

# Design och konstruktion av en automatisk kretskortsborrmaskin



---

**David Thomasson**  
**Johan Lennartsson**

Dept. of Industrial Electrical Engineering and Automation  
Lund University

# Abstract

This report is about constructing and building a controlled drilling machine for drilling circuit boards. The project has been carried out at Lund University, Faculty of Engineering at the department for Industrial Electronics and Automation. The report concerns mechanical, electrical and software aspects. But it also contains different problems you could face during a project like this, and also of course solutions to those problems.

There are other things discussed in the report as well, such as testing and calibration of the machine which has been taking a lot of time. CNC is another thing to read about in the report.

The result of the report and the project has been satisfactory. Several of the predetermined goals have been fulfilled. The drilling machine is going to be used as a part in the production process of circuit boards at the department and the report covers the work that has been carried out in the project in a good way. There is also a user manual for the machine.

Keywords: mechanic, electric, software, drilling machine, CNC

# Innehållsförteckning

<b>1. Inledning</b> .....	1
1.1 Bakgrund .....	1
1.2 Syfte .....	1
1.3 Målsättning .....	2
1.4 Metod .....	2
1.5 Avgränsning .....	3
1.6 Resultat .....	3
<b>2. Den mekaniska konstruktionen</b> .....	4
2.1 Maskinens uppbyggnad .....	4
2.2 De rörliga delarna .....	4
2.3 Tillverkning av maskinens mekaniska delar .....	7
2.4 Borrmaskinen .....	8
<b>3. Maskinens elektronik</b> .....	9
3.1 Motordrivkortet .....	10
3.2 Styr och reglerkortet .....	12
3.2.1 Programmering .....	13
3.2.2 Störningsreducering och avkoppling .....	14
3.2.3 AD-omvandlaren .....	14
3.2.4 EEPROM-minne .....	16
3.2.5 USB interface .....	18
3.3 Transformatorkortet .....	20
3.4 Kontrollpanelen .....	22
3.5 Lägesgivarna .....	23
3.6 Reläkort .....	23

<b>4. Mjukvara</b> .....	24
4.1 Programmeringsmiljöer .....	24
4.2 Styrprogrammet på datorn .....	25
4.3 Mikroprocessorns mjukvara .....	27
<b>5. Maskinen och styrsystemet</b> .....	34
5.1 CNC styrning .....	34
5.2 Nollpunkter .....	34
5.3 Kalibrering och testning av maskinen .....	36
<b>6. Problem och svårigheter</b> .....	37
6.1 Lägesgivarna .....	37
6.2 Problem vid detaljtillverkning .....	37
6.3 Glapp i maskinens rörliga delar .....	38
<b>7. Resultat</b> .....	39
<b>8. Specifikationer</b> .....	40
<b>9. Budget</b> .....	41
<b>10. Referenser</b> .....	42
<b>Bilaga A</b>	
<b>Bilaga B, kretsschema</b>	

# Bildföreteckning

2.1 Maskinens rörliga delar . . . . .	5
2.2 En av maskinens motorer som styr de rörliga delarna . . . . .	5
2.3 Exempel på hur kretskortet kan spännas fast inför borrning . . . . .	6
2.4 Hur X och Z-axel är monterade . . . . .	7
3.1 Styrelektroniken . . . . .	9
3.2 Kretsschema över ett drivblock baserad på L6203 . . . . .	10
3.3 Motordrivkortet till maskinen. . . . .	11
3.4 Tabell över motorns styrsignaler . . . . .	12
3.5 Digitala signaler . . . . .	13
3.6 MCP3208 (AD-omvandlare) . . . . .	15
3.7 EEPROM-minne M24256 . . . . .	17
3.8 Bild från hur enheten syns i enhetshanteraren i datorn . . . . .	19
3.9 MM232R USB Modul . . . . .	20
3.10 Transformatorkort. . . . .	21
3.11 Maskinens kontrollpanel . . . . .	22
3.12 Princip för lägesgivarna . . . . .	23
3.13 Bild över en av maskinens lägesgivare . . . . .	23
4.1 Styrprogrammet efter en fil laddats . . . . .	26
4.2 Förenklad schematisk översikt över programflödet i mikroproc. . . . .	28
4.3 P-reglering . . . . .	31
4.4 Regulator parametrar från mikroprocessorns källkod . . . . .	32
5.1 Bild på kretskort monterat i maskinen . . . . .	35

# 1. Inledning

Examensarbetet har utförts under hösten/vintern 2006 på institutionen IEA i maskinhuset på LTH i Lund. Handledare för arbetet har varit Gunnar Lindstedt. Även Getachew Darge har varit till stor hjälp främst genom att tillhandahålla verktyg och annan utrustning.

## 1.1 Bakgrund

På institutionen IEA på LTH finns det möjlighet att tillverka kretskort som sedan skall används för projekt eller laborationer. Kretskortets design bestäms av den som arbetar med projektet och sedan etsas kretskortet efter det. Efter etsningen återstår det att borra hål för komponenterna som skall monteras. Detta arbete utförs i dagsläget manuellt med en miniatyrborr av höghastighetstyp. Arbetet är väldigt tidskrävande då det ofta är hundratals hål som ska borraras. Det finns därför önskemål från institutionen att tillverka en styrd bormaskin som utför borrarbetet automatiskt. Detta skulle också kunna medföra andra fördelar som högre precision och färre avbrutna borrar.

## 1.2 Syfte

Syftet med examensarbetet är att tillverka en CNC-styrd bormaskin som underlättar arbetet med att borra hål i kretskort före komponentmontering. Filformatet som kommer att användas för importering från kretskortslayout-programmet är industristandardiserade och kallas även CAD/CAM (Excellon). Det kommer också att bli en intressant utmaning eftersom det är ett arbete som kräver kunskap inom flera olika områden som mekanik, elektronik och programmering.

## 1.3 Målsättning

Det finns flera olika mål att uppfylla. Några av de viktigaste är:

- Att lösa uppgiften på ett tillfredställande sätt, dvs. att ta fram en fungerande maskin för kretskortsborrning
- Att i stor utsträckning återanvända befintligt material som finns på institutionen
- Att tillverka en maskin med relativt hög precision till ett lågt pris
- Att fullfölja arbetet inom den utsatta tidsramen
- Att göra maskinen användarvänlig

## 1.4 Metod

Projektet består av några olika delar:

1. Att bygga upp bormaskinen mekaniskt
2. Att bygga elektroniken som behövs
3. Att skriva den programvara som ska styra maskinen
4. Att testa och kalibrera maskinen

Mest tid kommer att läggas på elektroniken och mjukvaran medan det förmodligen kommer att gå snabbare att bygga det mekaniska. Tanken är att snabbt komma igång med konstruktionen av elektroniken och den mekaniska designen på bormaskinen. Anledningen till detta är att det är viktigt att kunna beställa hem material och komponenter i ett tidigt skede så att inte arbetet blir försenat pga. det.

## **1.5 Avgränsning**

De krav som IEA har ställt på projektet är att de vill ha en styrd och fungerande bormaskin som utför arbetet med borrhningen på ett tillfredställande sätt. Här är pålitlighet och precision mycket viktigare än att det går snabbt eftersom det inte rör sig om någon tidspressad massproduktion. Det kommer att finnas begränsningar för hur stora kretskort som kommer att kunna borrar samt antal hål som kan borrar men detta kommer ändå att vara tillräckligt för institutionens krav.

## **1.6 Resultat**

Resultatet av projektet uppfyller flera utav de uppsatta målen. De problem som har uppstått och de krav som har ställts har lösts på ett bra sätt och maskinen fyller sin funktion. Projektet har framskridit i snabb takt och trots en låg budget har en fullt fungerande kretskortsbormaskin konstruerats och byggts. Överlag är projektmedlemmarna väldigt nöjda med resultatet. Även institution är väldigt nöjda med resultatet och maskinen kommer att underlätta arbetet med kretskortstillverkningen betydligt.



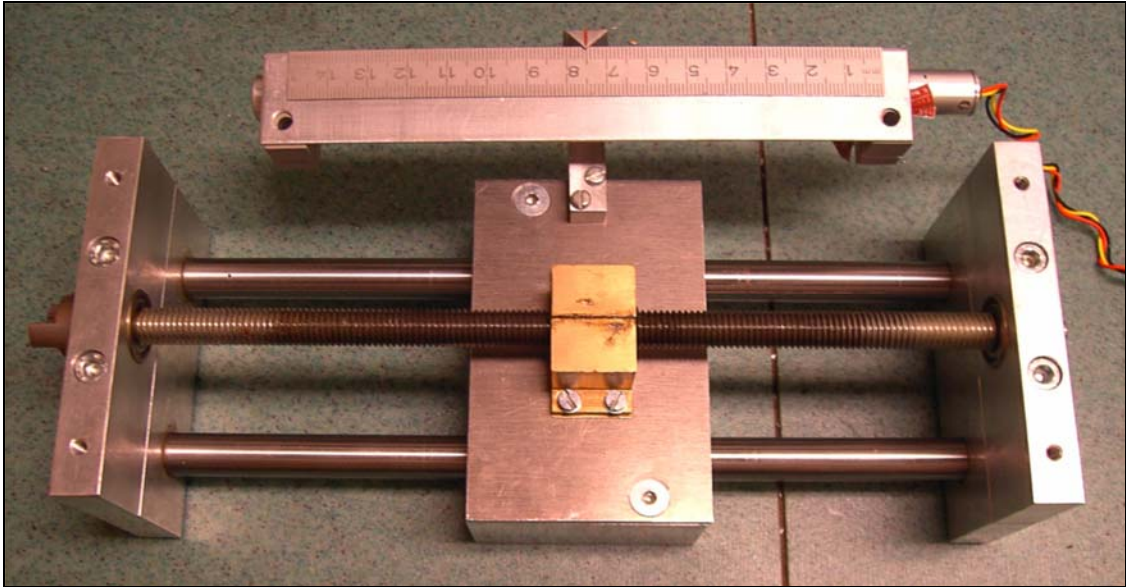
## **2. Den mekaniska konstruktionen**

### **2.1 Maskinens uppbyggnad**

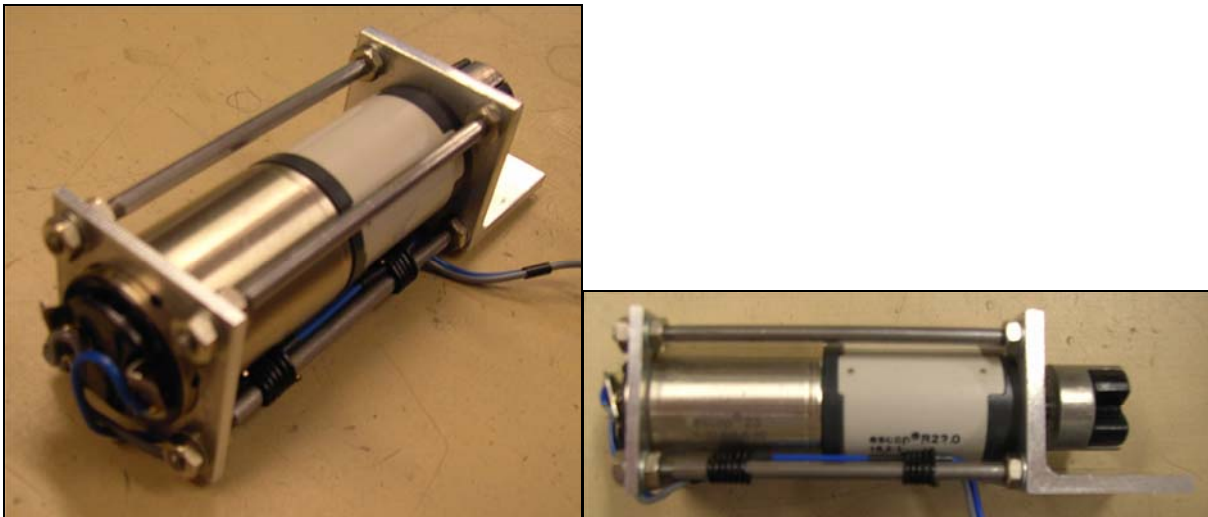
Maskinens mekaniska konstruktion påminner om de kommersiella CNC-styrda maskiner som finns att köpa på marknaden. Maskinen är till stor del uppbyggd av aluminiumprofiler. Här har stabilitet prioriterats framför att maskinen ska vara liten och smidig. Det är av största vikt att allt sitter på plats och att maskinens fasta detaljer inte flyttar sig under borrarprocessen eller med tiden. Aluminiumprofilerna har fixerats med skruvförband och låsmuttrar på de flesta ställen och eftersom aluminiumet lätt ger vika har extra metallbitar tillverkats för att kunna stå emot skruvarnas kraft på vissa platser. För extra stabilitet har två stråvor, också i aluminium, monterats på diagonalen mellan de balkar som går vertikalt och horisontellt. Detta förbättrade stabiliteten avsevärt.

### **2.2 De rörliga delarna**

Maskinens rörliga delar (X- Y- Z-led) kommer att förflyttas med hjälp av DC-linjärmotorer som tidigare har använts av institutionen i laborationer. Konstruktionen kring motorerna består av en gängstång och en invändigt gängad mässingsbit som är fäst i ett block. När motorn körs så snurrar skruven och blocket kan löpa längs två stänger som går genom blocket parallellt med skruven. Stängerna ger bra stabilitet längs skruven. I blockets hål sitter invändiga lager som minskar friktionen mellan stängerna och blocket. I konstruktionen finns även en lägesgivare som är monterad parallellt med skruven. På metallblocket sitter ett stift fastskruvat som löper längs lägesgivaren från vilken positionen kan erhållas.



*Figur 2.1 - Maskinens rörliga delar*



*Figur 2.2 - En av maskinens motorer som styr de rörliga delarna*

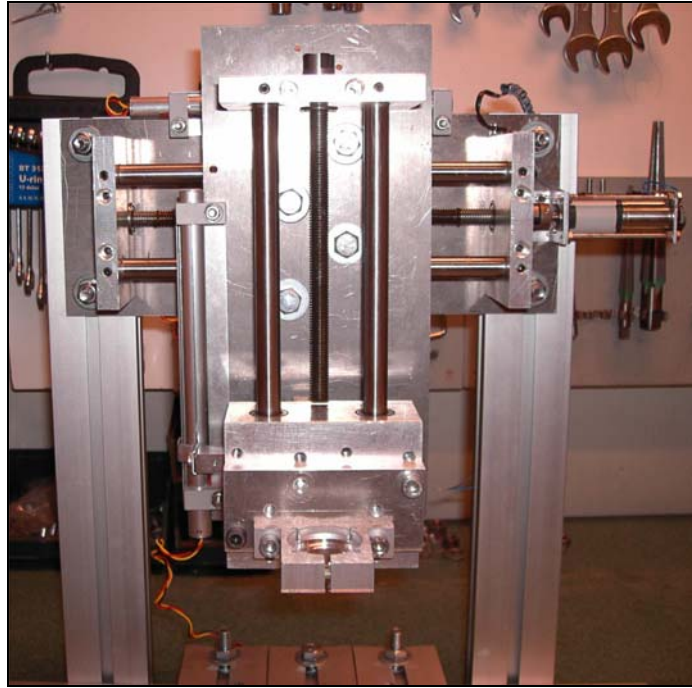
På bottenplattan sitter bordet som ska löpa i Y-led fastmonterat. Ovanpå det finns i sin tur en platta där kretskortet kan spännas fast inför bearbetning. Fastspänningsmekanismen består av en plastbit med två skruvar som man kan spänna fast i bordet med två vingmuttrar. Skruvarna löper i två urfrästa spår så att man kan fixera kretskort av olika storlek. Mellan kretskortet och bordet finns en skiva som borret kommer att borra ner i efter att ha gjort hål i kretskortet.

Detta för att förhindra att borret kommer i kontakt med den underliggande metall som bordet är gjort av och därmed skydda borret och förhindra att det bryts av.



*Figur 2.3 – Exempel på hur kretskortet kan spännas fast inför borring*

Maskinens konstruktion som ska gå i X-led är monterad i de vertikala aluminiumprofiler som sitter skruvade i bottenstativet. På blocket som går i X-led sitter sedan den motor monterad som kommer att köra borren i Z-led.



*Figur 2.4 - Hur X- och Z- axel är monterade*

## **2.3 Tillverkning av maskinens mekaniska delar**

Maskinens mekaniska delar har tillverkats i en verkstad som har stått till förfogande på skolan. De rörliga delarna som kan ses på figur 2.1 har demonterats från en redan färdig konstruktion som har använts i en laboration. Dessa delar är till för att skapa en förflyttning i en endimensionell linjär rörelse. Förutom detta har allt annat konstruerats och tillverkats från grunden. I princip all skärande bearbetning har skett med en pelarborrmaskin som också kan användas som fräsmaskin. Med tanke på att själva maskinen inte kommer att produceras mer än i en upplaga så har ett flertal speciallösningar används efter det material som har stått till förfogande. Ett av målen har ju varit att återanvända gamla detaljer. Genom att tänka ut olika lösningar som skulle kunna användas och sedan testa den som har verkat bäst så har den nuvarande konstruktionen uppstått. Det har alltså inte funnits någon färdig ritning för hur själva konstruktionen skulle se ut innan byggandet startade.

