== 11		BUSY (INV)	Drucker beschäftigt	Basisadr. + 1	7
12 <== 12		PE	Drucker hat kein Papier mehr	Basisadr. + 1	5
13 <== 13		SLCT	Drucker ist Online	Basisadr. + 1	4
14 ==> 14		AUTO FEED (INV)	Automatisches CR nach LF	Basisadr. + 2	1
15 <== 32		ERROR	Fehler bei Datenübertragung	Basisadr. + 1	3
16 ==> 31		INIT	Reset des Druckers herbeiführen	Basisadr. + 2	2
17 ==> 36		SLCT IN (INV)	Drucker Online schalten	Basisadr. + 2	3
18 - 25 <==> 19 - 30		GND	Masse		

Programm- Listing ab Seite 82!
Strukturgramme der wichtigsten Routinen finden sie ab Seite 93!

8. Zeitplan:

Obwohl der zu Schulanfang festgelegte Zeitplan nicht zu Gänze eingehalten wurde, hat sich das Projekt mit einwenig Überstunden in diesem Zeitraum realisieren lassen.

18.9.2000: Von 8:00 bis 10:00 Einführung in das Projektlabor 5; Rückplatte für 19“-Rack gefertigt (Ausnahme für Computernetzteil); Zusatzwelle für Z-Achse auf richtige Länge gedreht und anschließend montiert

25.9.2000: Überlegung für Bohrmaschinenhalterung und für die Grundplatte für das Y-Z-Modul; Fertigung der Grundplatte und der Bohrmaschinenhalterung; Projektbesprechung

2.10.2000: Feinjustierung der Z-Achse (durch 3. Welle war eine Schwergängigkeit vorhanden, die großteils behoben wurde); Montagearbeiten bei den Achsen; Komplette Berechnung der Schrittmotortreiberkarte (Referenzspannungsaufbereitung)

9.10.2000: Berechnung des PC- Interfaces; Testschaltung für Gabellichtschranke aufgebaut; Zeichnen des Schaltplans für Plotterausgabe

16.10.2000: Da alle Schaltpläne fertig sind, wurden mit Herrn Höfer noch alle Schaltungen besprochen. Es wurden noch kleine Modifikationen durchgeführt. Z.B. bei Interface-Auswahl statt Taster Schalter; Beginn mit dem Layout Plotterausgabe

23.10.2000: X-Mechanik; Bearbeitung des Aluminiumprofils; T-Nutsteine für Grundplatte gefräst.; Fixierung der Y/Z-Achse; Montage der Linearführungen -> Mechanik bis auf X-Antrieb fertig; Bei Y/Z- Modul Montage der Bohrmaschinenhalterung

30.10.2000: X- Antrieb: Besprechung mit Herrn Hackenbuchner; Planung der Lagerblöcke; Beginn mit der Fertigung der Frästeile; Gewindespindel zuschneiden und bemessen;

6.11.2000: Endfertigung der Lagerblöcke; Fertigung der Gewindespindelmutter, Alublöcke für Motor und Motorblock; Drehen der Gewindespindel und der Verbindungsbolzen von Motor und Gewindestange; Verbesserung der Zahnriemenspannvorrichtung bei der Y- und Z-Achse; Abdeckplatten für Aluprofil; Motorjustierung fräsen;

13.11.2000: Gewinde bohren für Motorhalterung; Letzte Endfertigung von Motorblock und Gewindestange; Spannvorrichtung des Zahnriemens der X- Achse; Montage u. Justierung des X- Antriebs; Mechanik komplett fertig!

20.11.2000: Layout Schrittmotorkarte 1. Teil; Anordnung der Bauteile, Routen der Versorgung und Ausgangsleitungen; Bei Kühlkörper muss eine Ausfräsung gemacht werden (Relais zu hoch, jedoch sollen sie auf die Ausgangsseite positioniert werden!)

27.11.2000: Layout-Schrittmotorkarte 2. Teil; Ausrichtung der Leitungen mit zu geringem Abstand; Vias verändern; Leitungen anpassen (Dicke)

4.12.2000: Ätzen der Schrittmotorkarte (großes Problem beim Ätzen, Laserdrucker deckt zuwenig auf der Transparentfolie) 2. Versuch mit Unterbelichtung (2min 10) klappt -> komplette Fertigstellung der Platine zuhause

11.12.2000: Vergebliche Fehlersuche bei der Z-Achse (Schrittmotorkarte)-> U_{ref} stieg sobald L297 in den Sockel gesteckt wurde auf 1,8V an. Grund dafür war eine Verbindung con der Phase C mit dem Kollektor der Referenzspannungsaufbereitung ; Kontrolle, ob Motordrehmoment für alle 3 Achsen ausreicht; Ausfräsung Kühlkörper;

18.12.2000: Alle Layouts wurden nochmals ausgedruckt, jedoch wurde bei der Backplane die 90°- Drehung vergessen, wodurch dieses Layout verkleinert war. Der Fehler wurde mir erst bekannt, nachdem ich die Platine schon zugeschnitten hatte. Daher konnte ich nur die Interface-Platine Ätzen; Platine löten

22.12.2000: Da ich die anderen Platinen noch vor Weihnachten ätzen wollt, habe ich die restlichen Platinen noch am Freitag geätzt.

