

# Utveckling av styrelektronik till en mobil mätrobot

Maria Kjelldahl

Luleå tekniska universitet  
Civilingenjörsprogrammet  
Rymdteknik  
Institutionen för Systemteknik  
Avdelningen för EISLAB

## Förord

Detta examensarbete utgjorde den sista delen i min utbildning till civilingenjör i Rymdteknik. Arbetet har genomförts i Luleå under läsåret 2005-2006.

Examensarbetet har huvudsakligen bestått av konstruktion av styrelektronik till en robot samt programmering av de mikrokontrollrar som utgör en del av denna elektronik. Arbetet har varit intressant och givit mig mycket värdefulla erfarenheter inför framtiden.

Jag vill tacka alla som hjälpt mig under arbetets gång. Speciellt vill jag tacka

Kalevi Hyyppä, min handledare och examinator som bidragit med tips och råd under arbetets gång,

Catherine Mann, för hjälp med hantering av Layout-programmet och etsning av kretskort,

Kennet Hartvik, för hjälp med finjusteringar av kretskortslayouten och beställning av kretskort,

Jerry Lindblom, för hjälp vid programmeringen av mikrokontrollrarna och för tips och hjälp vid ytmontering av komponenter, samt

Håkan Fredriksson, som hjälpt mig att förbättra mjukvaran och regleringen samt skrivit det MatLab-program som används för att testköra roboten.

Maria Kjelldahl

Luleå, 2006-10-25

## Sammanfattning

Denna rapport behandlar utvecklingen av styrelektronik för att styra likströmsmotorerna i en mobil robot. Målet var att roboten skulle kunna styras trådlöst via Internet. Ett motorstyrkort har konstruerats. I elektroniken ingår bland annat en mikrocontroller, som programmerats för att reglera motorens hastighet. Mikrokontrollern tar emot kommandon från en PC, och motorhastigheten regleras genom pulsviddsmodulering via en H-brygga. Till roboten används tre motorstyrkort, ett per motor.

På grund av tidsbrist är inte det slutgiltiga målet med roboten uppnått, men själva styrelektroniken är klar. Motorstyrkortet har testats och fungerar bra. Mikrokontrollerns program för reglering av motorhastigheten fungerar också. Tanken är att en PC104 ska installeras i roboten, och att denna ska skicka kommandon till mikrokontrollerna samt kommunicera trådlöst med andra datorer.

## **Abstract**

This report describes the development of control electronics for the DC motors in a mobile robot. The electronics controls the speed of the motors according to commands from a computer. The objective was to be able to control the robot via Internet through wireless communication. A Motor Control Card was constructed. An important part of the electronics is a microcontroller, which is programmed to regulate the motor speed. The microcontroller accepts commands from a PC. Pulse width modulation and an H-bridge are used to regulate the motor speed. Three motor control cards are used in the robot, one for each motor.

Due to lack of time, the final objective has not been achieved, but the control electronics are finished. The Motor Control Card has been tested and it works. The program in the microcontrollers that regulates the motor speed works too. The plan is that a PC104 should later be installed in the robot, and that the PC will send commands to the microcontroller. The PC104 will also take care of the wireless communication with other computers.

# Innehållsförteckning

<b>1 INLEDNING.....</b>	<b>5</b>
SYFTE.....	5
MÅL.....	5
AVGRÄNSNINGAR.....	5
<b>2 BAKGRUND.....</b>	<b>6</b>
OM ROBOTEN.....	6
MOTORERNA.....	6
<i>Motorernas egenskaper.....</i>	<i>7</i>
PULSVIDDSDMODULERING.....	8
<b>3 UTVECKLING AV HÅRDVARA.....</b>	<b>9</b>
ÖVERSIKT AV STYRELEKTRONIKEN.....	9
VAL AV MIKROKONTROLLER.....	10
H-BRYGGA.....	10
<i>H-bryggans funktion.....</i>	<i>10</i>
<i>Pulsviddsmodulering i H-bryggan.....</i>	<i>11</i>
<i>Val av transistorer.....</i>	<i>11</i>
<i>Simuleringar.....</i>	<i>12</i>
<i>Drivkretsar.....</i>	<i>13</i>
<i>Test av H-brygga och motor.....</i>	<i>15</i>
<i>Induktans i serie med motorn.....</i>	<i>16</i>
ENERGIFÖRSÖRJNING.....	17
MOTORSTYRKORTET.....	17
<i>Motorstyrkortets anslutningar.....</i>	<i>18</i>
PC OCH ÖVRIG HÅRDVARA.....	18
<b>4 PROGRAMMERING AV MIKROKONTROLLRAR.....</b>	<b>19</b>
HASTIGHETSREGLERINGEN.....	19
POSITIONSRÄKNING.....	20
KOMMUNIKATIONEN VIA RS232.....	20
<i>Hastighetskommandon.....</i>	<i>20</i>
<i>Avläsning av data.....</i>	<i>21</i>
KOMMUNIKATIONEN VIA I2C.....	21
<b>5 TEST OCH RESULTAT.....</b>	<b>22</b>
TEST AV REGLERINGEN.....	22
<i>Test av stegsvaret.....</i>	<i>22</i>
<b>6 DISKUSSION.....</b>	<b>25</b>
<b>REFERENSER.....</b>	<b>26</b>
<b>BILAGOR.....</b>	<b>27</b>
<i>Bilaga 1, kretsschema och layout för motorstyrkortet.....</i>	<i>28</i>
<i>Bilaga 2, flödesscheman.....</i>	<i>30</i>
<i>Bilaga 3, schematisk framställning av kommunikationsprotokollet.....</i>	<i>32</i>

# 1 Inledning

Denna rapport behandlar utvecklingen av styrelektronik till en mobil märobot. Arbetet har genomförts vid EISLAB och är ett examensarbete vid Luleå Tekniska Universitet.

## **Syfte**

Syftet med arbetet var att utveckla elektronik för motorstyrning samt energiförsörjningssystem till en robot. Denna skulle sedan kunna användas som en mobil märobot. På roboten ska det också finnas en mätplattform som kan vridas relativt horisontalplanet. På mätplattformen ska olika typer av givare kunna placeras, t.ex. för geometrisk mätning av rum och korridorer.

## **Mål**

Målet var att roboten ska kunna styras trådlöst via Internet med hög precision samt att mätdata från roboten ska kunna hämtas trådlöst via Internet.

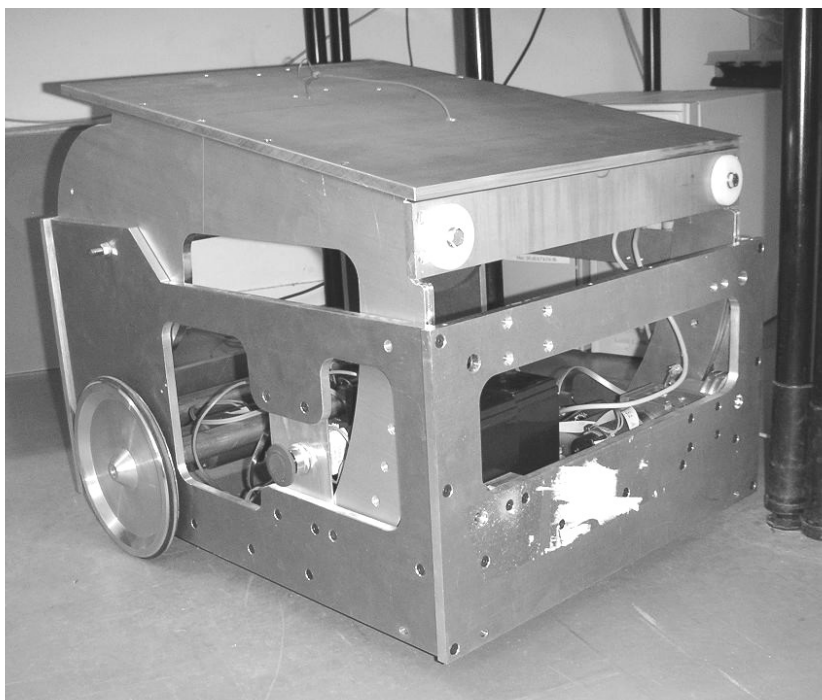
## **Avgränsningar**

Drivelektronik till motorerna skulle utvecklas. Till detta ska bland annat lämpliga mikrokontrollrar användas. Roboten behövde också ett energiförsörjningssystem som utnyttjar lämpligt standardbatteri. En befintlig PC ska användas för styrning och kommunikation via Internet. Befintlig programvara som utvecklats av Sven Rönnbäck ska användas för bland annat kommunikationen via Internet. Mätplattformen kommer att konstrueras av Kalevi Hyypä.

## 2 Bakgrund

### *Om roboten*

Den robot som används är en trehjulig mobil robot (Figur 1) som byggdes 1997 vid LTU. Den är mekaniskt mycket stabil och konstruerad för att ge god precision i dödräkningen. I roboten finns tre likströmsmotorer från Maxon motors. Två av motorerna har planetväxlar och inkrementella pulsgivare och används för framdrivning och styrning av roboten. Den tredje motorn kommer att användas för vridningen av mätplattformen. Den är försedd med planetväxel, snäckväxel och en inkrementell pulsgivare.



**Figur 1. Roboten, med ny plattform**

### *Motorerna*

Drivmotorerna är två stycken permanentmagnetsmotorer. De är kopplade till planetväxlar med nedväxlingsfaktorn  $91,12 : 1$ . Hjulen rör sig alltså ett varv för varje 91,12 motorvarv. Motorerna är också försedda med varsin inkrementell pulsgivare med upplösningen 500 cykler/varv. Sammantaget ger detta möjligheter till en mycket god noggrannhet i mätningen av hjulens hastighet och läge.

Även plattformsmotorn är en permanentmagnetsmotor. Dess inkrementella pulsgivare har en upplösning på 192 cykler/varv.

## Motorernas egenskaper

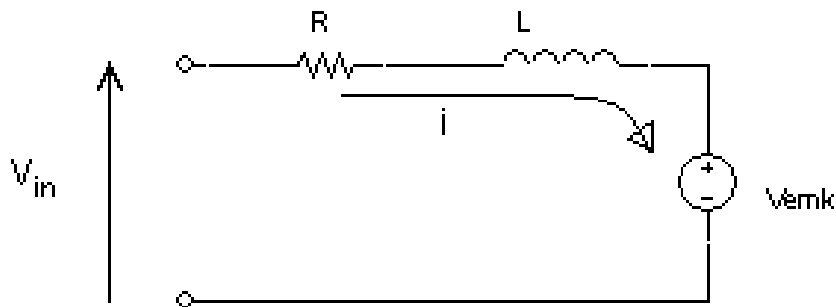
Då motorerna i roboten är permanentmagnetsmotorer ger de ett vridmoment som är proportionellt mot strömmen genom motorn. Sambandet mellan motorns matningsspänning och strömmen genom motorn beskrivs av ekvationen

$$v_{in} = L \left( \frac{di}{dt} \right) + R \cdot i + v_{emk} \quad (ekv.1)$$

där  $L$  och  $R$  är rotorlindningarnas induktans respektive resistans och  $v_{emk}$  är den spänning som induceras när motorn rör sig [1]. Den inducerade spänningen i sin tur är proportionell mot motorns vinkelhastighet, vilket ger

$$v_{in} = L \left( \frac{di}{dt} \right) + R \cdot i + K_e \cdot \omega \quad (ekv.2)$$

där  $K_e$ , motorns elektriska konstant, är en konstant som beror av motorns egenskaper. Sambandet kan även beskrivas som en elektrisk modell enligt Figur 2.