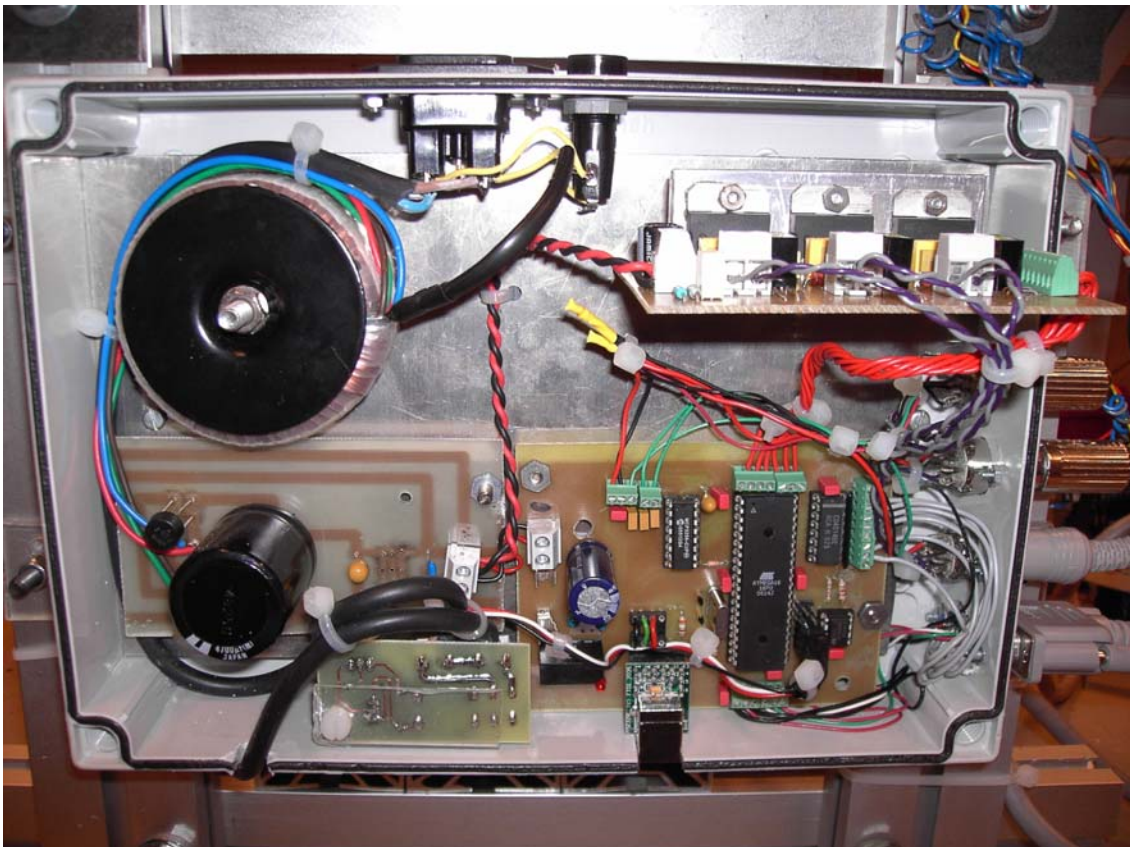
## 2.4 Borrmaskinen

Borrmaskinen är ett höghastighets multiverktyg med en chuck för olika borrhålsdimensioner. Fastspänning av borrar sker med hjälp av expanderdorn. Borrmaskinen sitter fast i en infästning som är inköpt ihop med borrmaskinen. Infästningen är i sin tur fastskruvad på det block som går i Z-led med en distans emellan. Det finns möjlighet att ställa in olika spindelhastigheter på borrmaskinen. Dessa hastigheter är mellan 5000 – 20 000 varv/min. Borrmaskinen drivs direkt från elnätet.



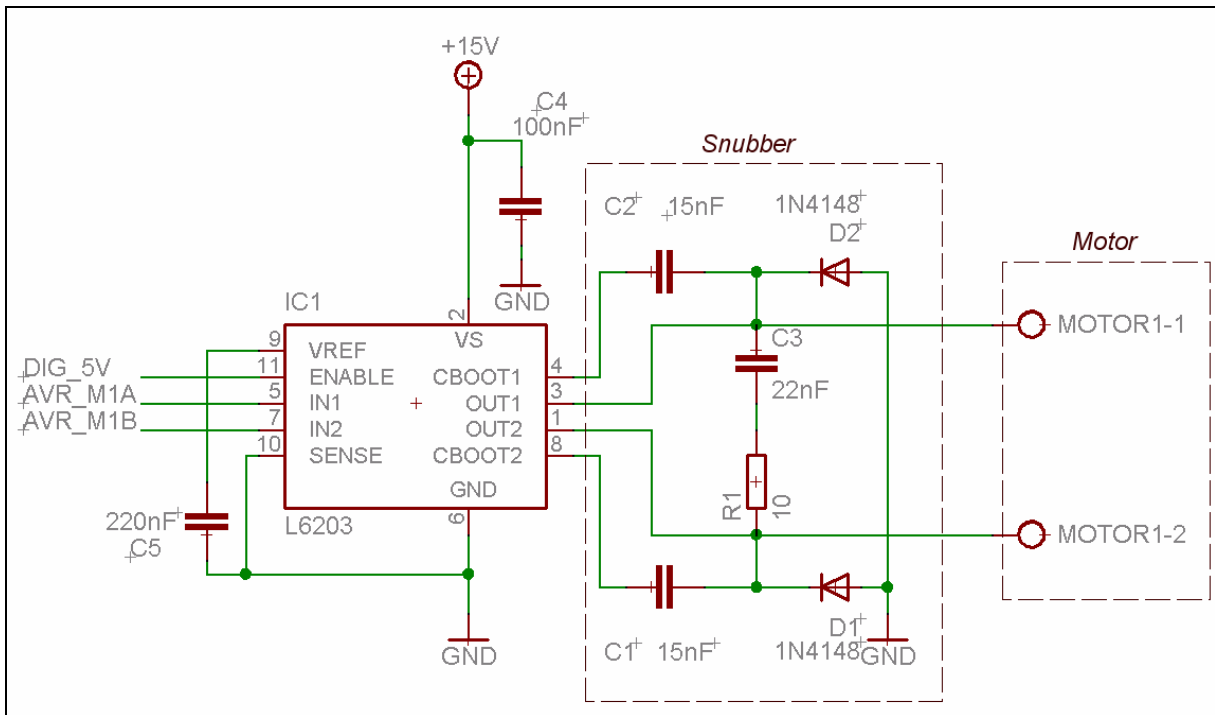
### 3. Maskinens elektronik

Elektroniken till maskinen består till stor del av två kretskort. Ett för kraftelektroniken till motorerna och ett för styrning och reglering av maskinen. Det finns även en del elektronik kring maskinens transformator. Kretskorten är egenhändigt designade och tillverkade av oss (David och Johan) på institutionen. Kretskorten har monterats i en elektronikbox som sitter monterad på maskinens baksida för att inte ta upp onödig plats eller vara i vägen. Det är dock väldigt lätt att skruva av locket på boxen och komma åt elektroniken om något skulle gå sönder. Återigen har ett av målen varit att återanvända så mycket befintligt material som möjligt. All design av kretslayout är gjord i ett program som heter EAGLE (version 4.16) [1].



*Figur 3.1 - All styrelektronik sitter i en apparatlåda monterad på baksidan av maskinen*

### 3.1 Motordrivkortet

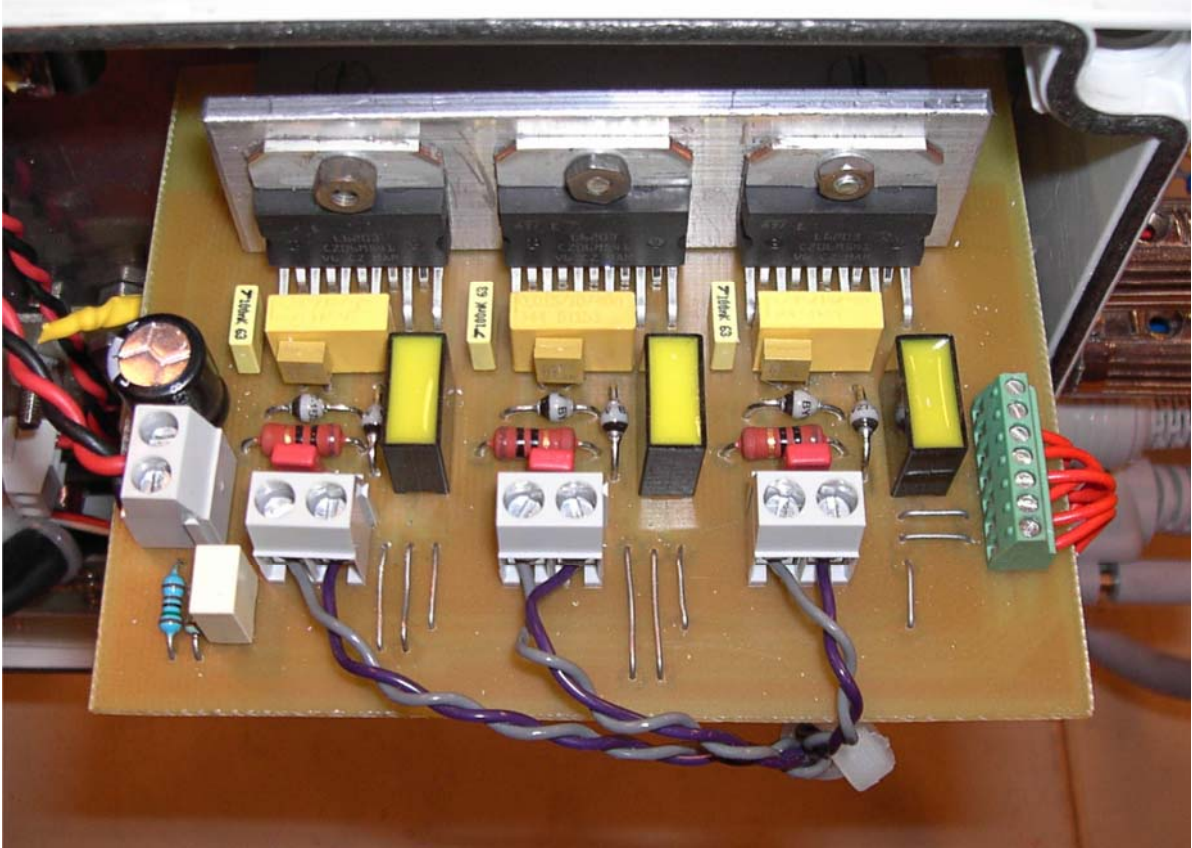


Figur 3.2 - Kretsschema över ett drivblock baserat på L6203

Motordrivkortet är uppbyggt kring drivkretsen L6203 (H-brygga, tillverkad av ST Microelectronics [2]). Detta är en komplett motordrivkrets med kontroll-logik för styrning direkt från mikroprocessor (TTL-nivåer) vilket drastiskt förenklar designen och minskar antalet kringkomponenter. Kretsen är kapabel att driva 4A med en maximal spänning på 36V vilket är med god marginal tillräckligt för motorerna som används och den kan även driva motorerna åt båda hållen. Ett alternativ hade varit att bygga drivkretsen kring diskreta komponenter, men detta ger ingen större fördel, varken kostnads- eller utrymmesmässigt eftersom strömnivåerna är låga.

Drivkortet är baserat på referensdesignen för L6203 enligt databladet för kretsen. Det består av tre identiska block (se figur 3.2 för kretslayout för ett block) där varje block driver en motor, en större avkopplingskondensator för hela drivkortet samt anslutningsplintar för motordrivning, spänningsmatning samt styrning. Ett transient-avstörningsfilter (även kallad snubber) används för att undertrycka spänningsspikar som uppstår då motorerna switchas av och på.

Komponentvärdena på kondensatorer och motstånd är inte beräknade utan även dessa är tagna ur referensdesignen. Ett oscilloskop har använts för att verifiera att inga synliga transienter existerar.



Figur 3.3 - Motordrivkortet till maskinen

Kondensatorerna C4 och C5 används för lokal avkoppling till kretsen och den större avkopplingskondensatorn till hela drivkortet är på 470  $\mu\text{F}$ .

*Enable* (pin 11) på H-bryggan används för att aktivera hela kretsen och är alltid satt till hög. Eftersom endast 15V finns tillgängligt på drivkortet skapas en hög signal (5V) genom spänningsdelning över två motstånd.

Motorerna som skall drivas är de som beskrivs i föregående kapitel.



Pinnarna *In1* (*pin 5*) och *In2* (*pin 7*) används för att styra motorn från mikroprocessorn och fungerar som digitala ingångar. Då *In1* sätts till hög så ges matningsspänning (15V) ut på *Out1* (*pin3*) och då *In1* sätts till låg så ges 0V ut (samma gäller för *In2* respektive *Out2*). Genom att variera vilka nivåer ingångarna sätts till så fås följande fall enligt figur 3.4.

<i>Out1</i>	<i>Out2</i>	<b>Spänning över motor</b>	<b>Resultat</b>
0V	0V	0V	Stopp
15V	0V	15V	Snurrar medurs
0V	15V	-15V	Snurrar moturs
15V	15V	0V	Stopp

*Figur 3.4 – Motorns styrsignaler*

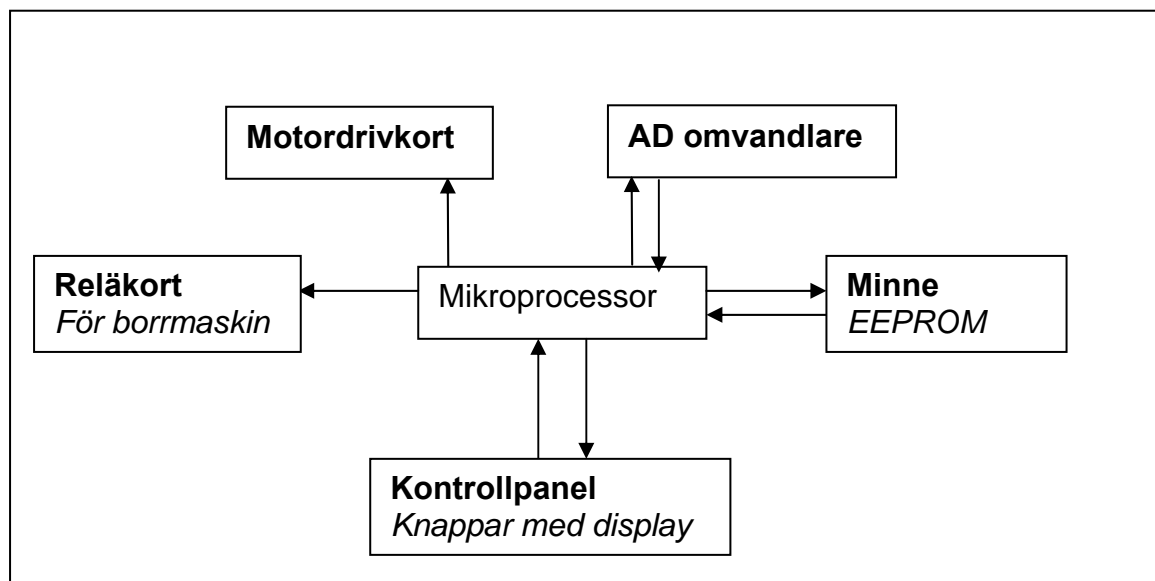
Genom att pulsbreddsmodulera signalerna till H-bryggan kan motorns hastighet varieras. En aluminiumprofil är monterad längs baksidan på kretsarna och används för att fästa fast kortet i elektronikboxen. Profilen fungerar också som kylfläns och fördelar värmen jämnt mellan kretsarna. Dock är effektbehovet så litet att dessa inte blir varma och därför har vi inte tagit med några beräkningar och dimensioneringar för kylningen.

## 3.2 Styr- och reglerkortet

Styr- och reglerkortet är den centrala delen i elektronikkonstruktionen. På detta sitter mikroprocessor, AD-omvandlare till positionsläsning, USB-interface som används för att koppla maskinen till en dator och alla anslutningar till display, kontrollpanel osv.

All styr och reglering sker med en mikroprocessor från Atmel, modell AVR ATmega16 [3]. Denna modell har en lagom mängd IO-pinnar och de interfacefunktioner implementerade i hårdvara till den kringutrustning som används. Detta förenklar programmeringen och designen samt gör den effektivare.