Weihnachtferien: Endfertigung der Bohrplotterplatinen; Verkabeln der Schrittmotoren und Gabellichtschranken; Planung der Rückplatte; Multiplexanzeige ansteuern; Fehlerkorrektur der einzelnen Platinen

8.1.2001: Fertigen und montieren des Kabelkanals für X-Achse; Fertigung der Rückplatte; Vorbereitung für die Programmierung am C++ Builder;

15.1.2001: Win98 auf Wechselfestplatte installiert; Anschließend C++ Builder installiert; Verkabelung der Steckdosen; Bearbeitungsplattenmontage; Einstellung der Schrittmotorkarte: Phasenstrom + Temperatur

22.1.2001: Z-Achse funktioniert nicht mehr korrekt: Fehlersuche Z-Achse

29.1.2001: Z-Achse läuft noch immer unruhig; Testprogramme zum Rechteck – fräsen; erste versuche mit KCAM

5.2.2001: Durch Unterbrechung der Inhibitleitungen konnte der Stromchopper nicht mehr funktionieren, daher L298 defekt; Reparatur von GLSY;

Semesterferien: Fertigung der Zusatzplatine für GSK- Codierung und Piepser; Frontplattengestaltung; Programmierung der Kalibrierung + Grundfunktionen;

26.2.2001: Überlegung + Programmierung der Rampe

5.3.2001: Perfektionierung des Rampenprogramms; Firmenpräsentation der Firma Dice

12.2.2001: Rampe auch für Y- und Z-Achse Programmieren; Firmenpräsentation der Firma D2 Vodafone

19.3.2001: Beschriftung der Rückseite; Kalibrierbügel aus Stahl fertigen; Veränderung des Kalibriervorgangs; Beginn mit der Fehlerkorrektur

26.3.2001: Dimensionierung eines Netzteils für die Schrittmotorbrücke (42V um schnellere Drehzahlen zu erreichen); Programmierung: einlesen der Bohrdatei, Sortierung der Durchmesser

2.4.2001: Kontrolle der Bohrdatei: Sortierung funktioniert!; Programmierung der Menüleiste

Osterferien: Reparatur der Z-Achse: Austausch des L298; Neuen 225VA-Trafo einbauen; Netzteil fertigen; Lüfter für Schrittmotorkarte einbauen; Messung der Schrittmotorkarte (Chopperaufnahmen); BP1 fotografieren; Fortsetzung Programmierung: Programmfehler von Fehlerkorrektur überarbeiten; Beginn mit der Dokumentation

23.4.2001: Besprechung wegen Netzteil : keine Spannungsregelung, Gefahr für Überspannung; Programmierung: Fehlersuche

30.4.2001: frei

7.5.2001: Abgabe der Dokumentation, Projektschluss

9. Kostenrechnung:

Interessant ist, dass das Ergebnis der Kostenschätzung meist niedriger liegt, als das der Kostenrechnung. Als ich Zu Projektbeginn eine Kostenschätzung machte, lag meine Schätzung bei max. 5000 ATS. Die folgende Kostenrechnung zeigt, dass meine Projektkosten auf 7500 ATS stiegen. Dabei ist zu beachten, dass ich einige aus alten Platinen ausgebaute Teile, für Y- und Z-Achse Druckerschlitzen und Schrittmotoren aus alten Druckern verwende. Der Kauf des teuren 19“- Racks blieb mir auch erspart, indem ich ein altes 19“-Rack von der Post verwendete. Im nachhinein würde ich sagen, dass für eine vernünftige Mechanik (keine Druckerschlitzen, sondern gekaufte Linearführungen) für das komplette Projekt mindestens 10000 ATS zu rechnen sind. Sehr vernünftig ist es, wenn man zur Finanzierung und Fertigung der Mechanik eine Firma findet. Z.B.: hat mir mein Nachbar, ein Chef bei der Firma Kitech gesagt, dass das Projekt in Verbindung mit seiner Firma als Sponsor kein Problem dargestellt hätte. Na ja, leider zu spät!