Figur 2. Elektrisk modell för motorströmmen

Utan extern belastning kommer motorns hastighet och acceleration att bero av rotorns tröghetsmoment och friktionen i motorn samt av vridmomentet (som är proportionellt mot strömmen) enligt

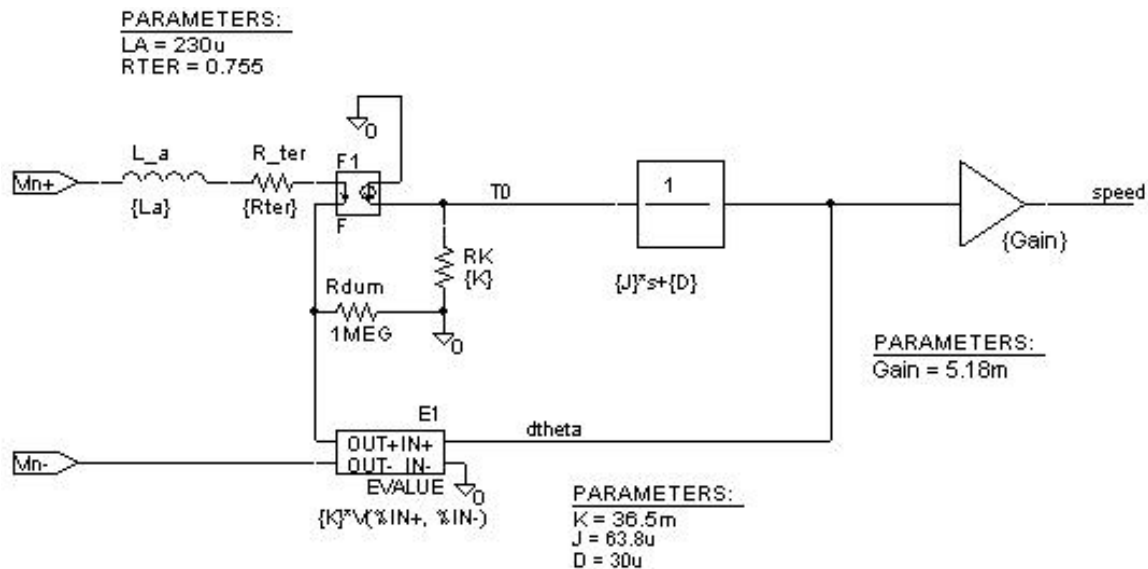
$$m(t) = K \cdot i(t) = J \left( \frac{d\omega}{dt} \right) + D \cdot \omega(t) \quad (ekv.3)$$

där  $m$  är vridmomentet,  $\omega$  är vinkelhastigheten,  $J$  är tröghetsmomentet och  $D$  är friktionsdämpningen i motorn [2]. Då SI-enheter används är konstanten  $K$  i *ekv.3* densamma som konstanten  $K_e$  i *ekv.2*. Genom Laplace-transformering av *ekv.3* kan vinkelhastigheten skrivas som

$$\Omega(s) = K \cdot I(s) \cdot \left( \frac{1}{sJ + D} \right). \quad (ekv.4)$$



Ekvationerna 2 och 4 beskriver den motormodell som används i simuleringarna (Figur 3). Värdena på  $L$ ,  $R$ ,  $D$ ,  $J$  och  $K$  får ur motorernas datablad.



Figur 3. Motormodellen som används i simuleringarna

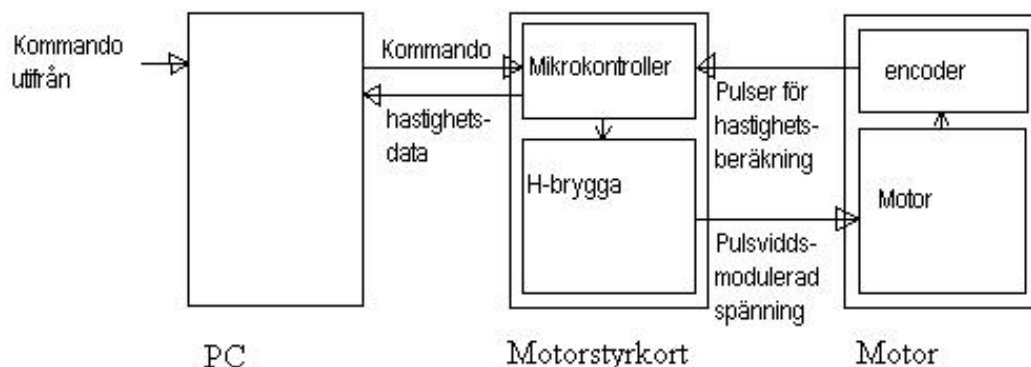
### Pulsviddsmodulering

En vanlig metod för att styra likströmsmotorers hastighet med en digital signal är att använda pulsviddsmodulering. Principen är att man låter den digitala signalen ange hur stor del av tiden som det ligger en spänning över motorn. En konstant periodtid används medan andelen av denna då spänningen är hög, dvs. pulstiden, varieras. Pulstiden brukar anges i procent av periodtiden. Detta är pulsviddsmoduleringens arbetscykel (på engelska "duty cycle"). Då motorn är en induktiv last kommer motorströmmen, om frekvensen för pulsviddsmoduleringen är tillräckligt hög, att endast variera lite grann under perioden. Den genomsnittliga motorströmmen beror av arbetscykeln, och motors moment är proportionellt mot den genomsnittliga motorströmmen.

### 3 Utveckling av hårdvara

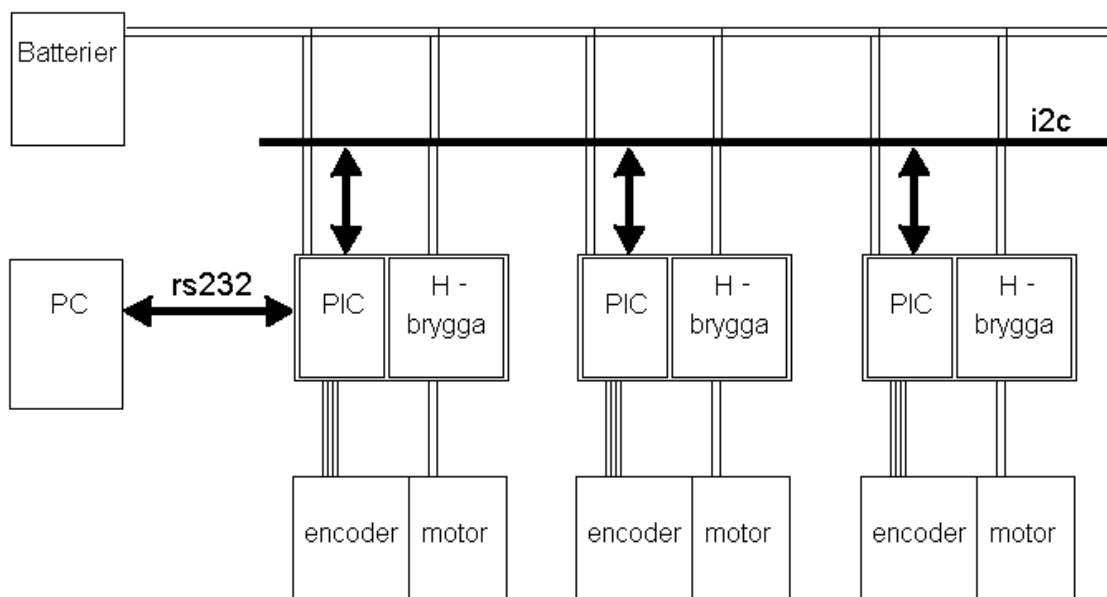
#### Översikt av styrelektroniken

Styrningen av motorerna sker genom att en PC skickar kommandon till en mikrokontroller på motorstyrkortet. Denna ska sedan kontrollera motorns hastighet. Detta sker genom pulsviddsmodulering och lämpliga elektronikkomponenter. Mikrokontrollern ska också kunna skicka data som beskriver motorns hastighet och antalet varv som motorn rört sig till PCn. Se Figur 4.



Figur 4. Översikt av styrelektroniken

Då roboten har tre motorer vars hastighet ska regleras så används tre motorstyrkort med en PIC mikrokontroller vardera. Endast en av mikrokontrollrarna kommunicerar med PCn via rs232. I övrigt skickas kommandon och data mellan mikrokontrollrarna via I<sup>2</sup>C-bussen. Ett blockschema över elektroniken kan ses nedan.



Figur 5. Blockschema för elektroniken

## **Val av mikrokontroller**

Den mikrokontroller som används i roboten är en PIC18F4431 från Microchip. Mikrokontrollarna i 18Fxx31-familjen är speciellt anpassade för motorstyrning, med inbyggda moduler för till exempel pulsviddsmodulering och hastighetsmätning. 18F4431 är den av dessa mikrokontrollrar som har mest minne och flest ben och valdes av denna anledning, då det i detta läge inte var klart hur mycket minne som skulle behövas samt för att göra det lättare att vidareutveckla stykortets funktioner.

Den valda mikrokontrollerns pulsviddsmodulering har åtta kanaler, med fyra av varandra oberoende pulsviddsgeneratorer. Detta gör att man i princip skulle kunna styra alla tre motorerna från en mikrokontroller. Däremot kan de inbyggda funktionerna för hastighetsmätning inte ta emot pulser från mer än en inkrementell pulsgivare. För att få återkoppling i hastighetsregleringen krävs givetvis att hastigheten kan mätas. Därför används tre mikrokontrollrar, en per motor.

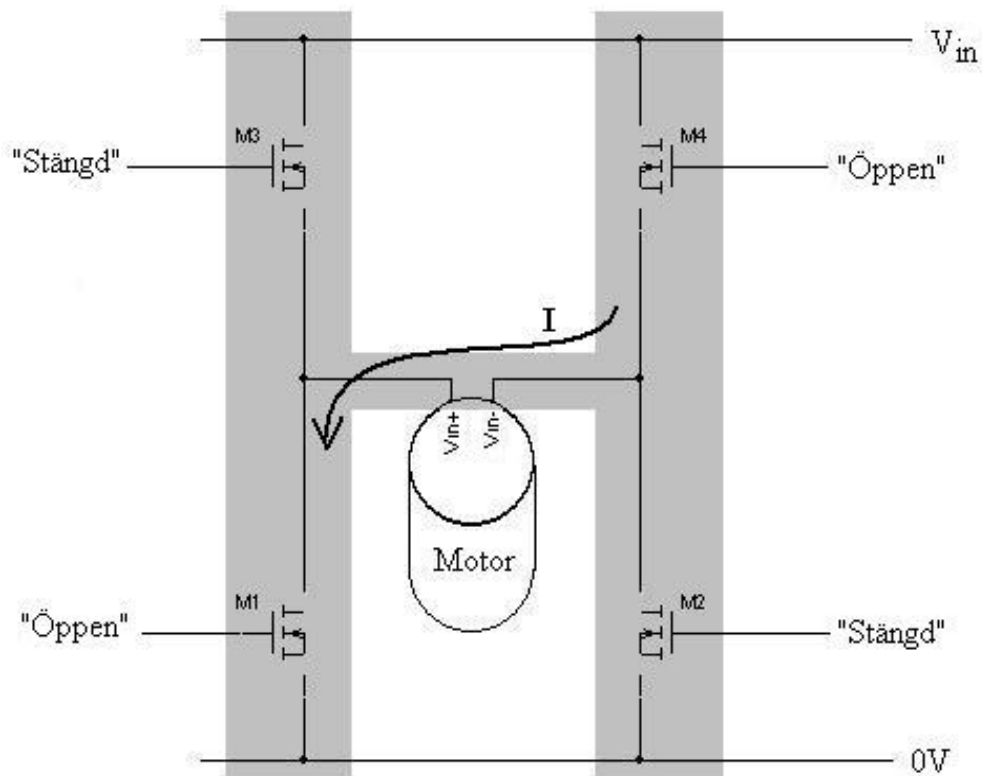
Mikrokontrollern kräver en matningsspänning på 5V (min 4.2V, max 5.5V) och ger också utsignaler (till exempel i pulsviddsmoduleringen) som är på 5V.