En kristall används för att få en exakt klockfrekvens och denna är satt till 14,7456 MHz, vilket är mer än tillräckligt för att hinna med den reglering och styrning som krävs. Denna frekvens är även väl lämpad till att använda till den seriella kommunikation till datorn då hela värden på klockfrekvensen t.ex. 8 eller 16 MHz kan ge ökad risk för felöverföring.



Figur 3.5 - Digitala signaler

### 3.2.1 Programmering

Processorn drivs precis som all annan digital elektronik i maskinen med 5V. Den programmeras via ett interface som heter JTAG och anslutningen för denna sitter bredvid som en 10 pinnars flatstift-kontakt. Den kan programmeras direkt monterad i kretskortet och behöver alltså inte plockas bort för att kunna testa ny kod.

## 3.2.2 Störningsreducering och avkoppling

Avkopplingskondensatorer för matningsspänningen har använts för att minska risken för störningsrelaterade problem. Alla digitala kretsar (AD, mikroprocessor, EEPROM, display osv.) har en 100nF polyesterkondensator monterad tätt intill. AD-omvandlaren har dessutom en 1uF tantalkondensator enligt databladets rekommendation. Större kondensatorer sitter monterade i anslutning med strömförsörjningen och spänningsstabilisatorerna och fungerar som grövre glättningskondensatorer.

Spänningen har kontrollerats med oscilloskop och är stabil, inga spikar existerar. Den positiva spänningen har ritats i layouten som en stjärna, dvs. att de utgår från en punkt nära matningsspänningens källa. Detta för att förhindra att störningar från exempelvis analoga komponenter letar sig in till digitala via matningen och vice versa och även för att delar där högre strömmar går inte skall skapa spänningsfall som stör andra känsliga komponenter t.ex. AD-omvandlaren. Jorden har lagts som ett heltäckande jordplan.

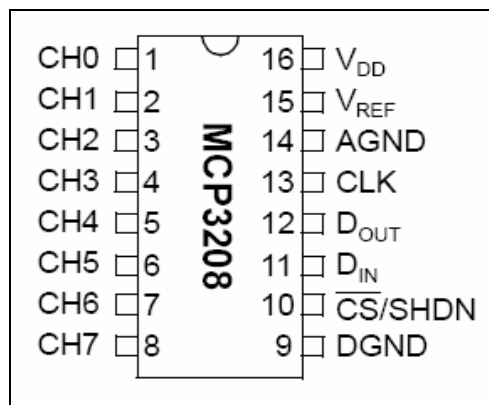
## 3.2.3 AD-omvandlaren

Lägesgivarna fungerar elektriskt sett som en potentiometer. Då en spänning läggs över så ges positionen som en elektrisk spänning som varierar mellan 0V och spänningen som används. För att kunna använda denna analoga spänning i mikroprocessorn behövs en AD-omvandlare som omvandlar signalen till ett digitalt värde. Atmega16 har inbyggda 10 bitars AD-omvandlare, men med denna precisionen fås en mätupplösning på:

$$150 \text{ (mm)} / 2^{10} = 147 \text{ } \mu\text{m}$$

Där 150 mm är längden på lägesgivarna. Eftersom en precision på 1 (eller upp till 2) bitar är rimligt att anta att regleringen klarar av så är detta för stora diskreta steg att ha då större noggrannhet än detta önskas. En diskret AD-omvandlare med högre upplösning används därför istället som ersättare till den interna. MCP3208 från Microchip [4] valdes eftersom:

- Den är billig
- Finns som hålmonterad
- Har kort omvandlingstid
- Jobbar med 5V matningsspänning
- Flera kanaler vilket gör att bara en krets krävs
- Använder SPI-buss vilket Atmega16 stödjer i hårdvara



Figur 3.6 - MCP3208

MCP3208 har 12 bitars upplösning, vilket med samma resonemang som ovan ger en mätupplösning på:

$$150 \text{ (mm)} / 2^{12} = 37 \text{ } \mu\text{m}$$

Pinnen  $V_{ref}$  sätter den analoga spänningen som motsvarar det maximala digitala värdet (som är  $2^{12} - 1 = 4095$ ).  $AGND$  är den analoga jorden och sätter den minsta spänningen, som motsvarar det digitala värdet 0. Pinnarna 10 till 13 används för kommunikationsinterfacet (SPI) som mikroprocessorn kommunicerar via.

En konvertering går till på följande sätt: mikroprocessorn skickar ett kommando för vilken kanal som skall avläsas. AD-omvandlaren gör en konvertering från den valda kanalen och svarar till mikroprocessorn med värdet på omvandlingen. Eftersom längden på värdet är 12 bitar och mikroprocessorn jobbar med 8 bitar, så måste två inläsningar göras och sedan efterbehandlas.

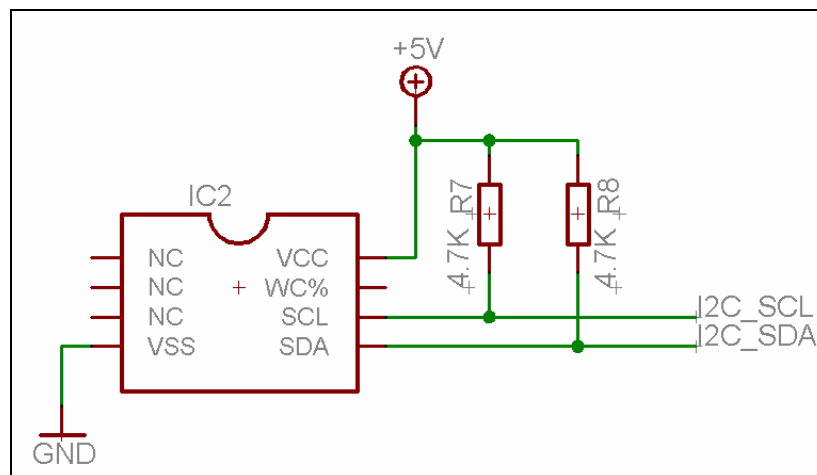
Då lägesgivaren förflyttar sig ger detta en brusig spänning och om den läses in till AD-omvandlaren utan någon filtrering så varierar värdena på varje sampel ganska mycket. I mjukvaran tas 16 sampel och medelvärdesbildas för att minska störningar, men detta är inte tillräckligt för att få ett stabilt digitalt värde. För att eliminera bruset i princip helt så har en keramisk kondensator på 100nF på varje AD-kanal monterats, kopplad till jord. Tillsammans med resistansen från lägesgivaren så skapas ett lågpassfilter som filtrerar bort det högfrekventa bruset. Storleken på kondensatorn är experimentellt bestämd, dess värde är inte kritiskt. Dock gör ett för litet värde att brytfrekvensen ligger för högt och filtret får ingen effekt och ett för stort värde gör att signalen kommer bli fördröjd likt uppladdningen av en kondensator i en RC-krets.

MCP3208 har en maximal samplingshastighet på 100 ksps vilket gör att ett antal sampel kan tas och medelvärdesbildas vid varje mätning samtidigt som en snabb reglering erhålles.

### **3.2.4 EEPROM minne**

För att kunna lagra alla positioner till borrhålen så krävs ett ganska stort minne. Det interna minnet i mikroprocessorn räcker endast till för ett fåtal positioner och målet är att alla positioner skall kunna överföras direkt från datorn och sedan att maskinen skall kunna jobba utan att någon anslutning till datorn krävs. Varje position är lagrad som 16 bitar för X och 16 bitar för Y. Med ett minne på 32Kx8 bitar så ger detta ungefär 8000 positioner som kan lagras. Detta är tillräckligt bra då ett normalt kort med dimensionerna maskinen klarar av att borra innehåller färre hål än detta.

Ett externt EEPROM minne som uppfyller kraven är M24256 från ST Microelectronics [5]. Detta minne har en storlek på 32Kx8 bitar och ansluts via seriell I2C-buss som ATmega16 stödjer i hårdvara. Kretsen finns tillgänglig i en 8-pinnars hålmonterad kapsel vilket gör den lätt att ersätta och priset är lågt.



Figur 3.7 – EEPROM minne M24256 från ST Microelectronics

I2C-interfacet är lätt att ansluta till mikroprocessorn, endast två IO-pinnar behöver användas; seriell klocka (*I2C\_SCL*) och dataöverföring (*I2C\_SDA*). Dessa två anslutningar kopplas även via pull-up motstånd till 5V. Storleken på pull-up motstånden bestämmer tillsammans med ledningskapacitansen den maximala överföringshastigheten som man kan se i databladet. Exakta värden på ledningskapacitanser är svårt att ta reda på, men detta är inte kritiskt då överföringshastigheten inte är flaskhalsen för minnet utan tiden för en skrivcykel samt ledningsbanorna i designen är korta och bör därför ha låg kapacitans. Värdet på pull-up motstånden har satts till 4,7kΩ.

Nackdelarna med ett EEPROM minne är att de har en begränsad livslängd när det gäller antalet skrivcyklar. Databladet garanterar minst 100 000 skrivcyklar vilket i vår applikation är mer än tillräckligt då maskinen inte skall användas för serieproduktion. En annan nackdel är att skrivhastigheten är relativt långsam, 10  $\mu$ s för varje byte. I praktiken ställer inte detta till något problem då det är acceptabelt om användaren tvingas vänta upp till 20 sekunder för att förbereda maskinen för borrning, eftersom maskinens hastighet inte har högsta prioritet.

En annan minnestyp hade gått att använda, t.ex. SRAM då dessa är snabbare och inte har någon skrivbegränsning. Interfacemöjligheterna är dock sämre på dessa minnen därför valdes ett EEPROM minne istället.

### **3.2.5 USB interface**

För att kommunicera med datorn via mikroprocessorn behövs ett interface. Det absolut enklaste sättet att lösa detta på är att använda sig av de seriella COM-portarna som finns, detta kräver endast en spänningsnivå-omvandlare som finns integrerat i färdiga kretsar. Stöd för detta interface finns även implementerat i hårdvara i mikroprocessorn och är väldigt enkelt att använda, både i programmet på PC:n och på mikroprocessorn. Nackdelen med detta interface är att det är gammalt, kontakterna är stora och stöd kan eventuellt försvinna i framtiden speciellt på bärbara datorer.

Ett nyare och bättre interface är USB gränssnittet. USB har även möjlighet för betydligt större dataöverföringshastighet, något som dock inte utnyttjas av denna maskin. Nackdelen med USB gränssnittet är att det är betydligt svårare att implementera en fungerande lösning i mjukvara.

Genom att använda en krets från FTDI [6] som heter *FT232R (USB to Serial converter)* så försvinner problemet med USB programmeringen. FT232R hanterar USB-protokollet helt själv och sköter all kommunikation med datorn. Denna krets simulerar en virtuell COM-port på datorn, det vill säga att på PC:n så skrivs programmet som om enheten varit ansluten till en vanlig seriell COM-port och mot mikroprocessorn används ett vanligt UART seriellt interface. Drivrutiner för enheten är gratis och finns till flera olika operativsystem.

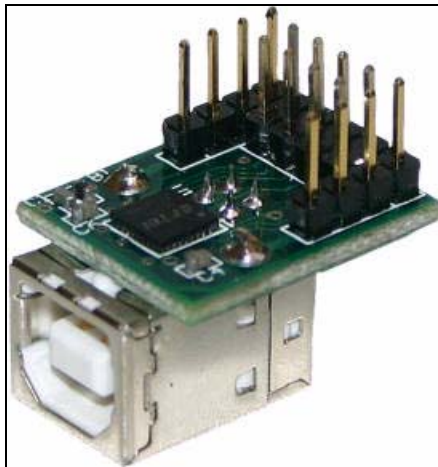


*Figur 3.8 – Bild från hur enheten syns i enhetshanteraren på datorn*

När användaren för första gången ansluter hårdvaran som använder USB-kretsen så dyker denna upp i Windows som vilken ny hårdvara som helst, och efter att drivrutiner har installerats så behöver detta inte göras fler gånger. I enhetshanteraren finns enheten sedan som en vanlig COM-port, se figur 3.8.

Nackdelen är ju naturligtvis att det blir en något ökad kostnad för denna krets, men till ett litet projekt likt detta som inte skall serieproduceras så är det till en stor fördel eftersom det sparar väldigt mycket tid och arbete.





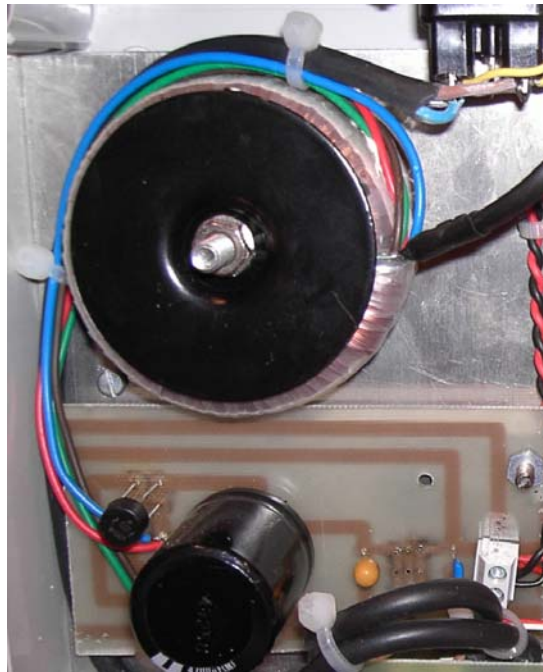
*Figur 3.9 - MM232R USB modul*

En färdig modul med krets och USB-kontakt monterad finns färdig från tillverkaren (modellen heter MM232R [7]) och förenklar arbetet ännu mer då lödning av den lilla ytmonterade kretsen inte behöver göras. Det är denna modulen som används i maskinen.

### **3.3 Transformatorkortet**

Maskinen behöver två olika likspänningsnivåer; 5V för den digitala elektroniken och 15V för motordrivningen. Ett gammalt kort fanns tillgängligt på institutionen med  $\pm 18V$  DC på utgången, komplett med transformator, likriktare och spänningsstabilisator. Detta kort har använts som nätaggregat till maskinen, fast spänningsstabilisatorn har ersatts med en som ger ut 15V samt den negativa delen har plockats bort för att spara plats.

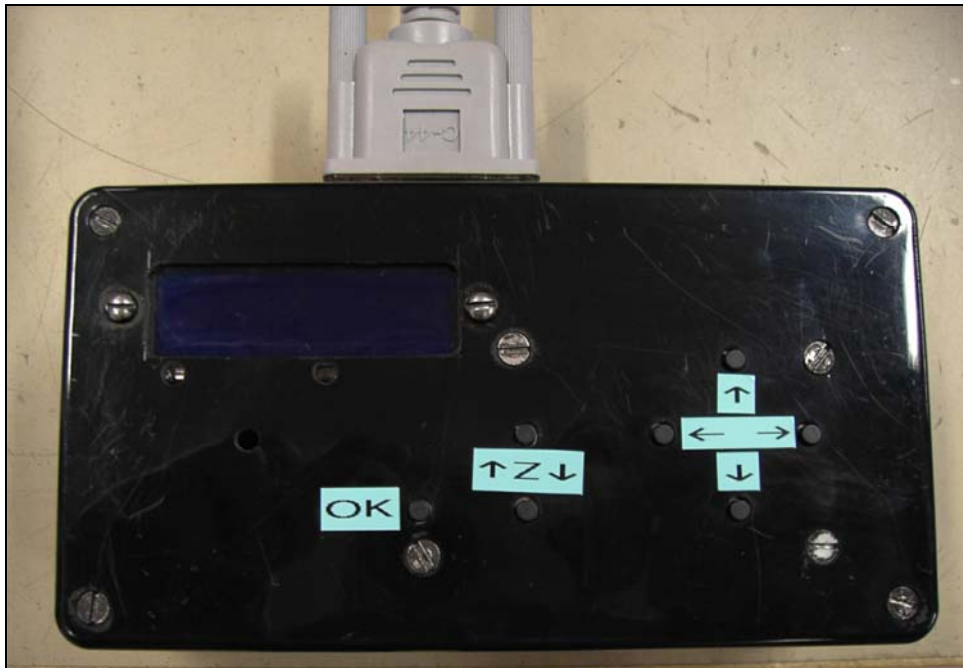
Transformatorn som sitter på kortet är av ringkärnetyp vilket reducerar brus och andra störningar. Denna har en effekttålighet på 30VA vilket räcker väl. Nackdelen är att den ger ut 23V likriktat vilket gör att en relativt stor spänningsskillnad bränns bort som värmeeffekt i spänningsstabilisatorn. En stabilisator som ger ut 5V är monterad på styrkortet och kopplad till 15V-matningen för att driva den digitala elektroniken. Stabilisatorerna är väl kyllda mot en metallplatta i botten som alla kretskort är skruvade mot. Detta är en kompromiss då målet har varit att återanvända så mycket som möjligt för att hålla ner kostnaden, det hade varit bättre att använda komponenter mer anpassade för ändamålet för att få så lite förluster som möjligt.



