

Examensarbete, Högskolan på Åland, Utbildningsprogrammet för elektroteknik

ELEKTRONISK ANKARFUNKTION

Anders Lind, Jimmy Martin



16:2018

Datum för godkännande: 15.05.2018
Handledare: Kjell Dahl

EXAMENSARBETE

Högskolan på Åland

Utbildningsprogram:	Elektroteknik
Författare:	Anders Lind, Jimmy Martin
Arbetets namn:	Elektronisk ankarfunktion
Handledare:	Kjell Dahl
Uppdragsgivare:	

Abstrakt

Examensarbetet omfattar programmering och konstruktion av ett eget positioneringssystem för en fritidsbåt, ett så kallat elektroniskt ankare. Systemets avsikt är att underlätta fritidsfisket genom att ersätta det traditionella ankaret.

Målet är att i förhållande till befintliga produkter konstruera ett förmånligt system som automatiskt reglerar styrvinkel samt gaspådrag på en konventionell elektrisk utombordsmotor, så att en position kan bibehållas så exakt som möjligt.

Produkten testas och utvärderas med praktiska experiment. Förbättringsförslag tas sedan upp baserat på erfarenheter från dessa experiment.

Nyckelord (sökord)

Positioneringssystem, Elektroniskt ankare, Elektrisk utombordsmotor, Arduino, GPS, Stegmotor

Högskolans serienummer:	ISSN:	Språk:	Sidantal:
16:2018	1458-1531	Svenska	56 sidor

Inlämningsdatum:	Presentationsdatum:	Datum för godkännande:
07.05.2018	15.05.2018	15.05.2018

DEGREE THESIS

Åland University of Applied Sciences

Study program:	Electrical Engineering
Author:	Anders Lind, Jimmy Martin
Title:	Electronic anchor
Academic Supervisor:	Kjell Dahl
Technical Supervisor:	

Abstract
<p>This bachelor's thesis includes the design and programming of a self made position control system for a pleasure boat, also known as an electronic anchor.</p> <p>The intention is to facilitate recreational fishing by replacing a traditional anchor.</p> <p>The goal is to design an affordable system in comparison to existing products that controls steering angle and throttle on a conventional trolling motor, in such a way that a locked position can be maintained as accurately as possible.</p> <p>The product has been tested and evaluated with practical experiments. Enhancement proposals are taken up based on experiences from these experiments.</p>

Keywords
Positioning system, Electronic anchor, Trolling motor, Arduino, GPS, Stepper motor

Serial number:	ISSN:	Language:	Number of pages:
16:2018	1458-1531	Swedish	56 pages

Handed in:	Date of presentation:	Approved on:
07.05.2018	15.05.2018	15.05.2018

INNEHÅLLSFÖRTECKNING

1. INLEDNING	6
1.1 Bakgrund	6
1.2 Syfte samt mål	6
1.3 Avgränsningar	7
1.4 Metod	7
1.5 Relevanta förkortningar	7
2. HUVUDKOMPONENTER	8
2.1 Elektrisk utombordsmotor	8
2.2 Mikrokontroller	9
Tabell 1. Arduino Due specifikationer (Arduino, 2018)	10
2.3 Stegmotor samt drivmodul	11
Tabell 2. Stegmotorn jämförd med ett servosystem (Burriss, 2018) (AMCI, 2018) (Lackey, 2017)	11
Tabell 3. Servosystem jämfört med en stegmotor (Burriss, 2018) (AMCI, 2018) (Lackey, 2017)	12
2.4 GPS-modul	14
2.4.1 Grunden för GPS	15
2.5 PWM-modul	16
2.5.1 PWM	16
2.5.2 Planering & tillverkning	17
2.5.3 Komponentlista för PWM-modul	25
2.6 Elektronisk kompassmodul	25
3. SYSTEMUPPBYGGNAD	27
4. PROGRAMMERING	29
4.2 Avstängt läge	30
4.2 Manuellt läge	30
4.2.1 Initiering av stegmotorns position	31
4.2.2 Styrvinkel, gaspådrag samt fram/back	32
4.3 Ankarläge	33
4.3.1 Låsning av GPS-koordinater	34
4.3.2 Distans, GPS-bäring samt styrvinkelberäkning	34
4.3.3 Reglering av gaspådrag	36
4.3.4 Kompass	38
5. PRAKTISKA EXPERIMENT	39
5.1 Precision av GPS-modul	39
5.2 Torrövning	40
5.3 Utvärdering till sjöss	42

6. FÖRBÄTTRINGSFÖRSLAG	44
7. KOSTNADSKALKYL	45
Tabell 4. Kostnadskalkyl i Excel	46
8. SLUTSATS	47
KÄLLFÖRTECKNING	48
BILAGOR	51
Bilaga 1 - Ritning på nyckelströmställaren (Lambert, 2016).	51
Bilaga 2 - Stegmotorns datablad & vridmomentkurva (Stepperonline, 2018).	52

1. INLEDNING

1.1 Bakgrund

Bakgrunden till projektet är en fisketur sommaren 2017. Det var blåsigt väder och vi ankrade båten. Båten hade en elektrisk utombordsmotor och vi började filosofera om det inte skulle vara möjligt att båten självant skulle hållas på plats, med hjälp av motorn. Detta system visade sig redan existera som en färdig produkt, men som var relativt kostsam. Därför tänkte vi ifall det skulle gå att tillverka en egen version med den motor vi har och samtidigt även fördjupa våra kunskaper i programmering och problemlösning.