Mikrokontrollrarnas uppgift är att ta emot styrdata från PCn, reglera motorernas hastighet genom PWM-signaler till H-bryggorna samt att skicka data om motorernas hastighet och läge till PCn. För att underlätta kommunikationen med PCn använder mikrokontrollern en extern kristall med frekvensen 19,6608MHz. Denna frekvens ger en mer exakt baud rate vid rs232-kommunikationen än vad mikrokontrollerns interna oscillator kan generera. Mikrokontrollrarna måste också kunna kommunicera med varandra, vilket sker med hjälp av den inbyggda I<sup>2</sup>C-modulen.

## **H-brygga**

### **H-bryggans funktion**

Motorns hastighet och rotationsriktning ska kontrolleras genom pulsviddsmodulering. För detta ändamål används en H-brygga. Den består i grunden av fyra effekttransistorer placerade som i ett "H" kring motorn (Figur 6). Konstruktionen gör att man, genom att öppna två transistorer i taget, kan bestämma i vilken riktning som matningsspänningen ligger över motorn och därmed också i vilken riktning strömmen går. Om transistorerna M2 och M3 är öppna medan M1 och M4 är stängda ligger spänningen i positiv riktning över motorn, motorströmmen går i den riktningen och motorn roterar i sin positiva riktning. Om istället transistorerna M1 och M4 är öppna så ligger spänningen över motorn i motsatt riktning.



Figur 6. H-brygga

### Pulsviddsmodulering i H-bryggan

Pulsviddsmoduleringen i H-bryggan fungerar så att transistorerna M1 och M4 är öppna under en viss andel av tiden och M2 och M3 under den övriga tiden. När motorn ska stå stilla används således en arbetscykel på 50 %, vilket gör att den genomsnittliga strömmen genom motorn blir noll.

Något som är viktigt att tänka på i detta sammanhang är att se till att två transistorer som är direkt seriekopplade i H-bryggan (som transistorerna M1 och M3) aldrig är öppna samtidigt, då detta skulle leda till att höga strömmar skulle gå genom dessa transistorer. Förutom att det skulle ge effektförluster samt göra att strömmen inte alls går genom motorn så är det också troligt att det skulle leda till att transistorerna gick sönder pga. den höga strömmen. Därför används en kort dödtid, där gatespänningen till alla transistorerna hålls låg.

### Val av transistorer

H-bryggan består i grunden av 4st transistorer. Det första steget i konstruktionen av H-bryggan blev därför att välja ut lämpliga transistorer. De måste tåla lika stora strömmar som motorn för att inte begränsa möjligheterna att kontrollera motorns hastighet. Dessutom ska de ha en låg ledningsresistans  $R_{ds,on}$  för att minimera effektförlusterna i transistorerna.

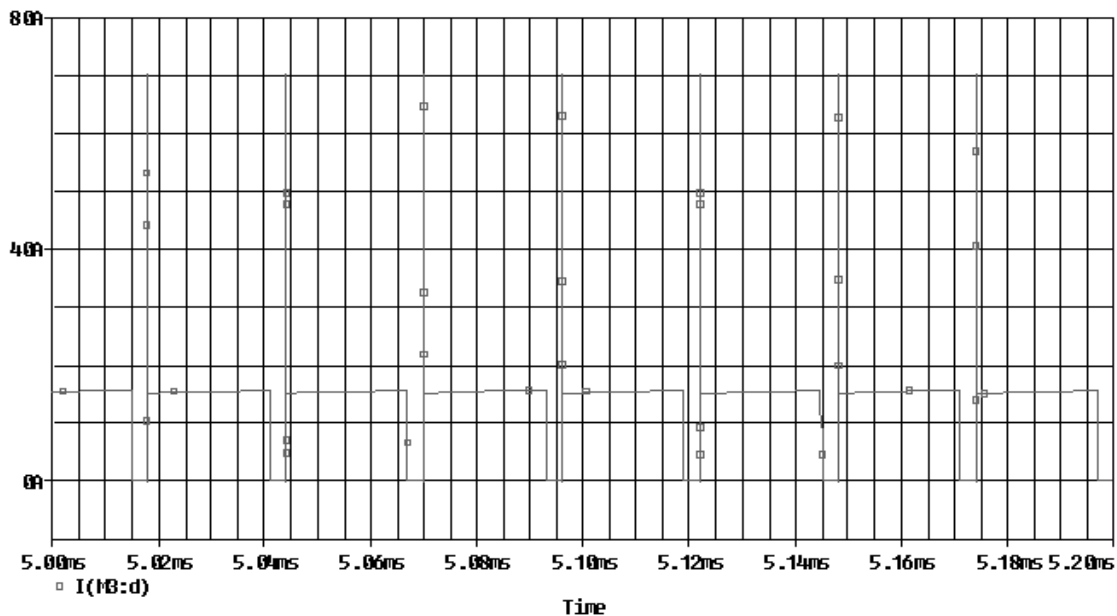
Transistorerna skall klara av:

- En kontinuerlig ström på 3.3A (maximal kontinuerlig ström för motorn),
- maxströmmar (motorns startström) på 23.8A, och
- en spänning mellan drain och source ( $V_{DSS}$ ) på 48V (dubbla matningsspänningen).

De transistorer som valdes ut är 4st N-kanals HEXFET-transistorer (IRF530N). Dessa har en ledningsresistans på  $0.11\Omega$ ,  $V_{DSS}$  på 100V och tål kontinuerliga strömmar på 17A och strömpulser på upp till 60A.

## Simuleringar

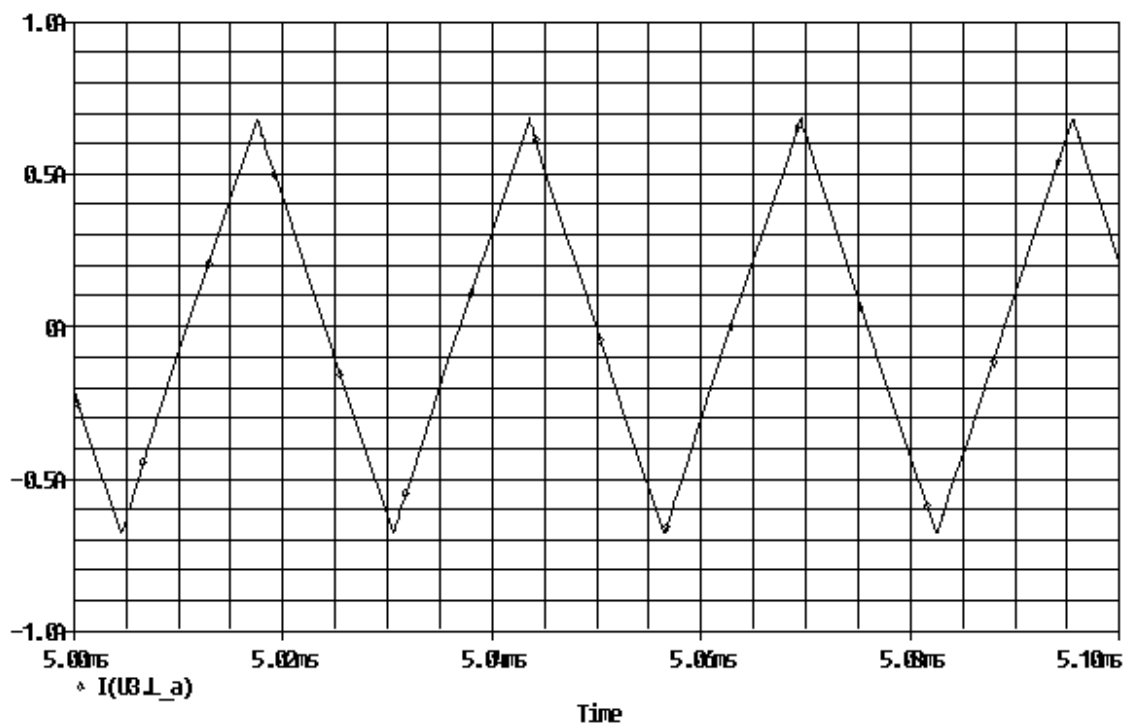
Simuleringar av en H-brygga med dessa transistorer gjordes. Kretsschemat ritades upp i OrCad Capture och simulerades med PSpice. En modell av motorn användes i simuleringen (Figur 3). Simuleringarna visade att det uppstår stora strömspikar genom transistorerna i tillslagsögonblicket (Figur 7), vilket kan ge upphov till ökade energiförluster samt störningar. Med en arbetscykel på 90 % går dessa strömspikar upp till 70A. Att strömmen genom transistorerna och motorn i denna simulering ligger så pass högt som på 14A beror på att simuleringen visar startförloppet. Det finns alltså ingen anledning att börja oroa sig över att strömmen är högre än motorns maximala kontinuerliga ström.



**Figur 7. Simulerad ström genom transistorn M1**

För att minska strömspikarna används externa frihjulsdioder vid transistorerna. Vid gaten på transistorerna placerades ett motstånd på  $33\Omega$  för att minska risken för självsvängningar på grund av resonans mellan ledningsbanornas induktans och transistorgatens kapacitans. Detta gjorde också att strömspikarna minskade ytterligare. I övrigt syns ingen skillnad i simuleringen av H-bryggan vid arbetscykeln 90 % jämfört med simuleringen utan resistorer men med dioder. Simulering med resistorer men utan dioder visar att strömspikarna inte minskas nämnvärt av att man använder resistorer om man inte samtidigt har dioderna.

Vid en simulering med en arbetscykel på 50 % kan man också se att motorns induktans inte är tillräckligt stor för att jämna ut strömmen genom motorn, det blir ett rippel på nästan 700mA (Figur 8). Dessa strömmar leder till effektförluster då motorn står stilla, samt när arbetscykeln ligger så pass nära 50 % att strömmen under en del av perioden går åt "fel" håll genom motorn. Då motorn inte ska dra mer än max 3.3A kontinuerligt så får 0.7A betraktas som en relativt stor rippelström. Detta kan lösas antingen genom att placera en extra induktans (spole) i serie med motorn eller genom att öka frekvensen i pulsviddsmoduleringen.