*Figur 3.10 – Transformatorkort*

### 3.4 Kontrollpanelen

För att kunna styra maskinen manuellt finns en kontrollpanel med tryckknappar och en display för kommunikation mellan maskinen och användaren. Det finns sju st knappar på panelen. Sex av dem är för att styra maskinen i X- Y- och Z-led i båda riktningarna (plus och minus). Den sjunde knappen är en OK-knapp för att bekräfta olika förfrågningar från maskinen. Kontrollpanelen kommer mest att användas när kalibrering ska ske eller när maskinen ska ställas in inför ett nytt jobb. Men kan användas för att göra andra förflyttningar också.

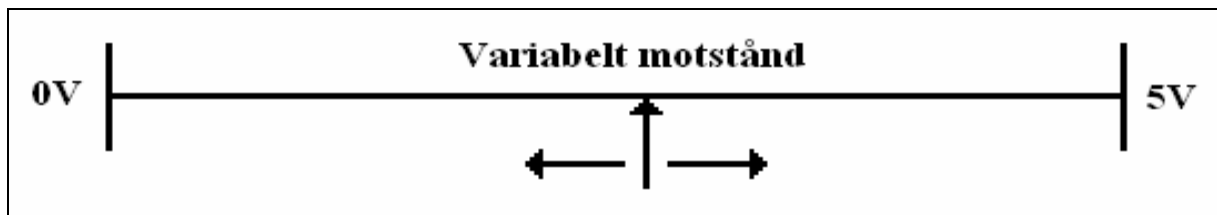


*Figur 3.11 - Maskinens kontrollpanel*

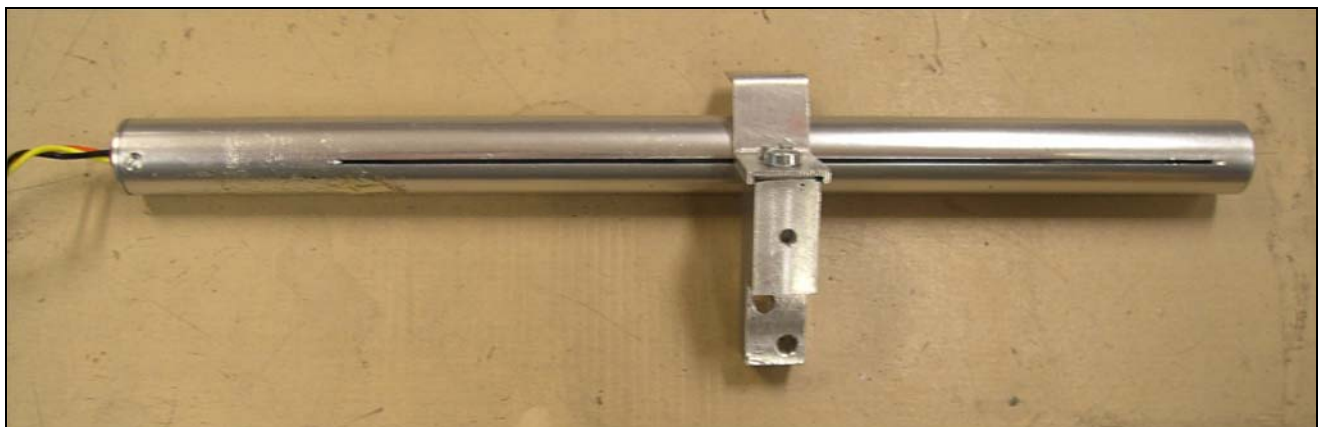
Displayen som används på kontrollpanelen är en bakgrundsbelyst LDC-display från Electronic Assembly [12]. Den använder ett seriellt kommunikationsinterface och är enkel att använda. Initieringsrutiner och instruktioner finns i databladet.

### 3.5 Lägesgivarna

De lägesgivare som sitter på varje motorkonstruktion är till för att kunna återge vilken position de olika axlarna befinner sig i. Lägesgivaren är egentligen en linjär potentiometer som det läggs en spänning över som sedan kan avläsas och ge positionen.



*Figur 3.12 - Princip för en av de lägesgivare som sitter monterad på varje motorkonstruktion*



*Figur 3.13 - Bild över en av maskinens lägesgivare*

### 3.6 Reläkort

Då t.ex. alla hål borrats eller då ett verktygsbyte skall ske så stängs bormaskinen av. För att kunna göra detta så har ett kort med ett relä lagts till som bryter elen till bormaskinen. Eftersom mikroprocessorns utgångar är för svaga för att kunna driva reläet så har en NPN transistor satts som en emitterföljare. Detta gör att strömmen som går genom mikroprocessorn blir långt under vad som är tillåtet.

## 4. Mjukvaran

Mjukvaran kan delas upp i två delar; styrprogrammet på PC:n och mjukvaran i mikroprocessorn. Huvuduppgiften för programvaran på datorn är att importera filer från PCB-layoutprogrammet och sedan överföra koordinater för alla borrhål och vilken verktygsstorlek som skall användas till de olika hålen. Mikroprocessorn styr hela maskinens elektronik vilket innefattar bland annat motorreglering, beräkningar för hålpositionering och tidsberäkning.

### 4.1 Programmeringsmiljöer

Styrprogrammet är programmerat för att användas i Windows och skrivs i C# [8] (Microsoft Visual Studio .Net 2005 Service pack 1). Detta ger programmet ett användarvänligt grafiskt gränssnitt. För att använda programmet måste .Net-plattformen vara installerad på datorn (om den inte redan är det). Denna uppdateringen är dock gratis och kan laddas ner från bland annat Microsofts hemsida [9].

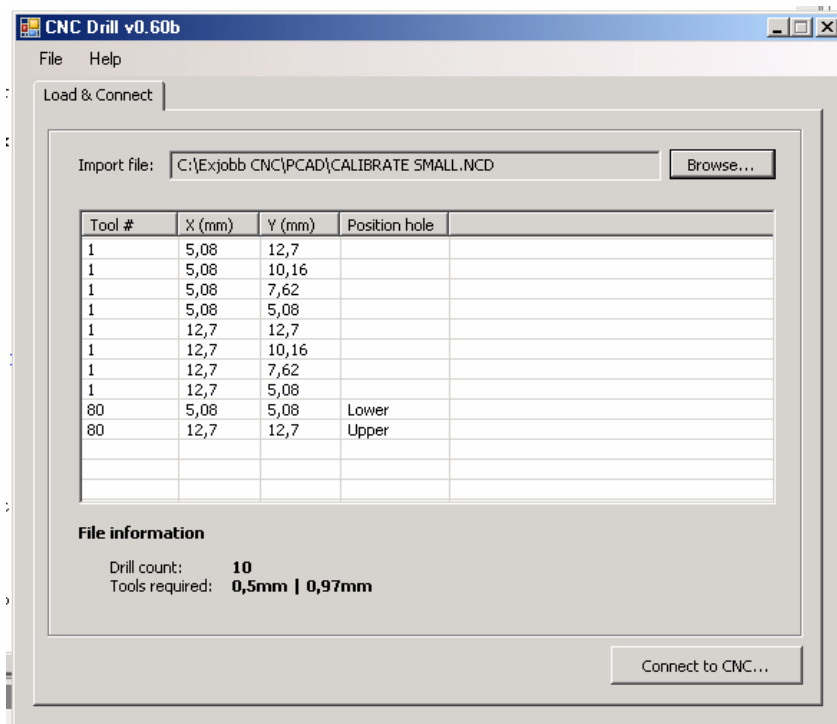
Mikroprocessorn är programmerad i AVR Studio [10] (version 4.12, service pack 4) från Atmel och det är C som används som programmeringsspråk. AVR Studio är ett grafiskt IDE (Integrated Development Environment) och underlättar programutvecklingen genom olika funktioner som realtidsdebugging. Ett tilläggspaket som heter WinAVR [11] (Build 2006-04-21) används för att ge extra stöd för C-programmeringen.

All kod har rikligt med kommentarer och därför utlämnas programmeringsdetaljer ur rapporten.

## 4.2 Styrprogrammet på datorn

Styrprogrammet importerar filer skapade av kretskortslayout-program, t.ex. P-CAD som används på LTH. När kretskortet är färdigritat så sätts två positioneringshål i på varsin sida (diagonalt) av kretskortet. Dessa kommer sedan att användas för att maskinen skall kunna avgöra om kretskortet ligger vridet eller inte, och i så fall kompensera för denna vridningen. Detta gör att användaren inte behöver vara noggrann med att allt ligger i rätt vinkel vid framkallning och montering. Varje håltyp som används i kretskortslayout-programet associeras även med en så kallad Tool Code som bestämmer vilken borrhål diameter som behövs för hålet. Maskinen kommer sedan att meddela när byte av borr skall ske.

Layouten exporteras till en fil i så kallad Gerber format (även kallat N/C Drill i P-CAD) som styrprogrammet kan importera och läsa in. Detaljer om hur detta görs finns som bilaga (Bilaga A). En lista på alla hål med deras verktygskod finns tillgänglig i styrprogrammet samt en lista över de borrhål storlekar användaren valt i layoutprogrammet. All data överförs till maskinen genom en enkel knapptryckning. Vilken COM-port maskinen är ansluten till behöver inte specificeras då styrprogrammet automatiskt kontrollerar vilka som finns tillgängliga och via handskakning letar upp vilken maskinen är ansluten till. Sedan överförs all data och maskinen meddelar via styrpanelen vad som användaren behöver göra för att förbereda borrhållningen.



Figur 4.1 – Styrprogrammet efter en fil laddats

Programmet subtraherar varje håls koordinat som skall borraras med det nedre positioneringshållets koordinat. Detta gör alltså att **det nedre positioneringshållet definieras som origo**. Positionerna som ligger lagrade i maskinens minne kommer alltså vara avståndet till det som användaren definierar som nollpunkt på kretskortet.

Varje hål sorteras med avseende på verktyg, med det minsta verktyget först. För att maskinen skall undvika att röra sig mer än den behöver vid borring, så sorteras hålen för varje verktyg på följande sätt:

1. Sätt position **p** = origo
2. Leta upp närmsta hål med avseende på **p** och sätt det nya hålet till **p**. I denna ordningen som det nya hålen hittas skall de borraras.
3. Flagga av hålet som hittat. Finns fler hål i samma verktygsserie som inte är avflaggade, upprepa ovanstående (2) kommando.

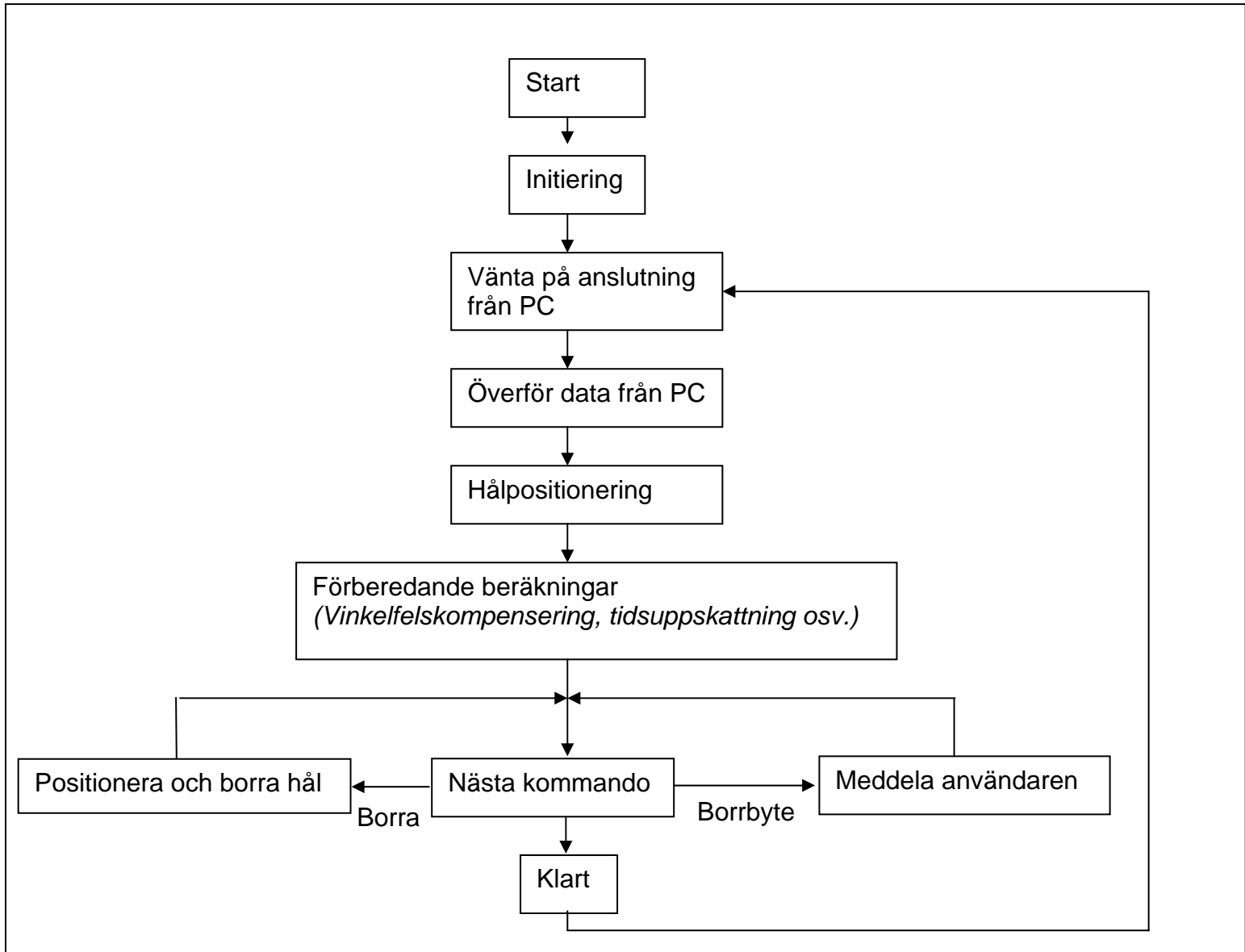
Detta görs enskilt för varje verktyg. Positionerna kommer att överföras till maskinen i denna ordning vilket gör att mikroprocessorn inte behöver sortera och bearbeta i vilken ordning som hålen skall borraras.

### **4.3 Mikroprocessorns mjukvara**

Mikroprocessorn (ATmega16) styr allting som finns på maskinen. Programmet är skrivet i C. Detta medför naturligtvis större kod men förenklar utvecklingsarbetet. Exekveringstiden för programmet är kort och ger inga flaskhalsar tidsmässigt i jämförelse med t.ex. skriv och lästid för EEPROM-minnet som används (tidigare kapitel). Som noterades innan så utelämnas programmeringsdetaljer då källkoden är väldokumenterad och tydlig.

Nedan syns en förenklad schematisk skiss som beskriver flödet i programmet. Varje steg i schemat beskrivs i detalj:





Figur 4.2 - Förenklad schematisk översikt över programflödet i mikroprocessorn

## ***Initiering***

Alla hårdvarufunktioner i mikroprocessorn initieras. Dessa är följande:

- *USART* (seriell kommunikation till PC:n): Sätts till 28.000 BAUD och 8 bitars datalängd.
- *Timers*. Timer1 (16 bitars upplösning) används för PWM-styrning av motorerna.
- *I2C* och *SPI* interface initieras

Även displayen på kontrollpanelen initieras enligt databladets instruktioner.

## ***Vänta på anslutning från PC***

Ett välkomstmeddelande visas på displayen och indikerar att maskinen väntar på anslutning från PC. Då användaren i styrprogrammet startar överföringen så skickar detta program ut ett handskaknings-kommando på varje COM-port för att leta upp vilken som maskinen är ansluten till. När detta kommando är mottaget så svarar maskinen, och överföringen av data startas.

## ***Hållpositionering***

I detta läge skall användaren markera ut två positioneringshål som sattes ut i kretskortslayout-programmet. Detta görs genom att hålet positioneras manuellt med hjälp av kontrollpanelen och sedan sänks borren ner i kretskortet. Djupet som borren har i Z-led används som referens för det nedre läget som borren skall ha vid borring. Positionerna som användaren markerat ut sparas sedan i minnet och jämförs med positionerna som angivits i layout-programmet. Vinkeln mellan det undre och det övre hålet jämförs, och om kretskortet ligger vridet så kommer detta att kompenseras. Som tidigare nämnts kommer det nedre positioneringshålet att definieras som origo som styrprogrammet i maskinen utgår från. Ett felvänt kretskort kommer alltså att vridas kring denna punkten för att kompensera vinkelfel.

### **Förberedande beräkningar**

Som beskrevs ovan så används hålpositioneringens koordinater tillsammans med positionerna från layout programmet för att ta reda på om kretskortet ligger snett. En mer detaljerad beskrivning av hur detta fungerar finns under kapitel 5.2 *Nollpunkter*. Vinkelfelet sparas som en variabel och kompenseras vid varje positionering.