Vi kom att tänka på diverse problem som troligtvis skulle uppstå vid realisering av systemet, men vi beslöt oss för att fullfölja projektet eftersom det verkade som ett intressant och utmanande projekt.

Det som gjorde projektet intressant för oss var det faktum att det skulle innehålla både teoretiska och praktiska element samt även bli en användbar produkt.

1.2 Syfte samt mål

Examensarbetets syfte är att med hjälp av diverse komponenter och programvara skapa av en elektrisk utombordsmotor en produkt som får en fritidsbåt att automatiskt återgå till den position där ankarfunktionen aktiverats, då den rådande positionen avviker mer än felmarginalen. För att uppnå detta kommer gaspådraget och motorns styrvinkel att regleras simultant i harmoni.

Vår vision är även att ha en fjärrkontroll för manuell styrning av motorn då den elektroniska ankarfunktionen inte är aktiverad.

Vårt mål är att båten skall hållas relativt väl positionerad även vid mild sjögång. Motorn skall jämväl gå att smidigt fällas in i båten manuellt, då den inte är i användning. Tanken är även att behålla projektet ekonomiskt rimligt utan att offra alltför mycket på prestandan.

1.3 Avgränsningar

Anordningen kommer att konstrueras för en Terhi 405:a som tjänstgör som vår testbåt för denna prototyp. Detta betyder att det inte direkt kommer att gå att flytta över systemet till en annan båt utan att göra små förändringar i programmet samt konstruktionen.

Eftersom ett av projektets grundstenar är att prototypen skall hållas relativt förmånlig men välfungerande, så kommer vi särskilt noggrant ta hänsyn till komponenternas prisvärdhet för ändamålet. Detta betyder att vi inte kommer att införskaffa de allra värdefullaste komponenterna fastän de teoretiskt skulle ge oss en måttlig ökning i prestandan.

1.4 Metod

Examensarbetet inleddes med bollande av idéer om hur projektet löses på rationellaste vis. De bästa innovationerna analyserades och utifrån dessa gjordes en plan för hur systemet skall konstrueras. Denna plan fungerade som grund när komponenterna sedan införskaffades. Då komponenterna fanns till hands programmerades och testades de enskilt för det syfte som de är avsedda för. Då komponenterna och koden fungerade som vi tänkt oss kombinerades dessa till en helhet. Det konstruerade systemet utvärderades slutligen med praktiska experiment.

Dokumentation i form av text samt ritningar skedde parallellt med uppbyggnadsprocessen. Information hämtades från internet, böcker, egna erfarenheter samt av övriga kunniga personer.

1.5 Relevanta förkortningar

GPS	Global positioning system
SBAS	Satellite-based augmentation system
PWM	Pulse-width modulation
TTL	Transistor-transistor logic
UART	Universal asynchronous receiver-transmitter
MOSFET	Metal-oxide-semiconductor field-effect transistor
NMEA	National Marine Electronics Association

2. HUVUDKOMPONENTER

2.1 Elektrisk utombordsmotor

Hela projektet handlar om att modifiera en befintlig elektrisk utombordsmotor. Därmed behövde vi införskaffa oss en lämplig motor att utgå ifrån. En av oss hade en Minn Kota Endura C2 30 (se figur 1) som vi började modifiera. Denna 12 volts motor har en dragkraft på 13,6 kg och är avsedd för båtar med maxlängden 4,2 m samt maxvikten 500 kg. Båten som vi kommer använda oss av (Terhi 405) är inom ramarna för dessa rekommendationer.

Eftersom det befintliga hastighetsreglaget samt fastsättningen inte behövs demonterade vi dessa. Kvar blev endast själva motorn med riggrör och kablar. På grund av att denna utombordsmotor är avsedd att monteras på akterspegeln hamnade vi förlänga riggröret på grund av att vi kommer placera den i fören.



Figur 1. Minn Kota Endura C2 30 (Navimare, 2018)

Ett mellanstycke svarvades så att ett förlängningsrör i syrafast stål kunde anslutas (se figur 2). Delarna sammanfogades sedan med epoxylim.