Figur 8. Simulerad motorström vid arbetscykeln 50 %

### Drivkretsar

Transistorerna ska styras med signalerna från mikrokontrollerns pulsviddsmodulering. För att transistorerna ska öppnas ordentligt krävs en större  $V_{gs}$  än de 5V som mikrokontrollern ger som ut signaler. Dessutom kan det bli ganska stora momentanströmmar i det ögonblick då transistorerna ska öppnas, eftersom gaten då behöver laddas upp. De övre transistorerna (M3 och M4) ska naturligtvis också ha en gatespänning som ligger över dess sourcespänning.

Det enklaste sättet att ordna omvandlingen från mikrokontrollerns PWM-signaler till lämpliga gatespänningar till transistorerna är att använda en färdig MOSFET-drivkrets som är speciellt avsedd för detta. För att H-bryggan ska fungera så bra som möjligt skall drivkretsen:

- kunna ge en lagom stor  $V_{gs}$  för att transistorerna ska öppnas ordentligt (ca 12V är lämpligt för de transistorer som används),
- ge den "övre" transistoren i H-bryggan en gatespänning refererad till transistorens source-spänning, dvs till spänningen på den sidan av motorn (ej till jord),

- ha låg kanalmatchningstid, dvs. den maximala skillnaden i tiden det tar från det att en ändring av insignalen ändrar utsignalen mellan de olika kanalerna i drivkretsen, samt
- ha en låg till/från-tid, för att inte utgöra en begränsning vid valet av frekvens för pulsviddsmoduleringen.

I detta fall är det också en fördel om drivkretsen inte har någon inbyggd, fast dödtid, då mikrokontrollern kan programmeras så att den genererar en dödtid. Detta ger en ökad flexibilitet. De två transistorerna som är på samma sida i H-bryggan ska drivas av samma drivkrets, eftersom den skillnad i tiden det tar för insignalen att ändra utsignalen annars kan bli onödigt stor. Detta skulle leda till att dödtiden måste vara längre och att frekvensen för pulsviddsmoduleringen begränsas. Alltså ska drivkretsen vara av typen "High/Low Side". Egenskaperna hos fyra drivkretsar som alla uppfyller dessa krav beskrivs i Tabell 1.

**Tabell 1. Alternativ på drivkretsar**

Drivkr.	Kapsel	spänning	strömförbr	utström, typ	Kanalmatchn	till/fråntid
IR2106	DIL8	+10/+20V	120 $\mu$ A typ	200/350mA	30ns max	220/200ns typ
IR2010S	SO16	+10/+20V	100 $\mu$ A typ	3,0/3,0A	15ns max	95/65ns typ
IR2181	DIL8	+10/+20V	120 $\mu$ A typ	1,9/2,3A	35ns max	180/220ns typ
LM5101	SO8	+9/+14V	250 $\mu$ A typ	1,6/1,8A	10ns max	25/25ns typ

MOSFET-drivkretsen LM5101 valdes, då den hade lägst till/från-tid, lägst kanalmatchningstid och lägst pris hos ELFA.

Till MOSFET-drivkretsen skall man använda en bootstrapkondensator som tillhandahåller den överspänning som krävs för att ladda upp gaten på den övre transistorn. Till många drivkretsar behövs även en bootstrapdiod, genom vilken strömmen som laddar upp bootstrapkondensatorn går, men den valda drivkretsen har en inbyggd diod. Bootstrapkondensatorn måste kunna tillhandahålla en laddning  $Q_{bs}$  enligt

$$Q_{bs} = 2Q_g + \frac{I_{qbs(max)}}{f} + Q_{ls} + \frac{I_{Cbs(leak)}}{f} \quad (ekv.5)$$

där  $Q_g$  är gateladdningen för den övre transistorn,  $I_{qbs}$  är "quiescent current" för drivkretsen,  $Q_{ls}$  är den nivåskiftesladdning som behövs för varje cykel,  $I_{Cbs(leak)}$  är läckströmmen genom kondensatorn och  $f$  är frekvensen för pulsviddsmoduleringen [3].  $I_{Cbs(leak)}$  är bara relevant om det är en elektrolytkondensator.

För att kondensatorn ska kunna ge ifrån sig denna laddning och ändå behålla spänningen så behöver laddningen i kondensatorn vara minst dubbelt så stor som  $Q_{bs}$ . Därför fås det lägsta värdet på kapacitansen av formeln

$$C > \frac{2Q_{bs}}{V_{cc} - V_f - V_{LS} - V_{min}} \quad (ekv.6)$$

där  $V_{cc}$  är drivkretsens matningsspänning,  $V_f$  är spänningsfallet över bootstrappedioden,  $V_{LS}$  är spänningsfallet över den nedre transistoren (då den är "öppen") och  $V_{min}$  är den lägsta gatespänningen man kan acceptera.

I det här fallet har vi

$$I_{qbs} = 0.2\text{mA} \text{ (enligt datablad [4])}$$

$$Q_g = (\text{transistorns gatekapacitans} * \text{spänningen}) = 640\text{pF} * 12\text{V} = 7.68\text{nC}$$

$$Q_{ts} = 5\text{nC} \text{ (enligt [3])},$$

$$V_{cc} \text{ är } 12\text{V}$$

$$V_f = 0,7\text{V}$$

$$V_{min} \text{ sätts till } 10\text{V}$$

$$V_{LS} = R_{ds,on} \text{ (för transistoren)} * I_{ds} \text{ (motorströmmen)} = 0.11\Omega * 3\text{A} = 0.33\text{V}.$$

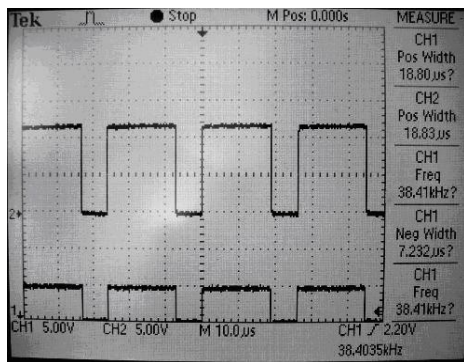
Beräknat på frekvensen 19,2kHz fås då för en keramisk kondensator att kapacitansen  $C > 63\text{nF}$ . Detta är dock det absolut lägsta värdet för kondensatorn som kan användas, och enligt en tumregel [3] bör man multiplicera denna kapacitans med 15, vilket ger  $C \approx 0.95\mu\text{F} \approx 1\mu\text{F}$ .

### Test av H-brygga och motor

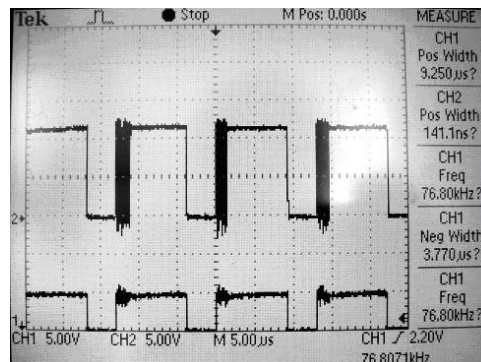
En H-brygga bestående av transistorer, dioder och resistorer kopplades upp på kopplingsdäck. Drivkretsarna monterades på ett litet, etsat kretskort och kopplades ihop med H-bryggan samt med en mikrokontroller.

Testerna visar att det går bra att styra motorns hastighet med denna H-brygga, så länge inte alltför höga frekvenser används. Vid höga frekvenser uppstår dock störningar som bland annat kan ge kortslutningar genom att två transistorer på samma sida i H-bryggan öppnas samtidigt. Störningarna uppstår vid omslaget mellan transistorerna för arbetscykler nära 0 eller 100 %. Se Figur 9 och Figur 10, som visar spänningen mätt vid transistorgaten och signalen från mikrokontrollern för två olika frekvenser. Då störningarna blir mycket värre vid höga frekvenser så bör inte frekvenser över 40kHz användas för pulsviddsmoduleringen.





**Figur 9. Pulsviddsmodulerad spänning vid arbetscykeln 80 %, frekvens 38.4kHz.**



**Figur 10. Pulsviddsmodulerad spänning vid arbetscykeln 80 %, frekvens 76.8kHz.**

Som tidigare nämnts går det en viss ström genom motorn även då arbetscykeln är 50 % och motorn står stilla pga. att spänningen över motorn varierar. Hur mycket ström det blir beror på motorns induktans och resistans samt på periodtiden i pulsviddsmoduleringen. Därför gjordes en mätning av strömförbrukningen vid några olika frekvenser. Strömmen mättes med en multimeter vid spänningskällan. I stort sett all den ström som förbrukas går genom motorn, då drivkretsarnas strömförbrukning är låg, även om den ökar något vid höga frekvenser. Mikrokontrollern hade i testuppkopplingen egen strömförsörjning.

Resultatet av mätningarna (Tabell 2) visar tydligt att en högre frekvens ger en lägre strömförbrukning.

**Tabell 2. Strömförbrukning vid stillastående motor för olika frekvenser**

Frekvens (kHz)	Strömförbrukning (A)
19,2	0.45
38.4	0.30
76.8	0.18

För att minimera effektförlusterna bör alltså en hög frekvens användas. Dessvärre har tidigare mätningar visat på stora störningar vid höga frekvenser. Den frekvens som valdes var därför 38.4 kHz. För att minska effektförlusterna vid arbetscykeln 50 % kan istället en induktans placeras i serie med motorn.

Drivmotorns strömförbrukning vid full hastighet och med en belastning som åstadkoms genom att hjulet bromsades med handen uppmättes till 1.6A.

### Induktans i serie med motorn

En induktans i serie med motorn måste tåla strömmar på ca 2A. Hos ELFA hittades två filterdrosslar för strömmar omkring 2A, den ena med induktansen 400μH och resistansen 180mΩ, den andra hade induktansen 1000μH och resistansen 280mΩ. Simuleringar visade att rippelströmmen vid arbetscykeln 50 %, som tidigare låg på 700mA, minskade till omkring 250mA med drosseln på 400μH och till 125mA med drosseln på 1000μH. I övrigt var skillnaderna i simuleringarna små, startströmmen blev något lägre med drossel än utan vilket gör att det kan ta lite längre tid för motorn att nå sin rätta hastighet. Drosseln med induktansen 1000μH valdes.

## **Energiförsörjning**

Då två av robotens motorer har en nominell spänning på 18V och den tredje har en nominell spänning på 24V ansågs det lämpligt att de batterier som används ger en spänning på 24V. Denna spänning är också tillräcklig för de instrument som är tänkta att användas på roboten. Till detta används två stycken 12V blyackumulatörer som seriekopplas. Tidigare nämnda tester har visat att drivmotorerna drar omkring 1.6A vardera vid full hastighet. Då det är två drivmotorer ger detta en strömförbrukning på 3.2A. Till detta kommer strömförbrukningen för plattformsmotorn, PC samt instrument som ska användas på roboten, men denna strömförbrukning bedöms vara relativt liten jämfört med drivmotorernas. För att möjliggöra ett par timmars användning av roboten på en uppladdning bör batterierna ändå ha en kapacitet på minst 7Ah. Då roboten har plats även för större batterier valdes ackumulatörer med kapaciteten 10Ah, vilket ger lite marginal.