För att kunna ge användaren en uppskattning till nästa verktygsbyte så beräknas även tiden som krävs för att borra alla hål för ett visst verktyg. Detta har lösts genom att mäta tiden det tar för att positionera olika långa avstånd med befintliga regleringsparametrar och sedan har dessa mätvärden plottats i Excel, där en matematisk funktionsanpassning gjorts. Naturligtvis stämmer detta inte exakt men med en tillräcklig noggrannhet för att kunna uppskatta tiden kvar med minutprecision.

Följande funktion beskriver tiden:

$$t(s) = 2.2536 \cdot s + 180$$

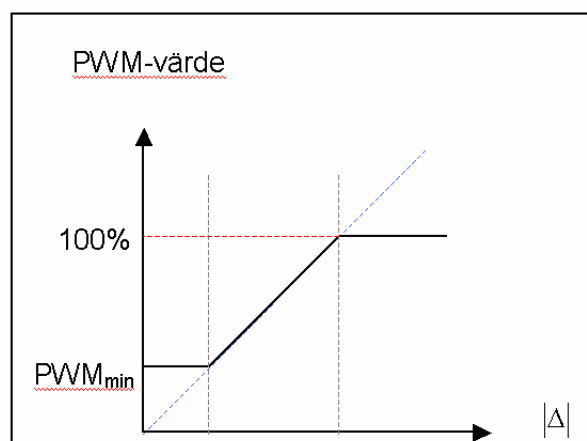
Där  $s$  är avståndet mätt i bitar ( $1 \text{ bit} \approx 0,366 \mu\text{m}$ ) och  $t(s)$  ger den uppskattade tiden för positioneringen i enheten  $s \cdot 10^{-2}$ . Anledningen till valet av enhet är för att värdet som funktionen returnerar skall passa in i en 16 bitars variabel (avståndet  $s$  kan med 12 bitars noggrannhet maximalt vara 4095, vilket gör att returvärdet passar). Utifrån hur långt som maskinen förflyttat sig vid varje positionering kan ett procentuellt värde på arbete som är utfört samt en tidsuppskattning på återstående tid i minuter tills nästa verktygsbyte visas på displayen.

### **Positionering**

Koordinater för varje hål som skall borrar ligger sparat i EEPROM minnet. Värdet som anges här är avståndet med avseende på det nedre positioneringshålet, som sattes till origo. När positionering skall göras så beräknas nästa position i maskinen som värdet som finns lagrat i minnet, och eventuellt kompenserat för vinkelfel.

Denna koordinat används som argument till en funktion som sköter positioneringsregleringen. Reglering som används är programmerad likt en PI-regulator. I normala fall används inte integraldelen utan detta är en säkerhetsåtgärd ifall motorn skulle fastna i närheten av det önskade värdet.

1. Läs in position från AD-omvandlaren och kalla denna position  $P_{AD}$ . Avståndet  $\Delta$  till den önskade positionen  $P$  beräknas då  $\Delta = P - P_{AD}$ . Beroende på om  $\Delta$  är positivt eller negativt så bestäms om motorn skall röra sig framåt eller bakåt.
2. Ett värde på motorstyrningens pulsbredd (*PWM-värde*) beräknas utifrån avståndet  $\Delta$  multiplicerat med en förstärkningsparameter  $K$ , och bestämmer således motorns hastighet. Det beräknade PWM-värdet kan aldrig överskrida 100% och aldrig bli mindre än ett visst tröskelvärde  $PWM_{min}$  figuren nedan. Tröskelvärdet är experimentellt bestämt och motsvarar den lägsta pulsbredden med marginal som krävs för att motorerna skall börja röra på sig. Genom att analysera stegsvar för olika längder så har parametern  $K$  bestämts till ett lämpligt värde. Ett för lågt värde på  $K$  gör att motorerna drivs onödigt långsamt samt ett för stort gör att motorn inte hinner stanna innan den har gått för långt och självsvängningar riskerar att uppstå innan motorn har reglerats till den önskade positionen.



Figur 4.3 – P-reglering

3. Om avståndet  $\Delta$  är mindre än ett förbestämt värde (1 bits längd är detta värdet satt till) under en viss tid (som är kort relativt hur fort motorn rör sig) så är motorn i rätt position och regleringen avslutas. Eftersom motorn reagerar snabbt på förändringar i pålagd spänning, den är lågt utväxlad samt att **K**-parametern har bestämts till ett korrekt värde så behöver inte kontrollen ske noggrannare än på detta sätt. Efter regleringsfunktionen har avslutats så kontrolleras avståndet till önskad position (alltså regleringsfelet) ytterligare igen, och om detta visar sig vara större än 2 bitar så anropas regleringsfunktionen igen. Dock har experiment visat att regleringsfelet sällan är större än  $\pm 1$  bit så detta är bara en extra försäkran om att rätt position har erhållits.
4. Integraldelen som reglerar bort konstanta fel beräknas på följande sätt: om motorns position har varit mindre än ett visst tröskelvärde under en viss tid så börjar integraldelens värde att öka. Detta värde läggs sedan till på pulsbreddsvärdet. Om däremot hastigheten är mindre så sänks integralvärdet. Parametrar på alla dessa värden finns tillgängliga överskådligt i koden och kan bestämmas enskilt för varje dimension. Inga av dessa värden är beräknade matematiskt utan studerade och provade fram via stegsvarsanalys.

```

// --- Regulator parameters -----
//
int REG_P_KParameter[3] =           X   Y   Z; // K factor (P-reg)
int REG_P_MinDistFullSpeed[3] =     {30, 30, 30}; // Min dist to get full speed

int REG_I_RampUpSpeed[3] =           {1,  1,  2}; // I ramp up value
int REG_I_RampDownSpeed[3] =         {10, 10, 10}; // I ramp down value
int REG_I_ThresholdLimit[3] =        {1000, 1000, 1000}; // I value threshold to active
int REG_I_MinLimitToActivateCnt[3] =  {3,   3,   3}; // Speed threshold to active

int REG_MotorSpeedMin[3] =           {200, 200, 280}; // Minimum motor speed

int REG_ZPosMaxSpeedDw = 450; // ZPos max speed when drilling (down)
char REG_Drilling = 1; // Is drilling?

#define TIMER1_TOP 1000 // Max TIMER1 PWM ramp value
#define POS_Z_RAISE 150 // Distance to raise when drilling
#define TimeToDrill 1300 // Time it takes to drill a hole (ms)
// -----

```

Figur 4.4 - Regulatorparametrar från mikroprocessorns källkod

5. För att skydda motorerna (och eventuella objekt) mot att något skulle t.ex. fastna i maskinen så utförs även en kontroll för detta. Detta skydd fungerar genom att om motorn har 100% pulsbredd (vilket den får även för korta avstånd via integraldelen i regleringen) samtidigt som hastigheten är låg under en kortare tid så räknas detta som ett fel, och borring bryts. Ett felmeddelande visas på displayen som indikerar att maskinen har stannat på grund av ett motorfel.
6. Pulsbreddslogiken som styrs av en hårdvarubaserad timer (Timer1) uppdateras efter det uträknade nya värdet på pulsbredden. Timern räknar från 0 till ett övre värde, när det övre värdet har nåts återställs den till 0. På detta sättet skapas en triangelvåg. Då timern återställs till 0 så sätts den digitala utgången till kraftelektroniken som styr motorn som regleras till 0, medans timerns värde når ett tröskelvärde (som är pulsbreddens storlek) så sätts utgången till motorn till 1. På detta sätt skapas en pulsbreddsmodulerad utsignal på traditionellt sätt med konstant frekvens. Bärfrekvensen bestäms av hur snabbt timern räknar upp samt värdet på den övre gränsen.
7. Status för hur långt det är tills nästa verktygsbyte uppdateras.
8. Alla dessa steg upprepas tills regleringen är klar. En fördröjning mellan varje upprepning gör att regleringen inte sker för snabbt. Detta eftersom processorn är snabb och hade hunnit utföra många upprepningar på kort tid hade den tolkat det som att motorn stod still fastän att den rörde sig på grund av positionsavläsningen sker digitalt, alltså med diskreta steg.

### **Verktygsbyte**

Displayen på kontrollpanelen indikerar att ett verktyg skall bytas. När användaren gjort detta så verifieras detta med en knapptryckning och programmet går vidare.

När allt är klart så visas detta på displayen och borren ställer sig i startposition. Kortet är nu färdigt.

## **5. Maskinen och styrsystemet**

### **5.1 CNC-styrning**

Tekniken som ska styra bormaskinen heter CNC (Computer Numerical Control). Detta är ett datorsystem för verkstadsmaskiner (svarvar, fräsar, borrar, slipar, mm) som används i tillverkningsindustrin. Med hjälp av styrsystemet kan man tillverka komplicerade detaljer automatiskt med hög precision. Tekniken passar även utmärkt för tillverkning av enstaka kretskort. En annan fördel med tekniken är att maskinen sköter sig själv efter att programmet har körts igång.

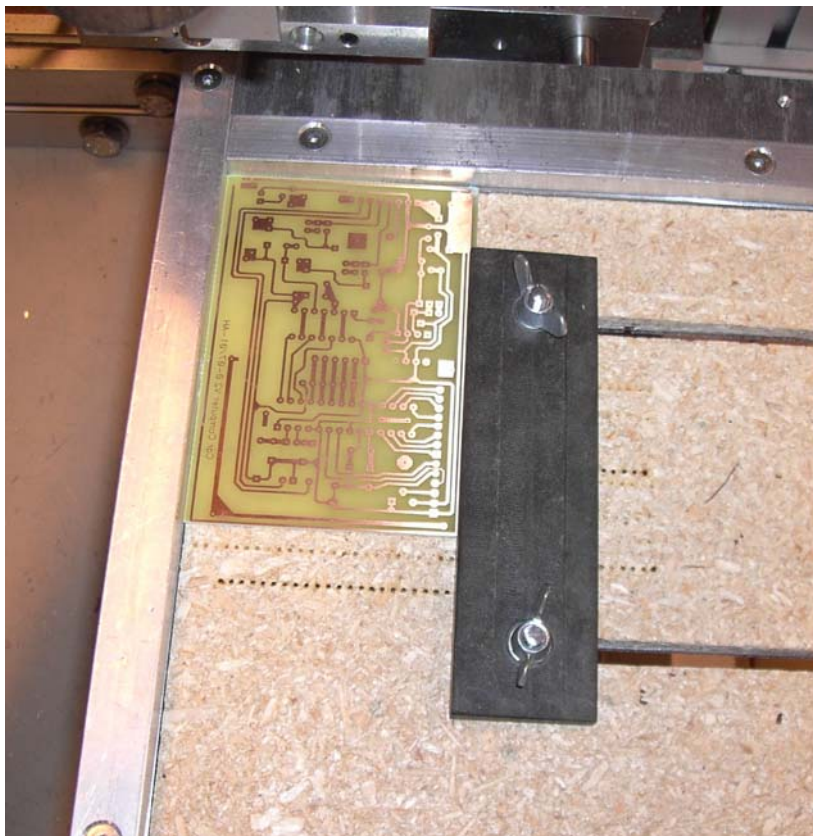
Det som maskinen ska kunna göra är att borra hål till komponenternas ben efter att kretskortet har etsats. Dessa hål är ofta i olika storlekar för olika typer av komponenter, typiskt är 2 – 3 olika diametrar på hålen för ett kretskort. Detta betyder att byte av borr måste göras under borrprocessen och detta kommer att ske manuellt.

### **5.2 Nollpunkter**

I en CNC-styrd maskin finns oftast en så kallad maskinnollpunkt i maskinens eget koordinatsystem. Denna nollpunkt är oftast placerad nära ändlägena i någon riktning på koordinataxlarna och vanligtvis körs maskinen till denna nollpunkt vid uppstart. Det finns sedan möjlighet att flytta nollpunkten till önskad position med tanke på det program man vill köra.

För att maskinen ska veta var den ska borra hålen så kommer hålens koordinater att läsas in till EEPROM-minnet från det program som kretskortet har designats med. Två av dessa hål har satts ut av användaren för att kunna ställa in maskinen för det aktuella jobbet. I det första hålet kommer nollpunkten för kretskortet att sättas i X-, Y- och Z-led och det andra hålet kommer att användas för att kompensera för vinkelfel, dels från etsningen, dels vid uppspänningen. Dessa två hål kommer att sitta på diagonalen tvärs över hela kretskortet, detta för att få en så lång sträcka som möjligt för att kunna minimera vinkelfelet. Mjukvaran kommer sedan att kompensera för fel så att alla hålen kommer att borraras på rätt plats.

För att kunna positionera borren rätt över hålen används maskinens kontrollpanel. Det är viktigt att maskinoperatören är väldigt noggrann vid positioneringen då alla andra koordinater sedan kommer att utgå ifrån detta. Ett fel här kommer att ge följdfel till de övriga hålen.



*Figur 5.1 – Kretskortet monterat och fastspänt inför bearbetning*



Det kan finnas två olika vinkelfel när kretskortet har placerats i maskinen inför en borrarprocess. Dels finns det risk att hela kretskortet ligger snett i förhållande till maskinens koordinatsystem. Detta kan bero på att kanterna på kretskortet inte är helt i vinkel. Dels kan det ha uppstått ett vinkelfel när kretskortet etsades dvs. om inte etsfilmen hamnar rakt i förhållande till kretskortets kanter. Genom att beräkna den verkliga vinkeln sett utifrån maskinens koordinatsystem så kan denna jämföras med vinkeln som positioneringshålen fick i kretskortslayoutprogrammet. Med enkel trigonometri kan detta fel korrigeras när detta vinkelfel är beräknat.

### **5.3 Kalibrering och testning av maskinen**

För att maskiner av denna typ ska hålla de krav som ställs på dem måste en kalibrering utföras. Ute i industrin görs kalibrering på maskiner och mätton enligt de regler som finns i företagets kvalitetscertifiering. Ofta någon gång per år men i vissa fall oftare eller mer sällan beroende på vilka krav som ställs. Det är väldigt viktigt att maskiner håller de utsatta kvalitetskraven så att de inte tillverkar felaktiga produkter vilket kan bli mycket kostsamt.

För att maskinen ska kunna hålla de kvalitetskrav som förväntas har en kalibrering gjorts. För att kalibrera maskinen har en testborrning utförts enligt följande. Fyra hål borrades i en fyrkantig plexiglasskiva. Hålen borrades i skivans hörn en liten bit in från kanterna med ca 100 mm mellanrum. Avståndet mättes sedan upp med skjutmått och kompensering i mjukvaran utfördes efter det reella måttet. För att sedan kontrollera att det nya måttet stämde gjordes en automatisk positionering för varje hål.

Det har även utförts flera tester av maskinens noggrannhet. Testerna har gjorts genom att rita ut olika kretsar och hål i mjukvaran för kretsdesign. Sedan har designen skrivits ut på papper och fästs på en hård yta som kan borraras. På detta sätt har avvikelser och andra fel upptäckts och rättats till. Det har även utförts tester genom att borra riktiga kretskort som har etsats på institutionen.

## 6. Problem och svårigheter

I detta kapitel kommer de problem och svårigheter som har uppstått under projektets gång att diskuteras. Problem och lösning kommer att vara presenterade. Flera problem och fel har undvikits genom att lösa saker och ting på ett eller annat sätt med tanke på designen.

### 6.1 Lägesgivarna

Det upptäcktes vid en provborring att lägesgivarna är olinjära. Felet uppenbarade sig genom att hål i vissa områden av arbetsytan hamnade utanför toleransen. För att på ett bättre sätt identifiera och isolera problemet ritades hål på en lång rad i P-CAD över hela arbetsområdet. Hålen skrevs sedan ut på ett papper och limmades på en träskiva och provborrades i både X- och Y-led. Det visade det sig att några hål inte hamnade på rätt ställe i vissa områden. För att lösa detta problem skrevs en kompensering i mjukvaran som sedan testades och justerades tills hålen hamnade på rätt ställe.

### 6.2 Problem vid detaljtillverkning

Vid tillverkning av de mekaniska detaljerna har det uppstått lite problem. Framst vid borring i de aluminiumprofiler som har använts. Eftersom ytan inte var jämn överallt där hålen har borrats så gled borret lätt åt sidan och hålet hamnade inte exakt där det skulle vara. För att avhjälpa detta har ibland större hål borrats än vad som har varit nödvändigt för att skruvarna ska passa.

Genom att göra detta kan detaljerna riktas in så att de hamnar i rätt läge och vinkel vid monteringen och positionen sedan fixeras med skruvförband. Det har dock krävts en del mätning med olika mätdon (skjutmått, stålskala) och vinkelhake för att få rätt på detta. Andra saker som har bidragit till detta problem är att det är en ganska gammal bormaskin/fräs som har använts samt att det är alltid svårt att mäta in, markera och få hålen på rätt ställe. Det är alltså flera felkällor som har bidragit till dessa problem.