Kostenauflistung:

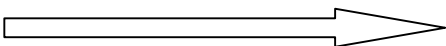
Mechanik:		
Linearführungen LF1 jeweils 250 ATS	500	
Wellenschlitzen für Linearführung LF1 jeweils 600 ATS	1200	
Grundplatte	1400	
Rechteckprofil, Gleitmuttern	580	
Versandkosten Isel	125	
Schrauben (alle Zylinderschrauben für Mechanik)	200	
Sonstige (Motoren + Druckerschlitzen von Schrotthändler)	180	
<i>Zwischensumme:</i>		4185
Elektronik:		
Schrittmotor ICs L297 u. L298 jeweils 3 Stück + 1Stück Ersatz	740	
Fast-Recovery-Dioden (teuer) 12 Stück	290	
Restlichen Bauteile	170	
Kabeln	110	
Platinen (Photolack)	210	
Kühlkörper	60	
Sonstiges (Buchsen, Stecker, Steckdosen, Relais....)	200	
<i>Zwischensumme:</i>		1780
Proxxon 100W Bohrmaschine mit Zubehör	1550	
<i>Zwischensumme:</i>		1550
Software:	0	0
Gesamt:	7550 ATS	7515 ATS

Vergleicht man die einzelnen Zwischensummen der 3 Gruppen, so ist sofort ersichtlich, dass die Mechanik den größten Teil ausmacht. Die Kompletten Materialkosten halten sich jedoch für einen Bohrplotter sehr niedrig. Nimmt man eine Technikerstunde mit 1500 ATS und multipliziert diese mit den Entwicklungsstunden (rund 300 Stunden) so erhält man einen Betrag von 450000 ATS. Dieses Beispiel zeigt, dass die Materialkosten im Vergleich zu den Entwicklungskosten sehr klein sind!

10. Schlusswort:

Ich kann mich noch erinnern, wie ich mich am Schulanfang für die Durchführung dieses Projektes eingesetzt habe. Während des Jahres habe ich mir manchmal, wenn ein Projekttag nicht so erfolgreich war, die Frage gestellt, warum ich dies getan habe. Wenn ich zurück denke, an die stundenlangen Fehlersuchen, die mich des öfteren zur Verzweiflung gebracht haben, frage ich mich noch manchmal, wieso ich mich für so ein arbeitsaufwendiges Projekt entschieden habe. Trotzdem bin ich im Nachhinein froh, dieses Projekt durchgeführt zu haben. Außer dass es für mich ein abwechslungsreiches und interessantes Thema war, habe ich dabei auch eine Menge Erfahrungen gemacht. Somit bin ich froh, dieses Projekt bis auf ein kleines Detail erfolgreich abgeschlossen zu haben. Für diesen Erfolg möchte ich mich speziell auch bei meinem Projekt- Betreuer, Herrn Dipl.-Ing. Höfer bedanken. Der Dank gilt auch den Werkstättenlehrern, Herrn Hackenbuchner, Herrn Vitale und Herrn Stuban, die mir bei der Fertigung der Mechanik hilfsbereit zur Verfügung standen. Nicht zu vergessen sind die Schüler der IAHE, die mir beim Drehen und Fräsen sowie anderen mechanischen Tätigkeiten Arbeit abgenommen haben. Zuletzt möchte ich mich noch bei meinem Freund Christian Scarlet bedanken, der für den Erfolg der Software zu einem großen Teil verantwortlich war.

Falls sie noch weitere Fragen zu meinem Bohrplotter haben, stehe ich gerne zur Verfügung. Meine Adresse ist unten angeführt. Den aktuellen Stand von meinem Bohrplotter können sie immer auf meiner Homepage www.bohrplotter.at.gs mitverfolgen. Die erste Änderung wird der Austausch der Y- und Z-Achse sein. Die verbesserte Version ist dann unter BP2.0 zu finden. Das Parallel- Komponent oder die aktuellste Version meiner Bohrplottersoftware BPSOft steht jederzeit auf meiner Downloadseite zum Downloaden zur Verfügung!

Wünsche, Anregungen, Beschwerden und Fragen zum Bohrplotter an: 	Mayr Bernhard Leitenbauerstr.9 4040 Linz mayr.bernhard@aon.at
--	---

Im Anhang 1 finden sie alle Schaltpläne, Layouts, Bestückungspläne, Messplots, Pinbelegungen, Strukturgramme und das Programmlisting!

Im Anhang 2 finden sie die wichtigsten Datenblätter und Application- Notes der verwendeten Bauteile!