Motorerna drivs direkt med 24V via H-bryggorna. För att få rätt spänning till mikrokontrollarna används en DC/DC-omvandlare (TMA2405S) som omvandlar 24V till 5V. Drivkretsarna till H-bryggan får sin ström från en spänningsregulator (TS78L12) som också den matas direkt med batterispänningen 24V.

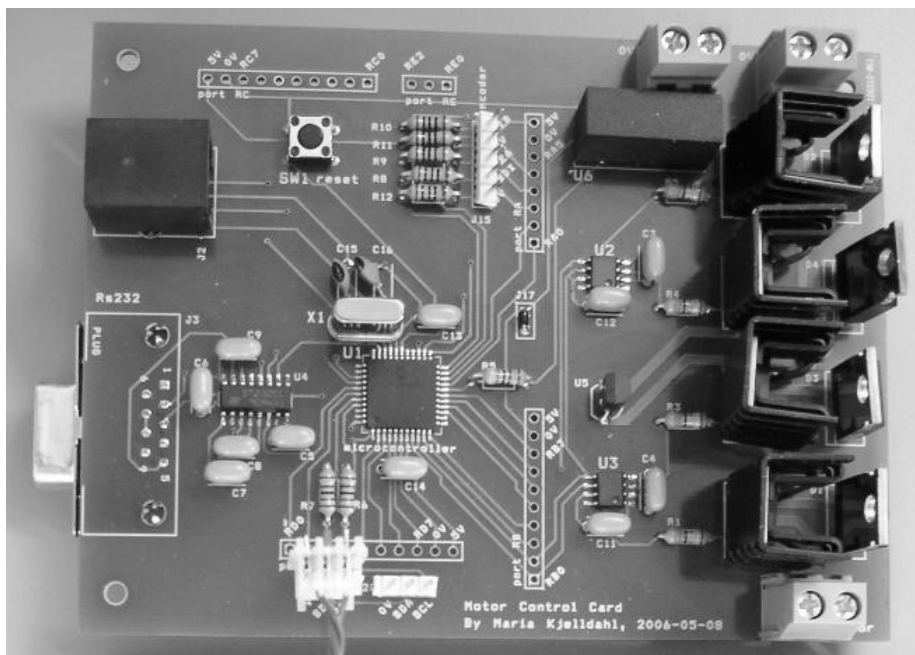
## **Motorstyrkortet**

Kretskortet konstruerades för att rymma H-bryggan samt mikrokontrollern. För att underlätta framtida vidareutvecklingar och användning av kortet i andra applikationer gjordes kortet för att styra *en* motor, vilket gör att roboten kräver tre motorstyrkort. Dessutom är samtliga oanvända ben på mikrokontrollern tillgängliga för inkoppling av annat via stiftlistor.

Själva layouten till kortet gjordes med hjälp av OrCAD Layout. Kretsschemat ritades först upp i OrCAD Capture, där en nätlista skapades. Nätlistan används som indata till Layout-programmet. Lämpliga footprints skapades också.

Då roboten är relativt stor så fanns det ingen anledning att försöka göra kortet väldigt litet. Därför används huvudsakligen hålmonterade komponenter, främst därför att sådana oftast finns hemma vid universitetet samt att de kan vara lättare att löda fast för dem som inte är så vana att löda.

Figur 11 visar det färdiga kretskortet med monterade komponenter. Komplet kretsschema samt layout-bild för motorstyrkortet kan ses i bilaga 1.



Figur 11. Det färdiga motorstyrkortet

### Motorstyrkortets anslutningar

Spänningsmatningen till kortet sker genom två kontakter (J22 och J23 i kretsschemat). Detta gör det möjligt att koppla in matningsspänning till mikrokontrollern samt programmera och testa denna innan motorspänningen kopplas in. Kontakten för motoranslutningarna är av samma typ som spänningsmatningens kontakter.

Motorstyrkortet har också ett rs232-interface med D-sub-kontakt för kommunikation mellan mikrokontrollern och en PC. För att kunna programmera och programmera om mikrokontrollern direkt på kortet finns även en modularkontakt (givetvis ansluten till lämpliga ben på mikrokontrollern). I<sup>2</sup>C-bussen är tillgänglig via två stiftlistor med vardera tre stift (för SDA, SCL samt jord). Det gör att man kan seriekoppla valfritt antal kort. I roboten används tre kort, ett per motor.

### PC och övrig hårdvara

På grund av tidsbrist har ännu ingen PC installerats i roboten. Tanken är att en PC104 ska användas, och att denna ska kommunicera trådlöst med andra datorer och skicka kommandon till mikrokontrollrarna. Någon form av sensorer för att känna av plattformens läge bör också läggas till. Det kan till exempel vara strömbrytare som stängs när plattformen är i sitt ändläge. Om dessa kopplas till mikrokontrollern som styr plattformen kan man då lägga till ett par rader i programmet som förhindrar att motorn försöker dra plattformen längre än till ändläget.

## 4 Programmering av mikrokontrollrar

För programmeringen av mikrokontrollrarna användes programvaran MPLAB IDE samt programmeraren/debuggern MPLAB ICD2 från Microchip. För att kunna programmera i programspråket C användes också en C-kompilator (PICC) som installerades som plug-in till MPLAB IDE. Nedan beskrivs i princip vad programmet i mikrokontrollrarna gör och vilka kommandon som används i kommunikationen med PC och mellan mikrokontrollrar. Flödesschema för programmen kan ses i bilaga 2.

### **Hastighetsregleringen**

Mikrokontrollrarnas främsta uppgift är att sköta regleringen av motorens hastighet. För att mäta den aktuella hastigheten används mikrokontrollerns inbyggda modul för hastighetsmätning. Den tar emot pulserna från den inkrementella pulsgivaren och sparar antalet pulser i ett 16 bitars register. Det finns också möjlighet att mäta tiden mellan två pulser, vilket skulle kunna användas för hastighetsmätning. Detta skulle dock ge problem vid mycket låga hastigheter, och därför valdes en annan metod. Hastigheten bestäms genom att positionsregistren läses av i en avbrottsrutin som exekveras 20 ggr/sekund. Därefter beräknas skillnaden mellan det nya avlästa värdet och det senaste tidigare avlästa värdet. Detta ger hastigheten i pulser per tjugondels sekund. För drivmotorerna skulle division med 50 ge hastigheten i varv per sekund.

I samma avbrottsrutin sker också själva regleringen. Den regleringsalgoritm som används är en diskret PI-regulator. Styrsignalen  $u(k)$  vid hastighetsmätning  $k$  kan beräknas genom formeln

$$u(k) = K_p e(k) + K_I \sum_{n=0}^{n=k} e(n) \quad (\text{ekv.7})$$

där  $e(k)$  är det aktuella felet och  $K_p$  och  $K_I$  är konstanter. Detta kan även skrivas på differentiell form som

$$u(k) - u(k-1) = K_p (e(k) - e(k-1)) + K_I e(k) \quad . \quad (\text{ekv.8})$$

För varje gång regleringen sker beräknas alltså en lämplig skillnad mellan den gamla styrsignalen och den nya. Styrsignalen begränsas också så att den alltid ligger inom det möjliga intervallet (pulsvidder som ger en arbetscykel mellan 0 och 100 %). Då den differentiella varianten av regleringen används undviks därmed också att den integrerande delen av regleringen går upp till alltför höga värden, något som annars kan ske om hastigheten är lägre än den önskade även vid maximal styrsignal.

Om inget kommando mottagits inom en viss tid genereras ett avbrott i mikrokontrollern som sätter hastigheten till noll. Detta för att förhindra att roboten fortsätter röra sig om rs232-kommunikationen upphör att fungera.

## **Positionräkning**

De mikrokontrollrar som styr drivmotorerna är programmerade till att räkna två pulser för varje cykel från den inkrementella pulsgivaren. Pulsgivaren har en upplösning på 500 cykler/varv. Antalet pulser sparas, som tidigare nämnts, i ett 16 bitars register i mikrokontrollern. Detta ger hur långt drivmotorn rört sig i upp till 65535 tusendels varv. För att kunna hålla reda på fler än 65 motorvarv initierades en variabel som håller reda på hur många gånger detta 16-bitars register ”slår runt” i ytterligare 16 bitar, vilket gör att man kan räkna upp till drygt 4 miljoner motorvarv. Detta kan användas för att med dödräkning bestämma hur roboten rört sig.

För den motor som styr plattformen är mikrokontrollern programmerad till att räkna fyra pulser/cykel, och pulsgivaren har en upplösning på 192 cykler/varv. Det gör att de 32 bitarna räcker till att räkna nästan 5,6 miljoner motorvarv.

## **Kommunikationen via rs232**

Den önskade hastigheten för motorn bestäms av en PC som skickar kommandon till mikrokontrollrarna. Då det är önskvärt att denna kommunikation kan ske via en av datorns serieportar så tar en av mikrokontrollrarna emot kommandon via rs232 och skickar sedan vidare kommandon till de andra två. Mikrokontrollern som reglerar plattformsmotorns hastighet valdes till att kommunicera med PCn. När ett kommando från PCn inkommer genereras ett avbrott för att läsa av inkommande data. Därefter avgörs till vilken mikrokontroller kommandot är riktat. Är det riktat till någon av de andra två så skickas kommandot vidare via I<sup>2</sup>C-bussen. En schematisk framställning av kommunikationen mellan PC och mikrokontrollrar finns i bilaga 3.

Mellan PC och mikrokontroller skickas kommandon och data i form av textsträngar, då detta gör det enkelt att testa programmet via Hyperterminalen på en vanlig dator. Alla kommandon avslutas med ”carriage return”. Det första tecknet i textsträngen anger vad man vill göra:

- m     nytt hastighetskommando till den motor som styrs av "mastern" (plattformsmotorn)
- r     nytt hastighetskommando till den högra drivmotorn
- l     nytt hastighetskommando till den vänstra drivmotorn
- h     läs av motorernas hastighet
- p     läs av nuvarande position

### **Hastighetskommandon**

Om det är ett nytt hastighetskommando som skickas så följs bokstaven m, l eller r av antingen ett plustecken eller ett minustecken (beroende på önskad riktning) samt av ett tresiffrigt tal som anger den önskade motorhastigheten. Då mikrokontrollrarna mäter hastigheten i pulser per tjugondels sekund så är det praktiskt om inkommande hastighetskommandon också anger en hastighet i pulser per tidsenhet.

Då de mikrokontrollrar som styr drivmotorerna programmerats till att räkna två pulser/cykel, blir antalet tusental pulser per sekund (eller pulser/ms) exakt detsamma som hastigheten i motorvarv/sekund. Därför valdes denna enhet för hastighetskommandona.

Även för plattformsmotorn låter vi hastighetskommandot ange antal pulser/ms. Vill man bestämma hastigheten i varv/sekund för denna motor så gäller att [antalet varv/sekund] = [(antalet pulser/millisecond)\*1000 / 768].

Till exempel så innebär alltså "m-010" att man vill att plattformsmotorn ska gå i 10 pulser/ms i negativ riktning och "r+055" att man vill att den högra drivmotorn går i 55 pulser/ms (vilket också är 55 varv/sekund) i positiv riktning. För att bekräfta att kommandot är mottaget skickas samma kommando tillbaka till PCn.