### **6.3 Glapp i maskinens rörliga delar**

På maskinens rörliga delar (se figur 2.1) finns ett glapp mellan gängstången och den gängade mässingsbiten. Glappet har inte varit något problem vid positioneringen som man skulle kunna tänka sig. Det beror på att positionen har avlästs från lägesgivarna och inte genom att motorn har snurrat ett visst antal varv eller en viss tid för att komma till rätt position. Däremot har det uppstått ett problem som beror på glappet när maskinen ska borra. När borren kommer i kontakt med kretskortets yta så trycks borrens upphängning upp så mycket som glappet är stort och borret styrs åt sidan en aning. Detta kan göra att maskinen borrar fel. För att avhjälpa detta så har en tyngd på ca 1 kg fästs på bormaskinen så att detta ska motverka att den trycks upp. Denna lösning gjorde att borren styrde mycket mindre åt sidan eller inte alls.

## 7. Resultat

Det viktigaste målet med projektet har varit att tillverka en kretskortsbormaskin som kan användas av institutionen för att borra hål i kretskort för komponentmontering. Andra mål med projektet finns specificerade under rubriken målsättning i rapporten. Dessa mål har uppfyllts på ett bra sätt och resulterat i en maskin som kommer att användas av institutionen. Maskinen kommer att underlätta det arbete som krävs för att borra hålen och både tid, borrhast och tålmodighet kommer att sparas.

Maskinens hastighet är något som inte har prioriterats. Tiden det tar att borra ett kretskort beror i huvudsak på två faktorer: antal hål och avstånd mellan hålen. Även om det är flera olika håldiametrar påverkar tidsåtgången. Det tar ca 1 - 2 sekunder att fullfölja själva borrhastprocessen (ner och upp) och maskinens förflyttningar har en hastighet på ca 10mm/sekund. Uppskattningsvis tar det ca 5 minuter att borra ett mindre (200 hål) kretskort medan ett stort (2000 hål) tar ca 15 minuter. Detta är ganska grova uppskattningar.

Maskinens precision är något som däremot har varit viktigare. Det har utförts flera tester för att prova precisionen och kompenseringar har gjorts i mjukvaran i områden där precisionen har varit sämre. Kalibrering av maskinen är också något som har förbättrat noggrannheten så att hålen hamnar på rätt ställe.

Överlag har det varit ett mycket intressant och lärorikt projekt att genomföra. Det har varit ett ganska praktiskt examensarbete som har krävt kunskap inom flera olika områden. Det är väldigt roligt att ha lyckats med projektet och att ha byggt en maskin som kommer att användas av institutionen.

Något numeriskt värde på noggrannheten är svårt att uppskatta exakt men provborringar och tester ger en noggrannhet på ungefär 0,1 mm.

## 8. Specifikationer

- Konstruktion: chassi av aluminiumbalkar
- Bearbetningsyta: 120x120 mm
- Bordsyta: B205 x D165 mm
- Maskinmått: B450 x D540 x H540 mm
- Vikt: 25 ca kg
- Driftsart: DC-linjärmotorer
- Lämpar sig för bearbetning i plastmaterial
- Styrts direkt från kontrollpanel för att borra och positionera
- Chuck med expander, 1,0 – 3,2 mm
- Spindelhastighet: 5000 – 20 000 varv/min

## 9. Budget

Priserna är ungefärliga. Ej köpta delar har uppskattade priser.

Motordrivkretsar	150 kr x 3
Mikroprocessor	60 kr
Display	160 kr
USB modul	145 kr
AD-omvandlare	50 kr
EEPROM minne	50 kr
Kontakter och knappar	200 kr
Inkapslingar	300 kr
Borrmaskin	2000 kr
Nättaggregat	300 kr
Diverse komponenter	200 kr

**Summa: Runt 4000 kr**

## 10. Referenser

[1] <http://www.cadsoftusa.com/>

[2] Datorblad hos Elfa för motordrivkrets. Hämtad 29 januari, 2007: <http://www.elfa.se/pdf/73/731/07314636.pdf>

[3] Datorblad hos Elfa för microprocessor (Atmel ATmega16). Hämtad 29 januari, 2007: <http://www.elfa.se/pdf/73/736/07367246.pdf>

[4] Datorblad hos Elfa för AD-omvandlare. Hämtad 29 januari, 2007 <http://www.elfa.se/pdf/73/734/07342058.pdf>

[5] Datorblad hos Elfa för EEPROM-minne. Hämtad 29 januari, 2007 <http://www.elfa.se/pdf/73/739/07397169.pdf>

[6] <http://www.ftdichip.com/>

[7] Datorblad hos Lawicel för USB-modul. Hämtad 29 januari, 2007 [http://www.ftdichip.com/Documents/DataSheets/Modules/DS\\_MM232R\\_v100.pdf](http://www.ftdichip.com/Documents/DataSheets/Modules/DS_MM232R_v100.pdf)

[8] <http://msdn2.microsoft.com/en-us/vstudio/default.aspx>

[9] <http://www.microsoft.com/downloads/details.aspx?familyid=0856each-4362-4b0d-8edd-aab15c5e04f5&displaylang=en>

[10] [http://www.atmel.com/dyn/products/tools\\_card.asp?tool\\_id=2725](http://www.atmel.com/dyn/products/tools_card.asp?tool_id=2725)

[11] <http://sourceforge.net/projects/winavr/>

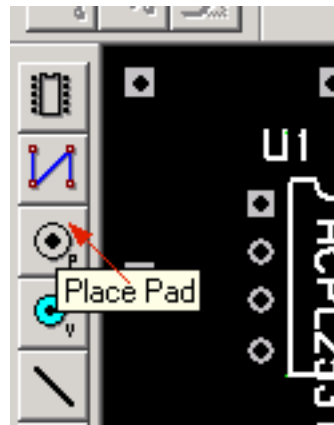
[12] Display, Hämtad 29 januari, 2007. <http://www.lawicel.se/datablad/dog-me.pdf>

# Bilaga A

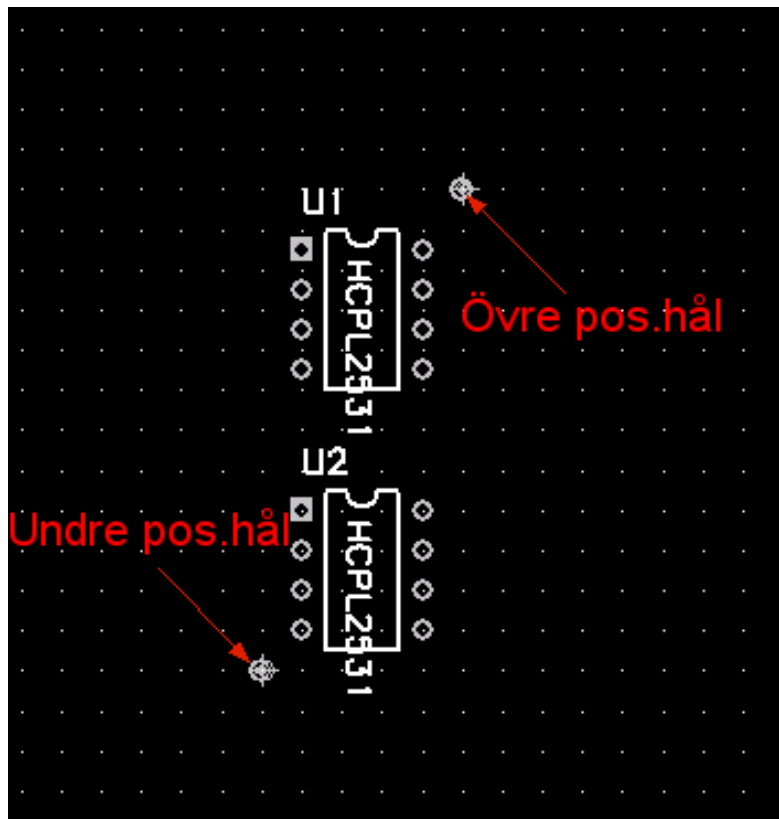
## Instruktionsmanual

### 1. Markera positioneringshålen i kretskortslayouten

- Öppna layouten i P-CAD PCB till kretskortet som skall borraras.
- Lägg till två nya hål (genom att välja **Place pad**)



*Place pad*



*Exempel på positioneringshålens placering*



- Tänk på att relativt stora hål till borren så att den kan styra in rätt
- **Markera dessa två hål, högerklicka och välj properties.** Dessa hål skall markeras som positioneringshål genom att de väljs till en egen Pad style med en egen tool code. Tryck på **pad styles**, kopiera en befintlig pad style (med copy...), döp denna till t.ex. Calibrate och tryck ok, markera denna i listan och tryck på modify (simple). Välj en unik håldiameter som inte används av något annat hål, t.ex. 0.77mm. Shape kan med fördel sättas till t.ex. mounting hole så syns dessa positioneringshål som ett kryss. Positioneringshålen kan läggas ovanpå ett befintligt hål på kretskortet. **Sätt alltid det undre positioneringshålet till vänster om det övre och heller inte inne på kretskortsytan, de skall vara de yttersta hålen eller ligga på de yttersta**, se exempelbilden ovan.

**OBS! Notera att kalibreringshålen kommer att spegelvändas p.g.a. framkallningen**

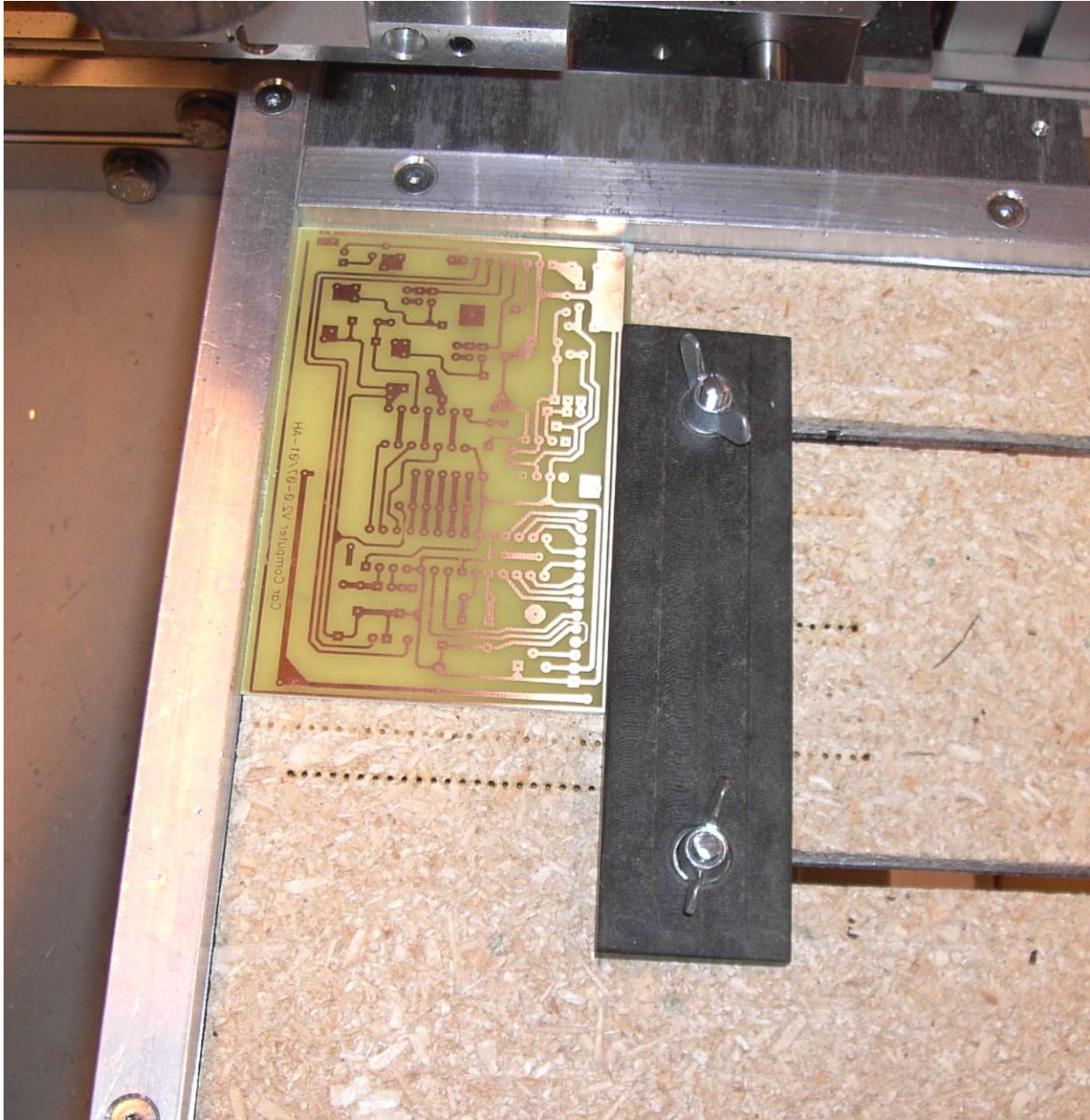
## 2. Export av kretskortslayouten till NCD-fil

- Tryck på menyn **File** och välj **"N/C Drill..."**.
- Välj **"Setup Output files"**. Markera det lager som skall borraras i Layers-listan (t.ex. om ett enkelsidigt kort skall borraras, välj bara bottom layer. Om dubbelsidigt används så ska varje lager borraras för sig)
- Skriv **NCD** på **"File extension"**
- *Filen kommer att lägga sig i samma katalog som P-CAD PCB-filen ligger i om man inte ändrar sökvägen i "Output path"*
- Klicka på **Add** och sedan **Close**
- Välj **Tools**. Under Tools skall varje hålstorlek tilldelas en borrhyp. Markera hål i listan och välj **Assign**. När assign väljs kan borrhypstorlekar (Tools) läggas till. Lägg till lämpliga diametrar (t.ex. 0.8mm, 1.0mm osv) och ge dessa en unik Tool code. **Positioneringshålen skall ha Tool code 80**. Välj close.
- Under **"N/C Drill format"**, se till att följande alternativ är valda:
  - Output units = millimeters
  - Output code type = ASCII none
  - Zero suppression = Leading
- Tryck på generate output files

En fil genereras med filändelsen NCD i arbetskatalogen där PCB-layouten ligger (filen har samma namn som PCB-filen)

### 3. Montering av kretskort i maskinen

- Placera kretskortet i maskinen enligt följande bild:



*Kretskortet monterat och fastspänt inför bearbetning*

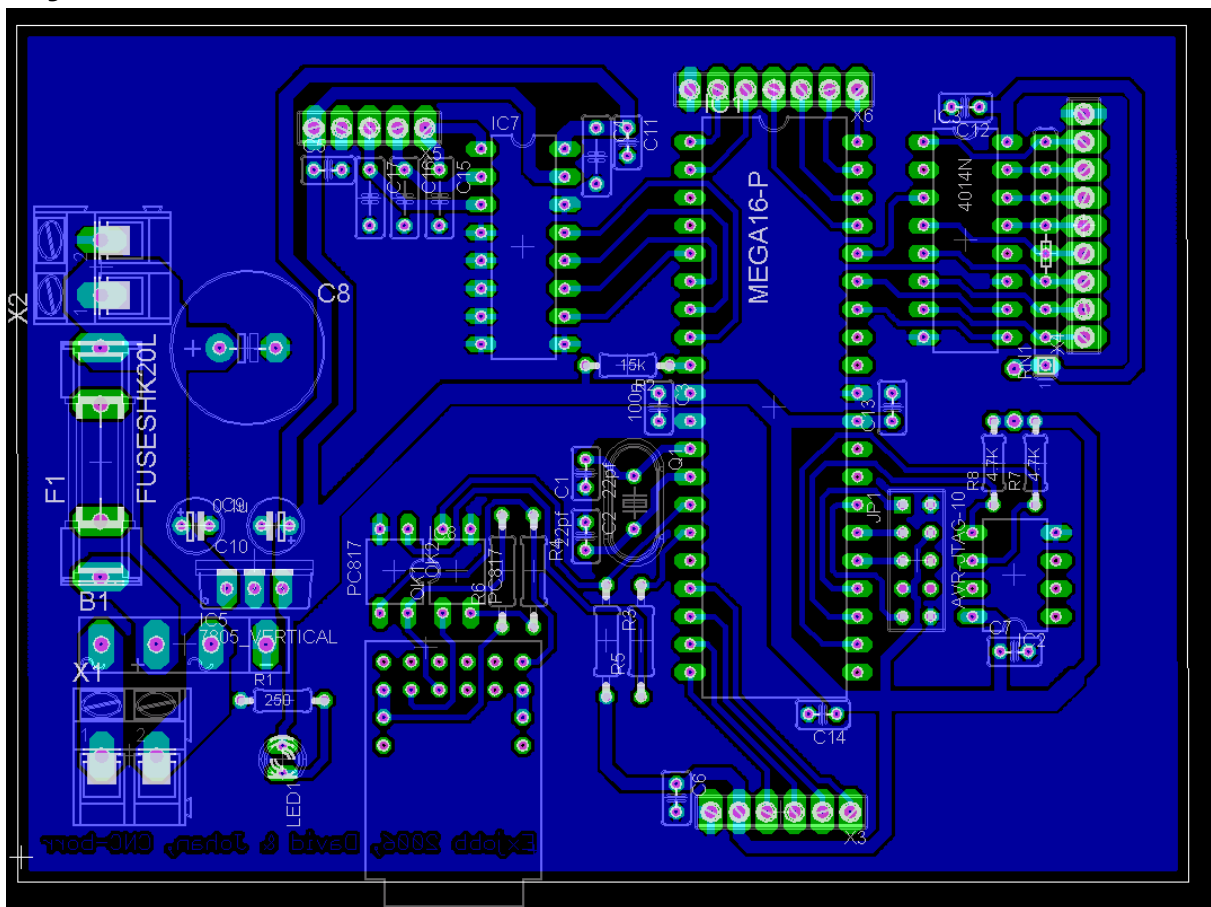
## 4. Maskinuppstart

- Sätt i nätsladden och USB-sladden till maskinen (och datorn). Nätsladden skall vara på ovansidan av elektronikboxen på baksidan av maskinen, och USB-sladden ansluts undertill.
- Öppna programmet "**CNC Drill**". Detta används för att kommunicera med maskinen från datorn. Tryck på **Browse** (under Import file) och välj NCD-filen som skapades i steg 2. Nu ansluter programmet till maskinen och överför all data.
- Displayen på kontrollpanelen till maskinen visar nu: "Change tool to: Position hole tool". Sätt i borren med den minsta diametern som skall användas och tryck på Ok.
- Nu visas: "Set position to: lower zero". Detta innebär att positioneringshålet närmast origo skall borraras (se bild nedan). Leta upp detta hålet med hjälp av kontrollpanelen och borra ner i Z-led så djupt som önskas. När Ok trycks så kommer denna positionen att sparas och återanvändas för alla andra hål. Maskinen ställer sig nu på positionen för det övre markeringshålet (och visar "Set position to: upper zero"). Korrigera om detta behövs, (på grund av att t.ex. kortet ligger snett) borra ner och tryck på Ok.
- Följ anvisningarna på displayen