### **Avläsning av data**

Kommandot "h" returnerar en sträng som innehåller hastigheten för de olika motorerna. Hastigheten anges i pulser per tjugondels sekund, med 5 tecken, för varje motor. Detta är den hastighet som mäts och används i regleringen, så att ange hastighetsdata i denna enhet ger högsta möjliga upplösning. Hastigheten kan vara positiv eller negativ.

Kommandot "p" används för att läsa av positionsvärden och returnerar en teckensträng som innehåller ett 8-teckens hexadecimalt tal (dvs. ett 32-bitars tal) per motor. Det tal som anger positionen är alltid positivt.

### ***Kommunikationen via I<sup>2</sup>C***

Internt kommunicerar mikrokontrollrarna med varandra via I<sup>2</sup>C. Den mikrokontroller som har kontakt med PCn fungerar som "master", det är alltså i den som det bestäms när kommunikationen ska ske och vad som ska skickas. De andra mikrokontrollrarna är slavar och skickar bara data på kommando från mastern.

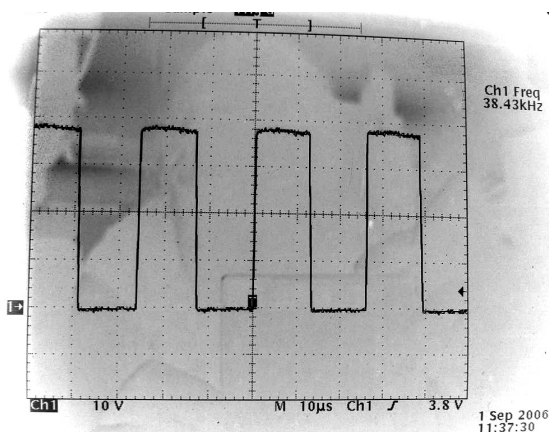
Kommunikationen börjar med att mastern skickar ut ett startkommando följt av en adress till en av de andra mikrokontrollrarna. Adressen avslutas med en bit som anger om mastern ska skicka ett kommando eller om den vill läsa data.

De kommandon som skickas till den adresserade slaven motsvarar de kommandon som skickas till mastern via rs232. Om mastern vill läsa data skickas först ett 'p' eller ett 'h' för att tala om vilken typ av data den vill läsa. Därefter adresseras mikrokontrollern på nytt, denna gång anger den sista biten i adressen att mastern vill läsa data. Hastighetsdata skickas som ett 16-bitars tal och positionsdata skickas som ett 32-bitars tal.

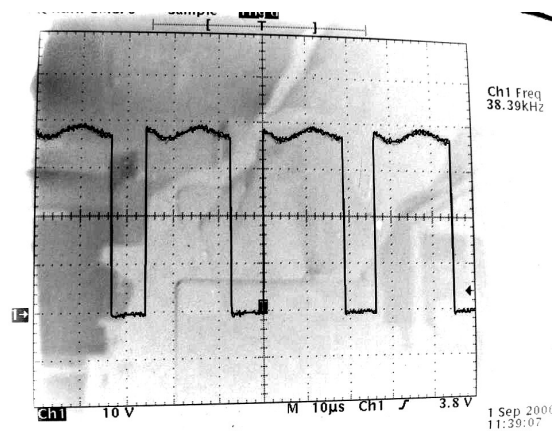
## 5 Test och resultat

Tester av motorstyrkortet visar att detta fungerar bra. DC/DC-omvandlaren ger en något för hög utspänning, mätning med multimeter visar att den hamnar på 5.5-5.7V. Det beror förmodligen på att strömförbrukningen är så pass låg. Detta verkar dock inte ha någon större betydelse, då mikrokontrollrarna fungerar som de ska ändå. Ett fel i kretskortets layout hittades, nämligen att D-subkontakten är spegelvänd. Enklaste sättet att komma runt detta problem är att helt enkelt montera kontakten på undersidan av kortet, men om nya kort ska beställas för senare tillämpningar är det förstås lämpligt att rätta till felet istället.

Med detta åtgärdat verkar kommunikationen mellan PC och mikrokontroller samt kommunikationen mellan mikrokontrollrarna fungera bra. Även pulsviddsmoduleringen fungerar som den ska, mätningar av spänningen vid gaten på transistorerna visar att signalerna är fina (Figur 12, Figur 13) och motorn rör sig när den ska göra det.



Figur 12. Spänning vid transistorgaten på transistor M3, arbetscykel 50 %



Figur 13. Spänning vid transistorgaten på transistor M3, motorhastighet 40 rps.

Robotens strömförbrukning med stillastående motorer mättes med en multimeter. Denna ström blev som väntat mycket mindre med drosslar i serie med motorerna. Utan drosslar drog motorerna och styrelektroniken sammanlagt 0.83A, med drosslar var det istället 0.11A. Strömmen mättes vid batterierna.

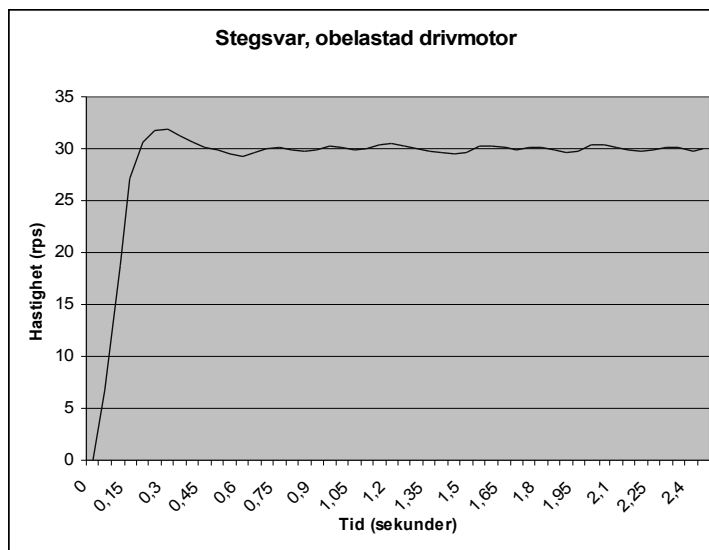
### Test av regleringen

Efter grovjusteringar av förstärkningen så verkar regleringen fungera ganska bra. Med hjälp av ett MatLab-program som skrivits av Håkan Fredriksson kan roboten styras med joystick från en vanlig PC. Hastigheten kan ändras snabbt utan att roboten går ryckigt.

### Test av stegsvaret

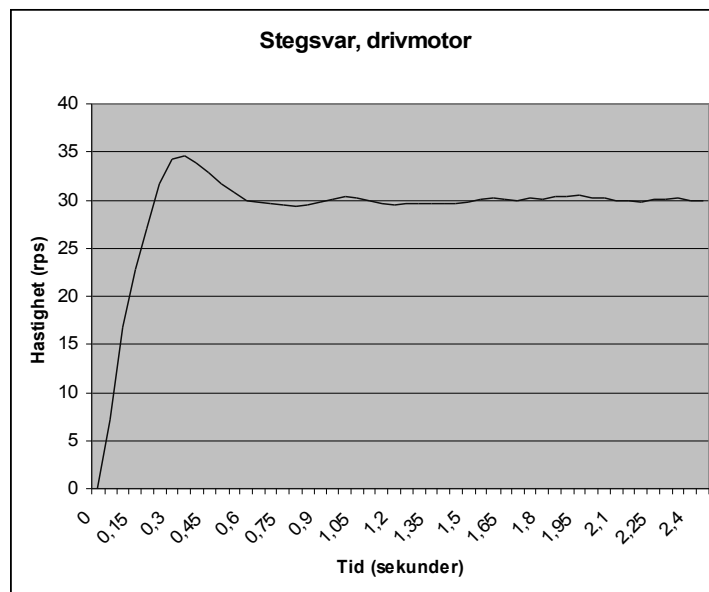
För att få en bild av hur regleringen fungerar testades hur hastigheten förändras vid en börvärdesförändring från 0 till 30 pulser/ms, vilket för drivmotorerna motsvarar motorhastigheten 30 varv/sekund. Testet genomfördes genom att mastern fick läsa av hastigheten för den motor som testades 20 ggr/sekund och skicka uppmätta hastigheter via rs232. Vid detta test styrdes motorerna från Hyperterminalen på en vanlig PC.

Stegsvaret för en av drivmotorerna testades först med roboten placerad så att hjulet inte rörde vid marken, det vill säga utan belastning. Resultatet kan ses i Figur 14. Det uppstår en liten översläng, dvs. hastigheten går upp till lite över det önskade värdet i början. Efter detta kan man se att hastigheten stabiliseras och ligger relativt konstant kring den önskade hastigheten.



**Figur 14. Stegsvvar för obelastad drivmotor**

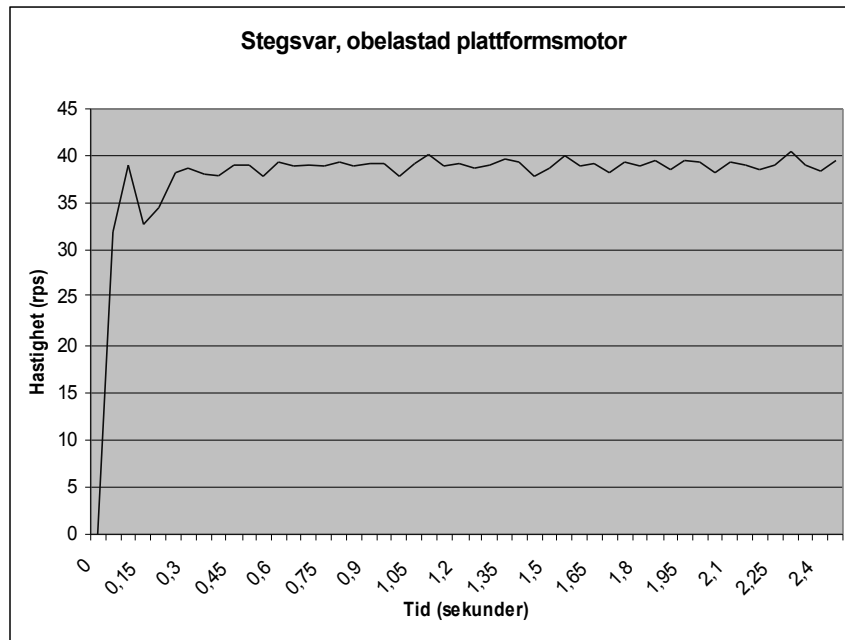
Då roboten placerades på golvet, så att motorn var belastad, blev överslängen lite större och det tog något längre tid att få motorn att gå i rätt hastighet (Figur 15).



**Figur 15. Stegsvvar för belastad drivmotor**



Plattformsmotorn testades obelastad, då linorna från motorn till plattformen ännu inte var på plats. Ett hastighetskommando på 30 pulser/ms bör här ge en hastighet på 39 varv/sekund. Resultatet ses i Figur 16. Regleringen verkar inte fungera lika bra som för drivmotorerna, hastigheten varierar mer efter det att motorn nått den önskade hastigheten. I det här fallet skiljer sig dock testförhållandena mycket från de förhållanden som kommer att råda när roboten är i drift, då plattformen kommer att utgöra en relativt stor belastning för motorn. Regleringen kommer att behöva justeras när allt är monterat. Därför har ingen tid lagts ner på att försöka optimera regleringen ännu.



Figur 16. Stegsvär för obelastad plattformsmotor

## 6 Diskussion

Elektronik för motorstyrning, i form av ett motorstyrkort med mikrokontroller och H-brygga, har utvecklats. Tester har visat att styrkortet fungerar. Hårdvaran för motorstyrning är därmed så gott som klar. Motorstyrkortet bör också gå bra att använda för att styra andra motorer. Eventuellt kan vissa komponenter i så fall behöva bytas ut, beroende på vilken spänning man vill använda till motorn.

Programmet i mikrokontrollrarna fungerar, både när det gäller kommunikation och reglering. Regleringen kan förmodligen förbättras, då parametrarna än så länge endast är grovt inställda till några värden som fick regleringen att fungera. Kanske bör regleringen göras till en PID-regulator för att minska överslängen i stegsvaret.

På grund av tidsbrist har ännu ingen PC installerats i roboten eller programmerats för styrningen av denna. Inte heller har den trådlösa kommunikationen mellan denna tilltänkta PC och andra datorer behandlats.

## Referenser

[1] Alciatore, David G. & Hstand, Michael B (2003); *Introduction to Mechatronics and Measurement Systems*, 2 ed. New York: The McGraw-Hill Companies, Inc. ISBN 0-07-119557-2

[2] Kenjo, T. & Nagamori, S (1985); *Permanent-Magnet and Brushless DC Motors*, Oxford: Oxford University Press, ISBN 0-19-856217-9

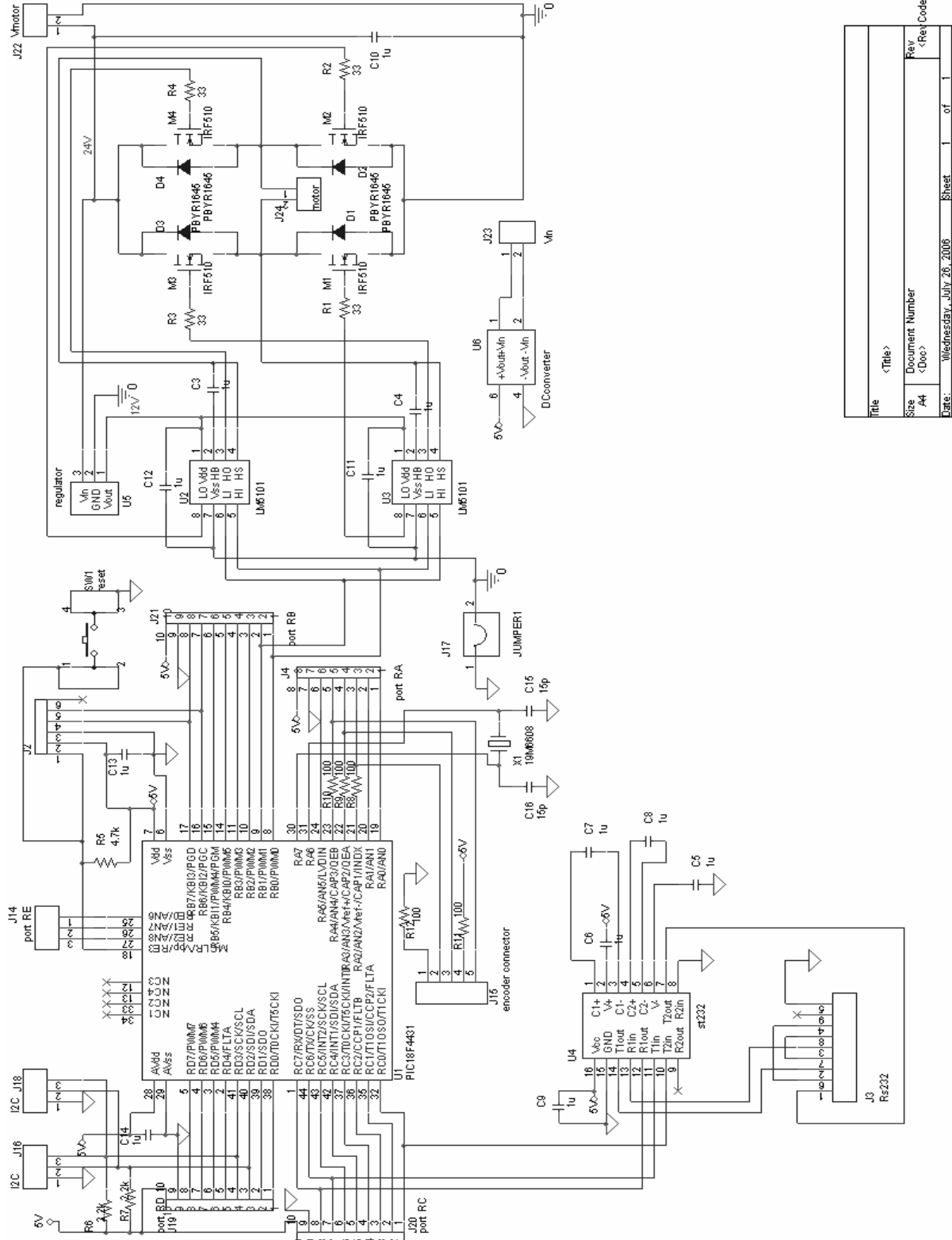
[3] Adams, Jonathan (2001): *Design Tip: Bootstrap Component Selection for Control IC's*. URL: <http://www.irf.com/technical-info/design/tp/dt98-2.pdf> (2006-01-26)

[4] Datablad för LM5101 (2005), URL: <http://www.elfa.se/pdf/73/732/07326784.pdf>

## Bilagor

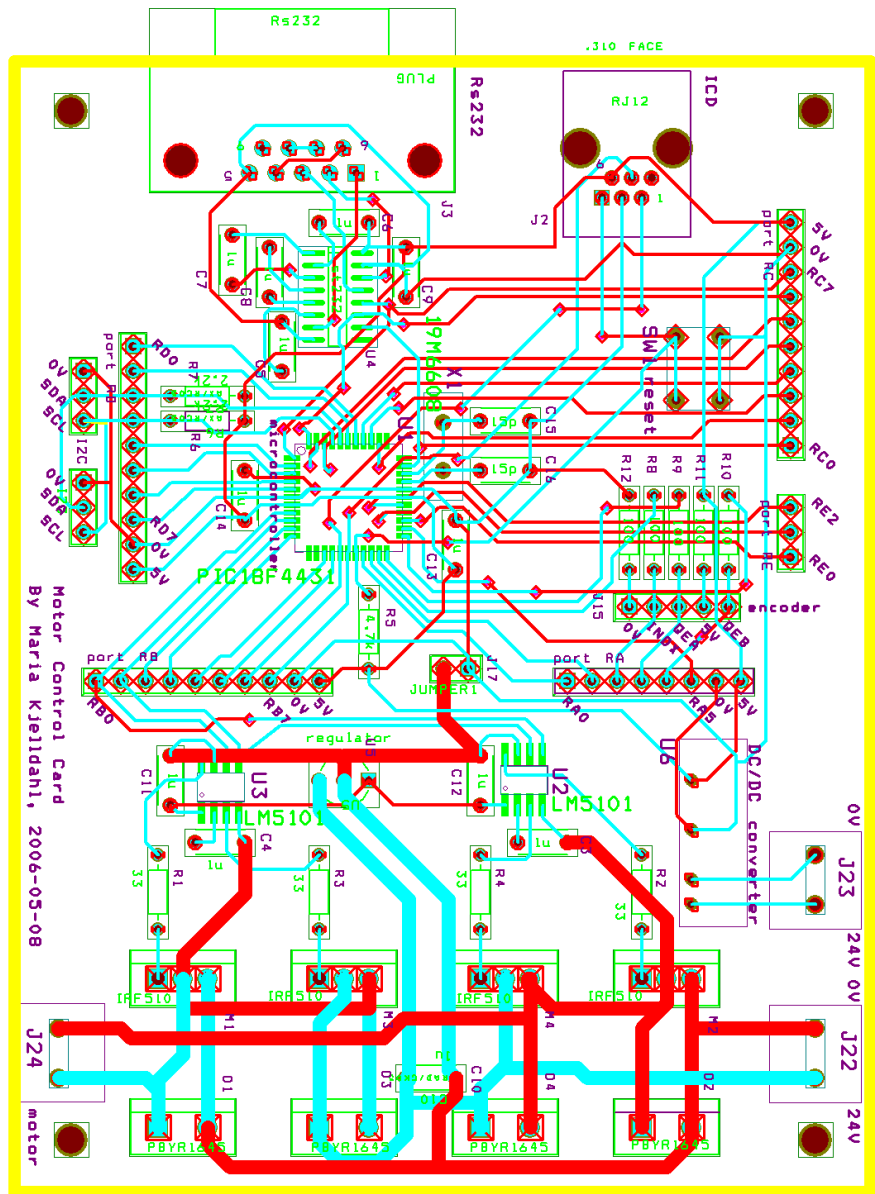
- Bilaga 1. Kretsschemat för motorstyrkortet (1 sida), Layout för motorstyrkortet (1 sida)
- Bilaga 2. Flödesscheman för programmen i mikrokontrollrarna (2 sidor)
- Bilaga 3. Schematisk framställning av kommunikationen mellan PC och mikrokontrollrar (1 sida)