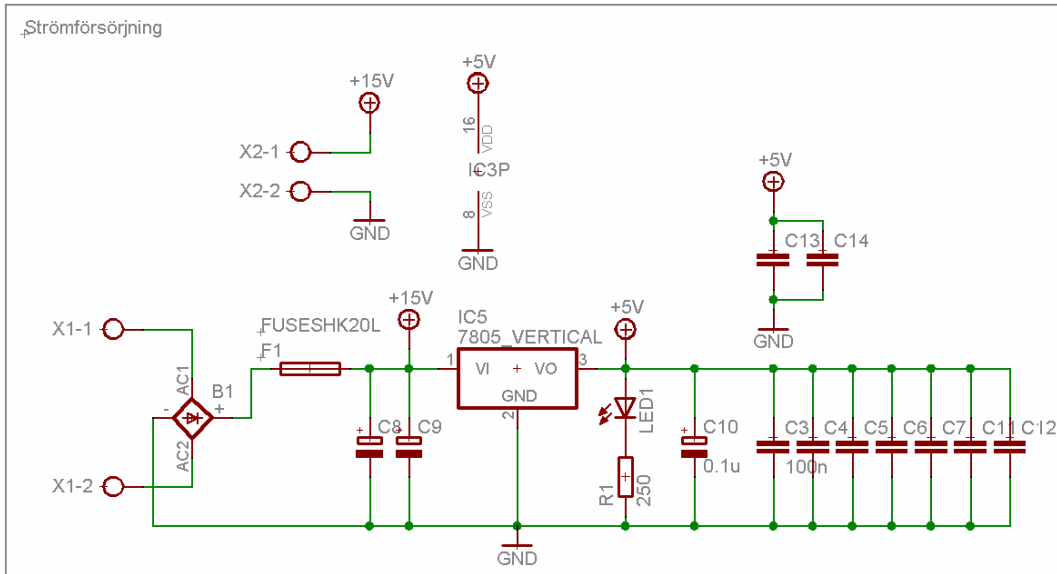
# Bilaga B

## Krettschema

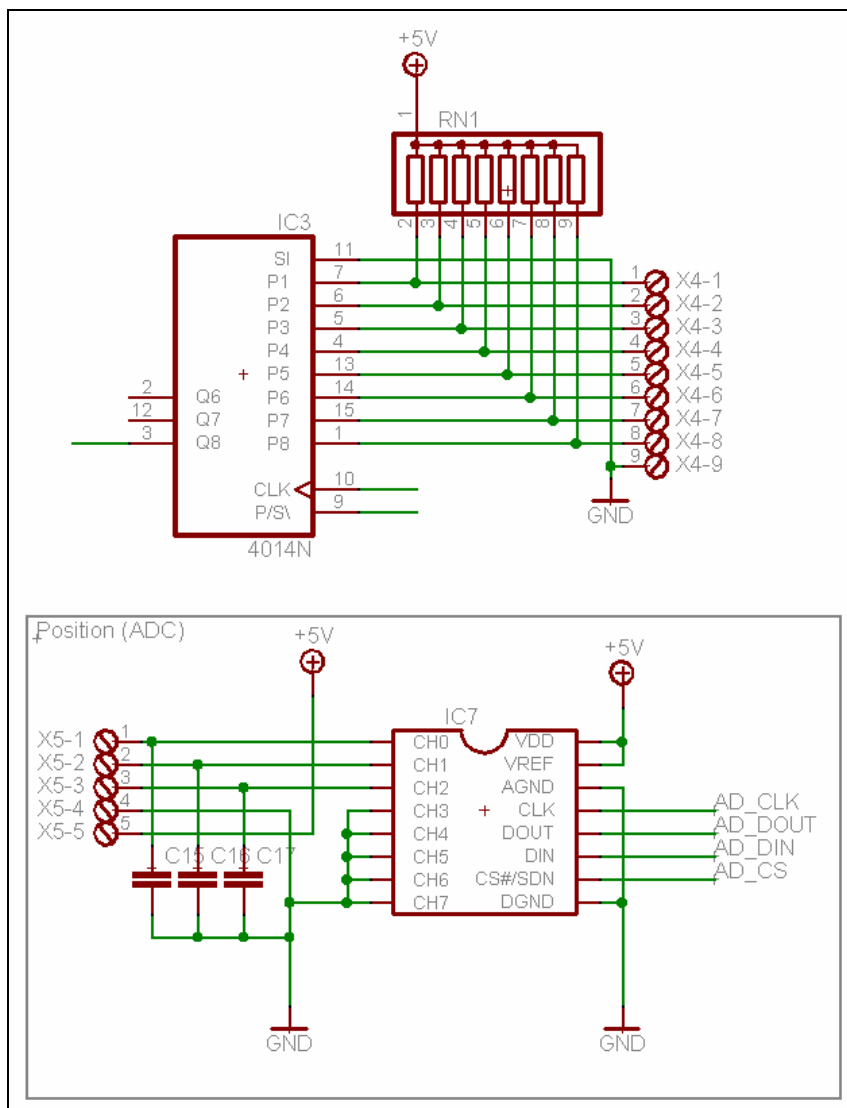
### Styrkortet



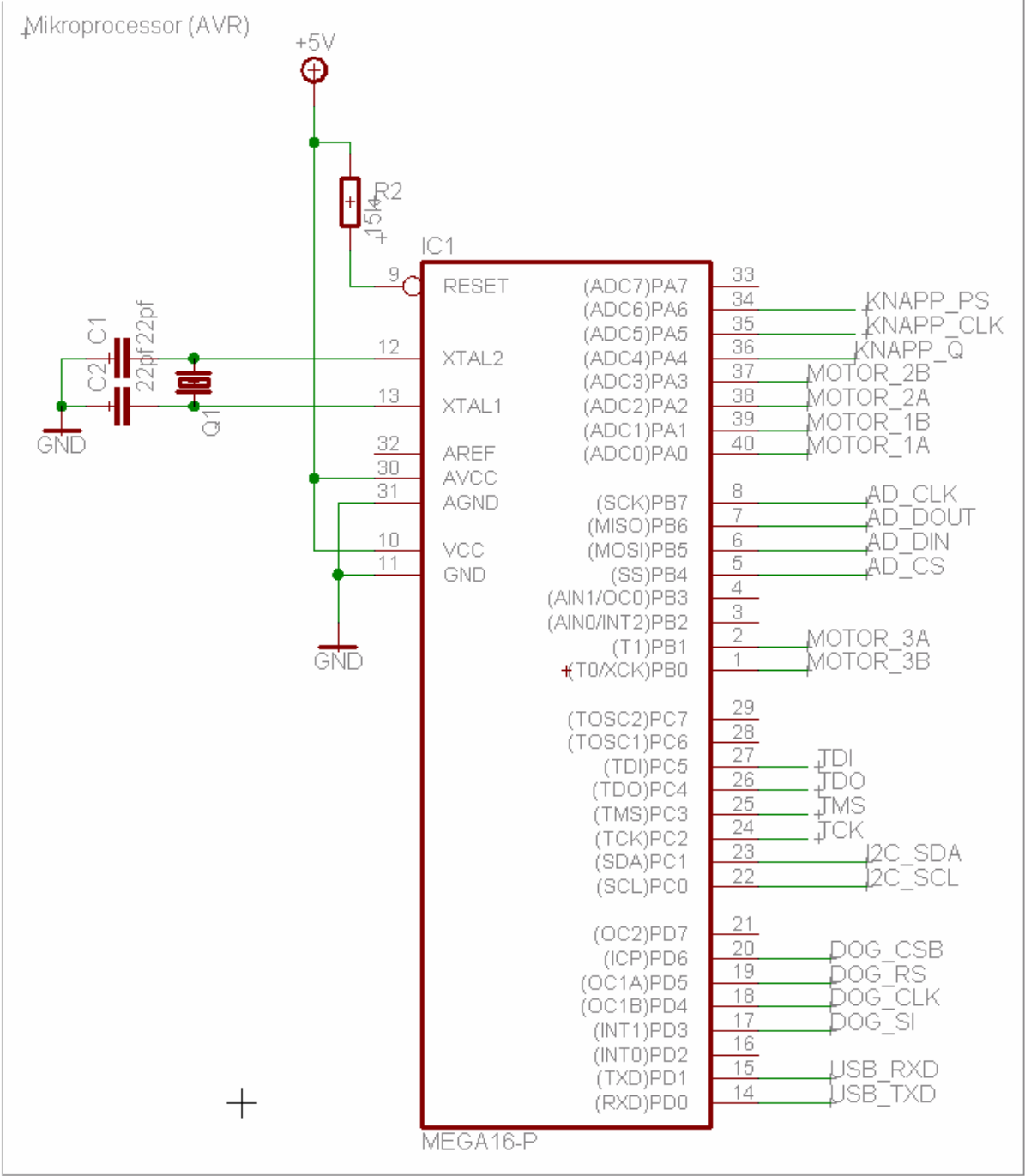
Styrkortets PCB layout



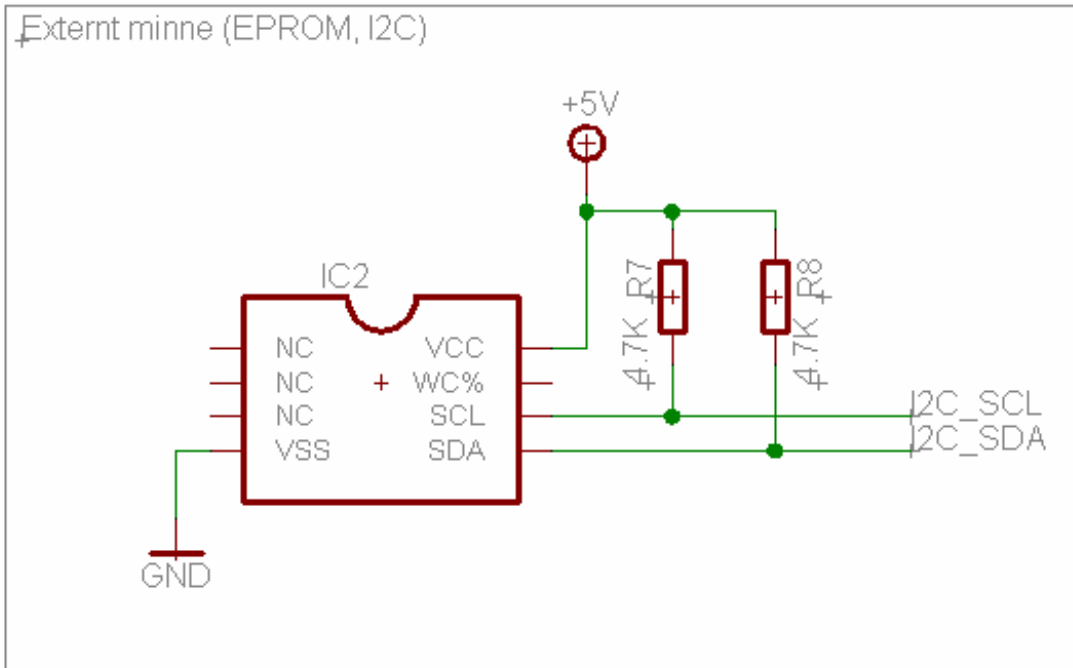
Strömförsörjningen till styrkortet



AD-omvandlare och skiftregister till tangentbordet

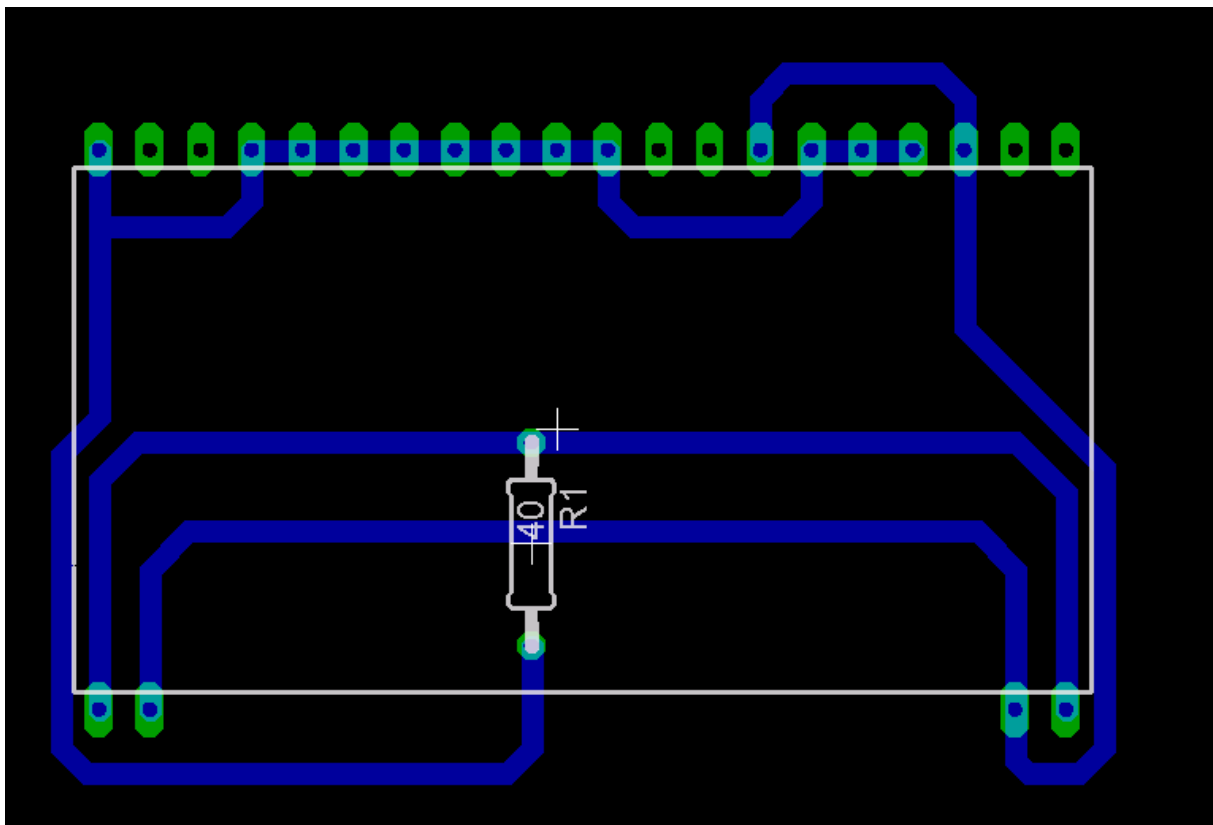


Mikroprocessorn

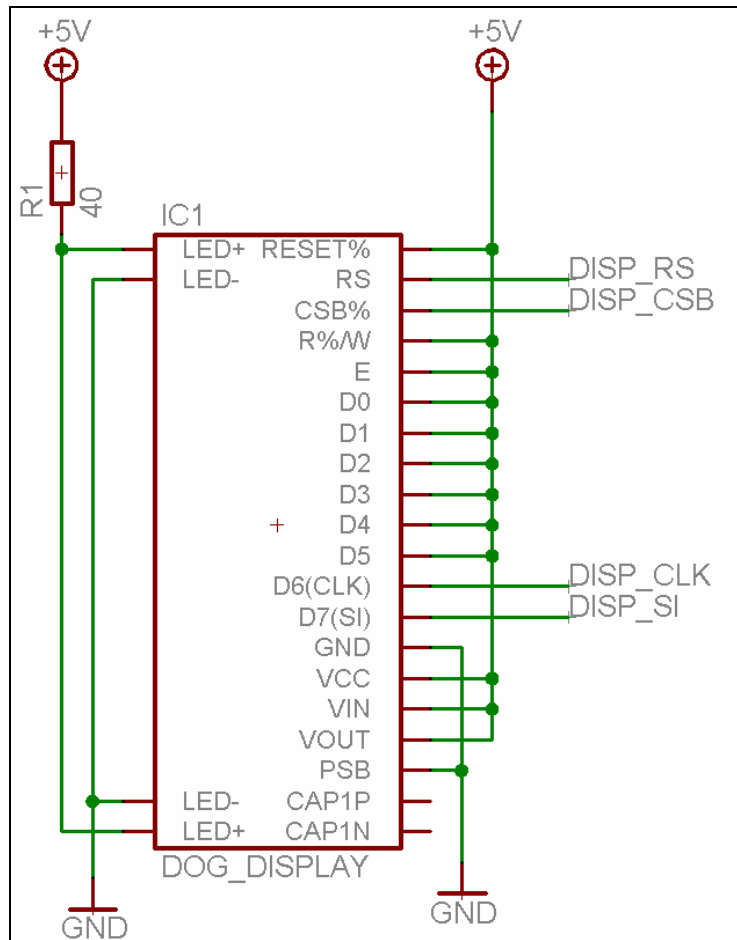


*EEPROM minnet med I2C-interface*

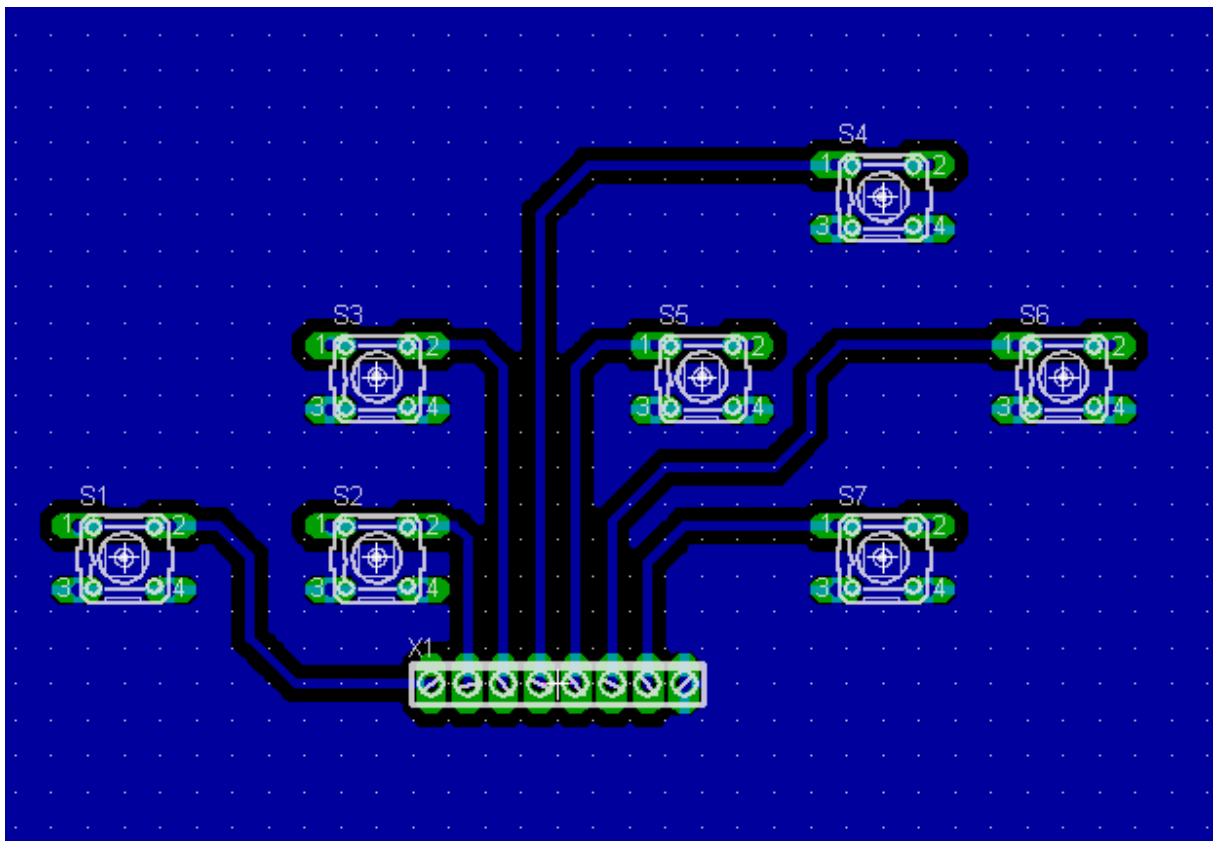
## Kontrollpanelen



*Displayens PCB layout*

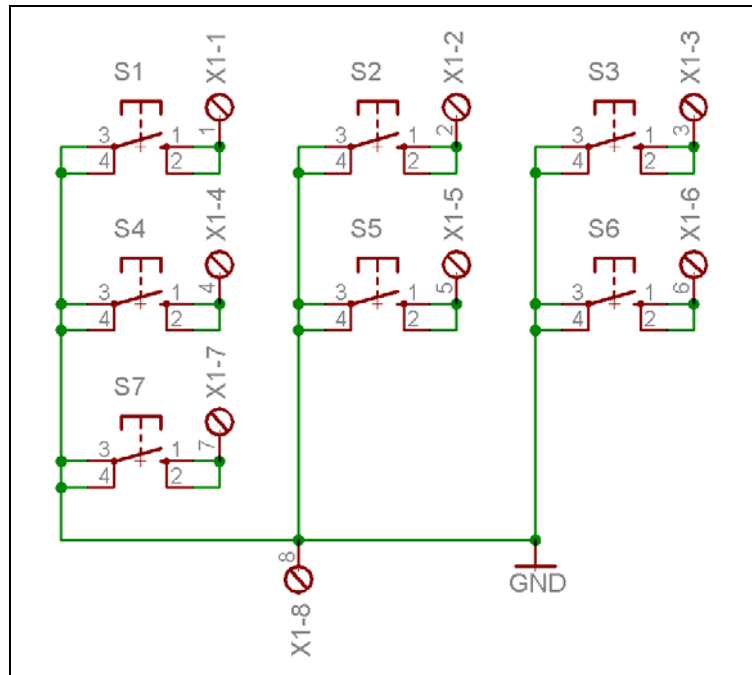


*Displayen på kontrollpanelen*



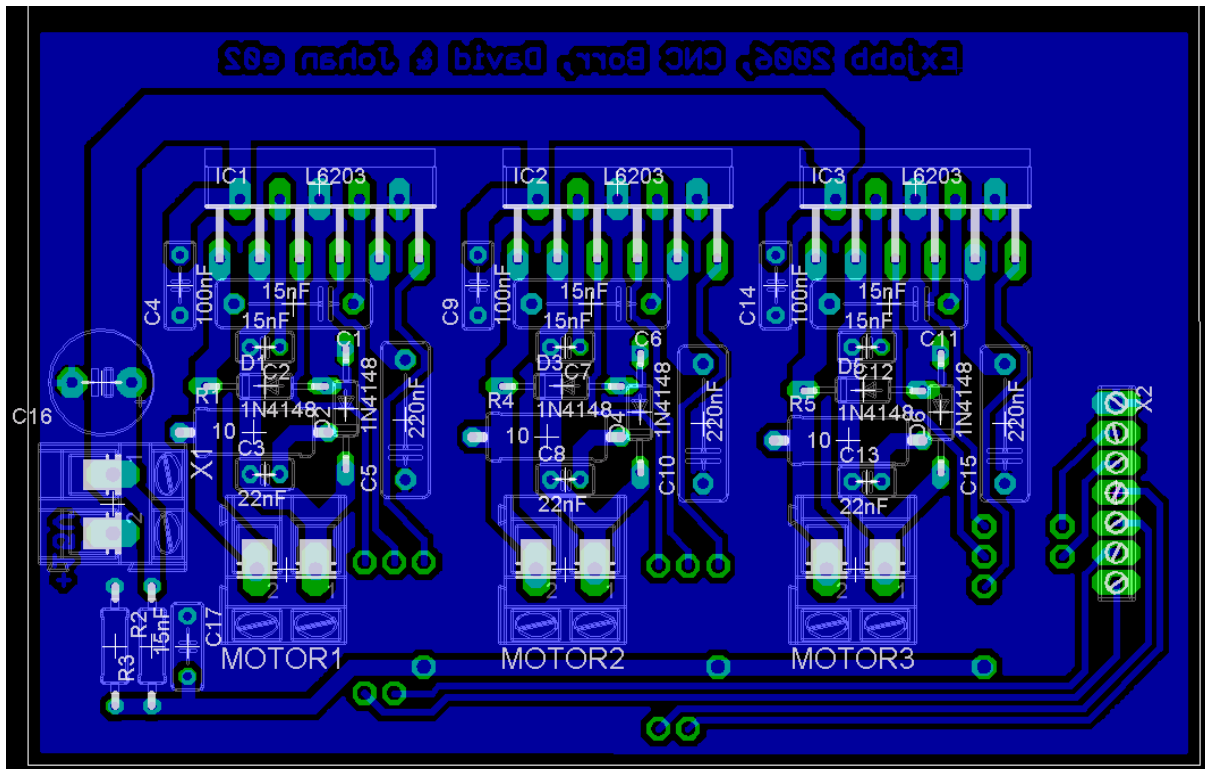
*Knappsatsens PCB layout på kontrollpanelen*



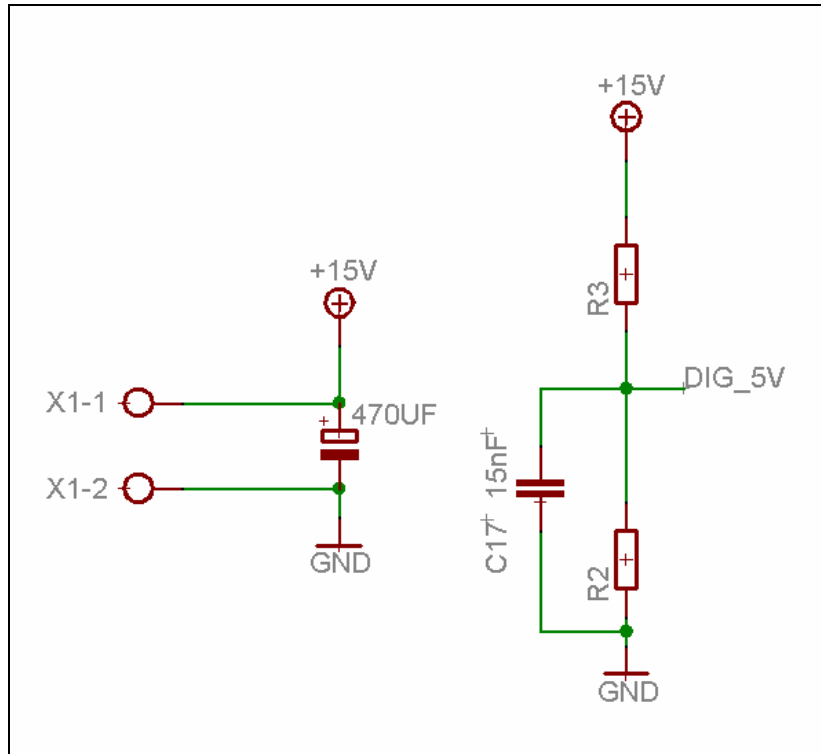


*Knappsatsen*

## Motordrivkortet



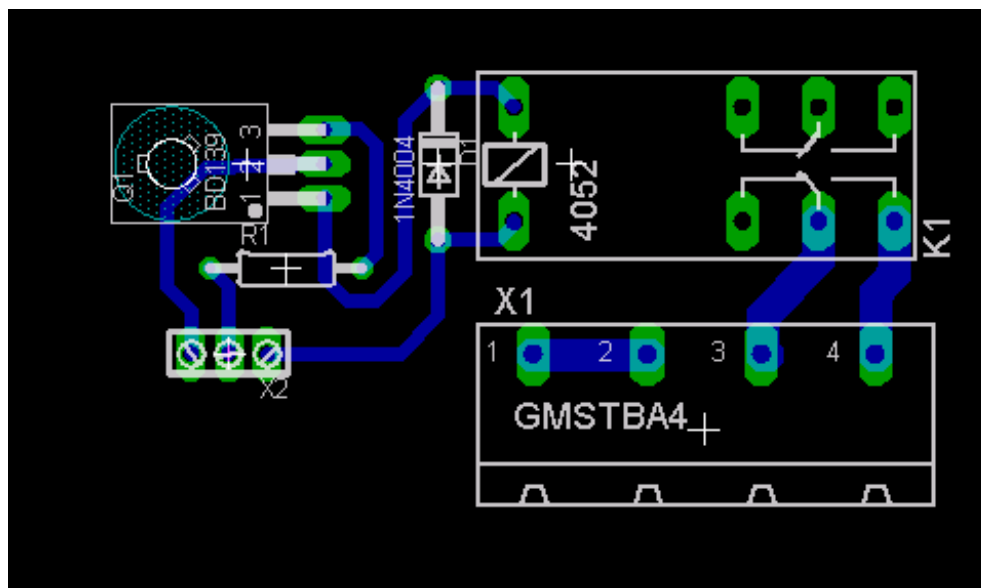
*Motordrivkortets PCB layout*



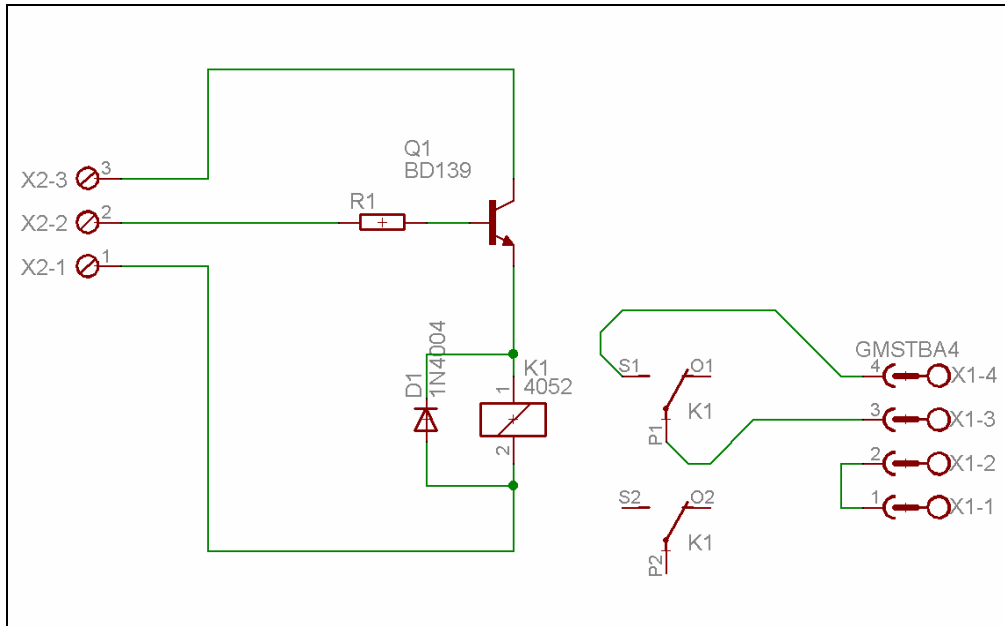
*Avkoppling och 5V signal*

Se under rapportens motordrivkort-del för att se kretsschema på ett drivblock.

## Reläkortet



*Reläkortets PCB layout*



*Reläkortet*

Layout för transformatorkortet finns ej tillgänglig då detta var ett färdigt kort från institutionen som har återanvänts.