# Bilaga 1, kretsschema och layout för motorstyrkortet



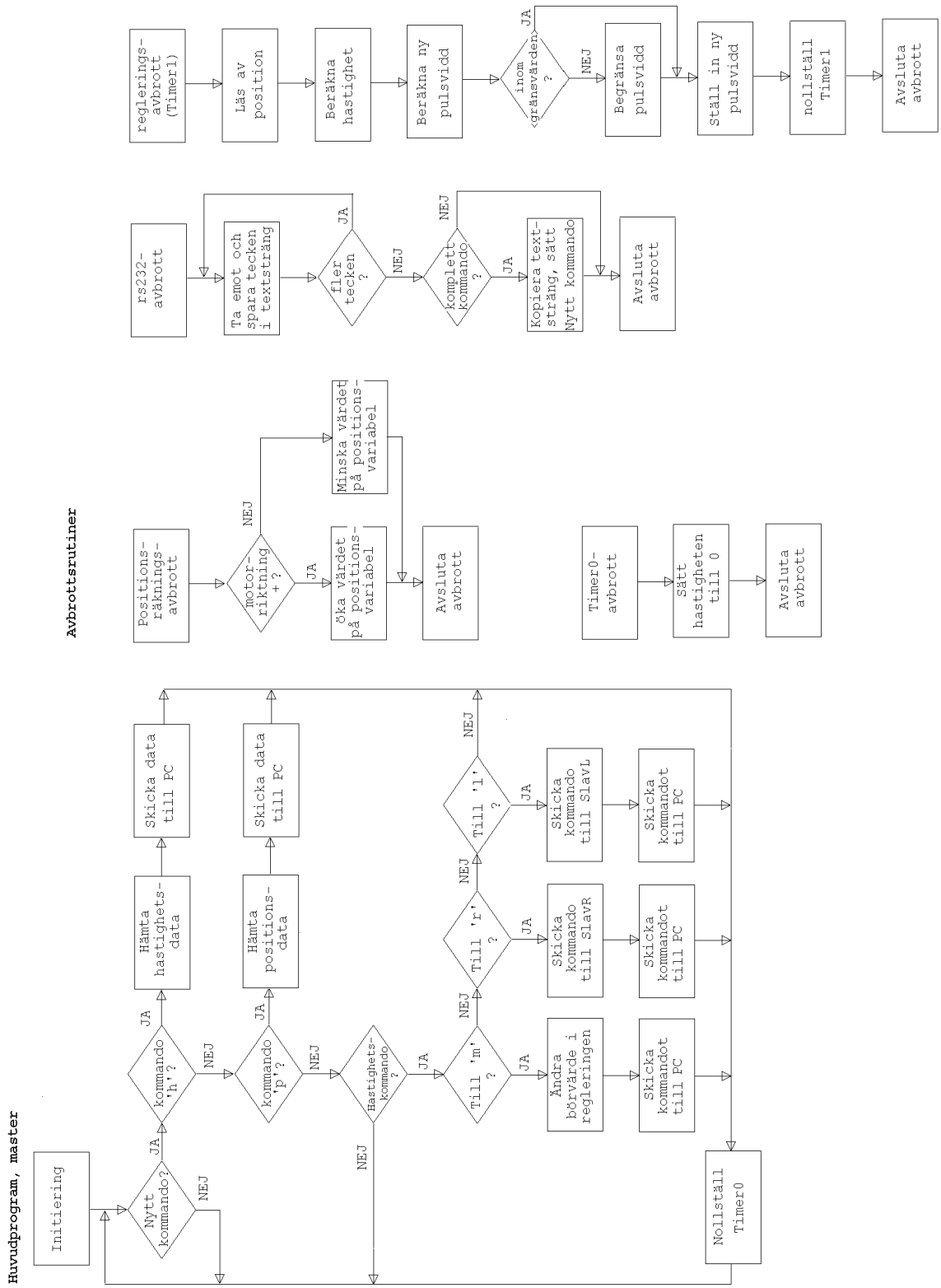
Kretsschema för motorstyrkortet

Title	<Title>
Doc Number	<Doc>
Rev	<Rev Code>
Date:	Wednesday, July 26, 2006
Sheet	1 of 1



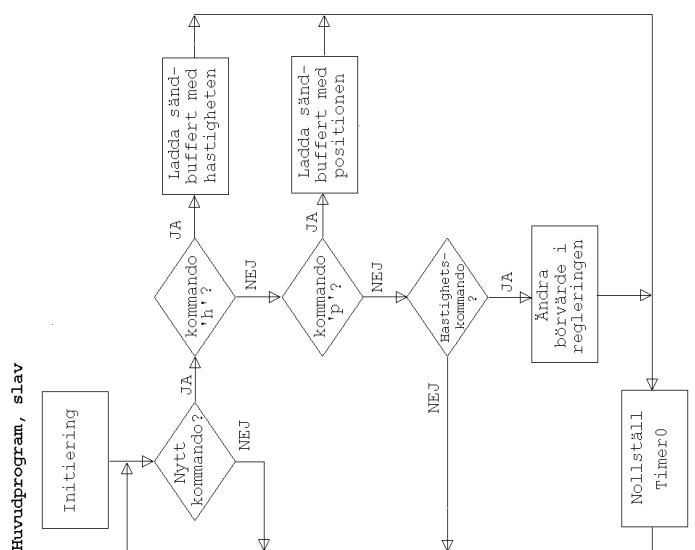
Layout för motorstyrkortet

## Bilaga 2, flödesscheman

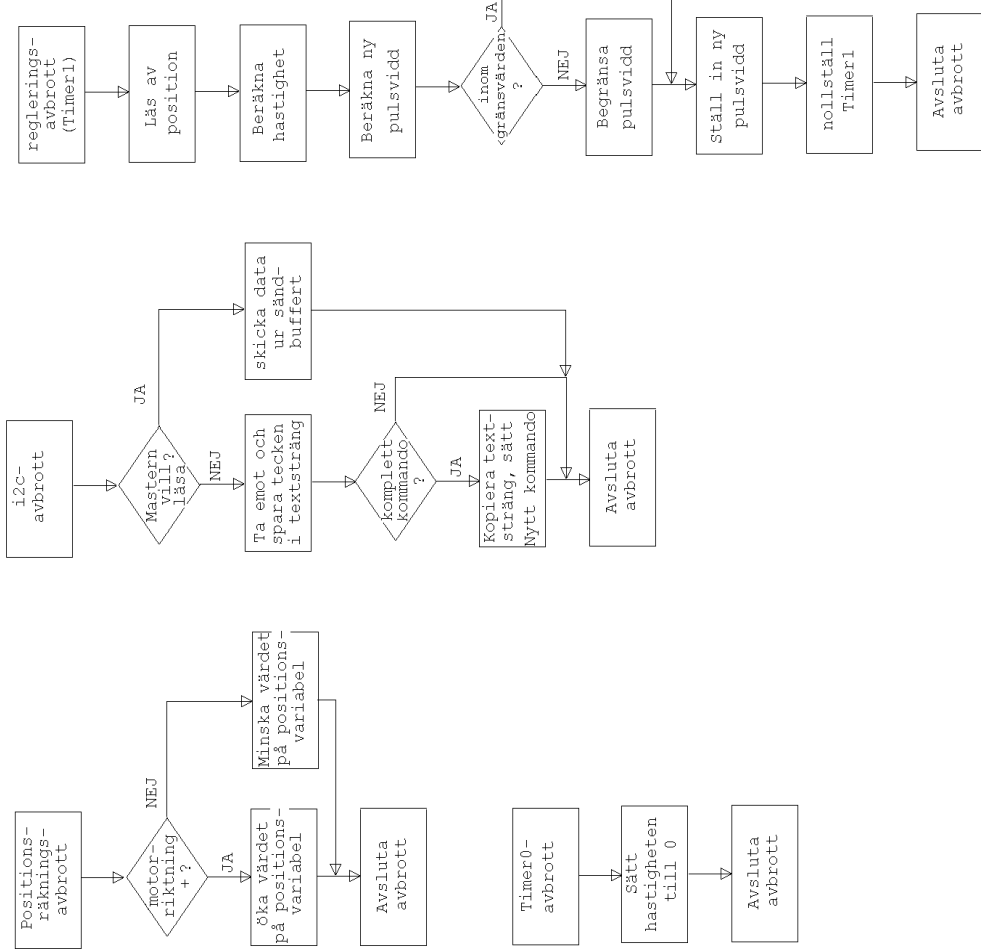


Flödesschema för "mastern"

Flödesschema för "slav"

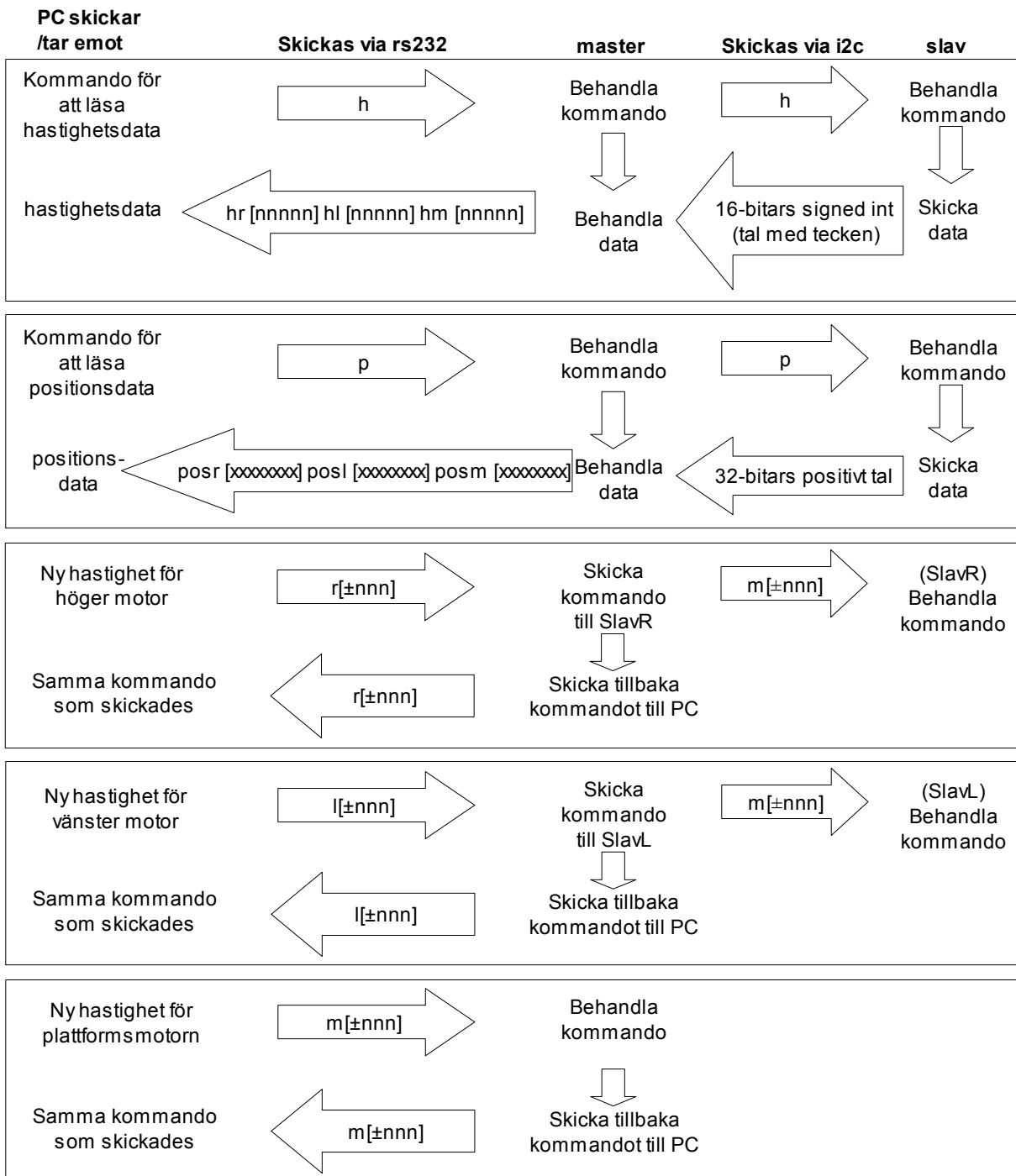


**Avbrottsrutiner**





### Bilaga 3, schematisk framställning av kommunikationsprotokollet



#### Kommentarer

Alla kommandon och allt som skickas via rs232 skickas som textsträngar som avslutas med "carriage return".  
 All kommunikation via i2c påbörjas med att mastern adresserar den slav som ska ta emot kommandot. Kommandon till båda slavarna skickas till en slav åt gången.

#### FÖRKLARINGAR:

[nnnnn] = en nolla eller ett minustecken följt av 4 siffror

[xxxxxxx] = åtta tecken (0-f) som utgör ett hexadecimalt tal

[±nnn] = ett plus eller ett minus följt av tre siffror