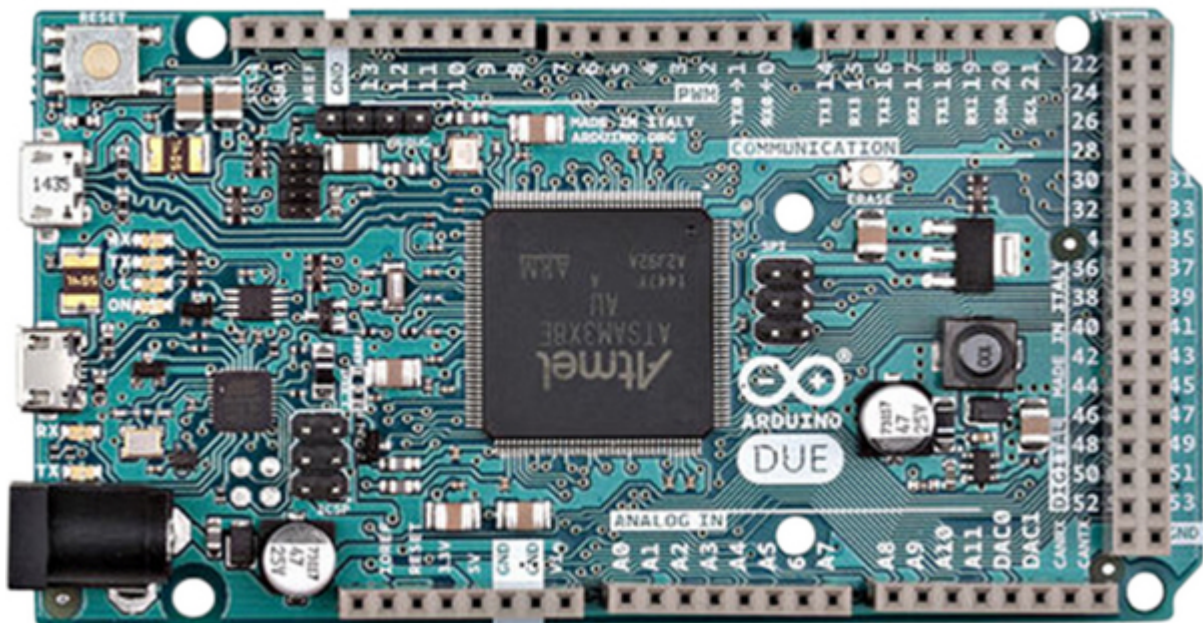


Figur 2. Förlängning av riggröret

2.2 Mikrokontroller

Grundstenen för att kunna styra, avläsa givare, göra beräkningar etcetera är mikrokontrollern, som i princip är en liten dator med processor, primärminne (ex. SRAM), programminne (ex. Flash) och I/O-enheter på samma integrerade krets (IC).

Vid valet av mikrokontroller satte vi stor vikt vid tidigare erfarenheter. Då vi tidigare gjort små projekt och laborerat med Arduino Uno samt Arduino Mega och varit hyfsat nöjda med Arduino IDE-mjukvaran och kodspråket som liknar till stor del C/C++. Därmed var det ett av de lättare besluten. Stora frågan i detta sammanhang var således vilken Arduino mikrokontroller vi skulle införskaffa. Då vårt projekt jobbar med en hel del data och beräkningar så var vi övertygade om att Unons och Megans 8-bitars processorer med 16 MHz klockfrekvens inte skulle vara ett alternativ för oss. Den enda Arduinons specifikationer som uppfyllde våra önskemål var Arduino Due (se figur 3) som har en 32-bitars Atmel SAM3X8E ARM Cortex-M3-processor med en 84 MHz klockfrekvens. (För fler specifikationer se tabell 1.)



Figur 3. Arduino Due (Distrelec Group Inc., 2018)

Tabell 1. Arduino Due specifikationer (Arduino, 2018)

Mikrokontroller	AT91SAM3X8E
Driftspänning	3,3 VDC
Rekommenderad ingångsspänning	7-12 V
Antal digitala I/O-stift	54 (varav 12 har möjlighet för PWM)
Antal analoga ingångar	12

Antal analoga utgångar	2 (DAC)
Antal UART	4
Antal TWI (I ² C)	2
Maximal ström alla I/O-stift	130 mA
Flashminne	512 kB
SRAM	96 kB
Klockfrekvens	84 MHz
Längd	101,52 mm
Bredd	53,3 mm
Vikt	36 g

Självklart fanns det också “nackdelar” med Due. Till skillnad från exempelvis Uno är Dues driftspänning 3,3 V medan Unos är 5,0 V. Detta innebär att då Uno är väldigt populär så är en stor del av kringutrustningen gjord för 5,0 V och på grund av detta finns det inte samma breda urval för Due. Dock var detta inte ett större problem för oss, men ett ting som vi hamnade ta i beaktande vid beställning av komponenter. En annan “nackdel” är att då största delen av modellerna använder sig av en 8-bits AVR-processor så är en stor del av de färdigt gjorda biblioteken inte direkt kompatibla med Dues 32-bitars ARM-processor, så det krävs modifieringar i koden, vilket gör programmeringen mer tidskrävande.

2.3 Stegmotor samt drivmodul

En av de viktigaste aspekterna i vårt system är förmågan att kunna justera elutombordarens styrvinkel elektroniskt. Efter en tids funderande kom vi fram till att en elmotor med remöverföring skulle vara en smidig lösning. Däremot var det inte lika självklart vilken typ av elmotor vi skulle använda, det stod mellan stegmotor och servomotor. För att determinera vilken av dessa som lämpar sig bäst för vårt ändamål, togs i beaktande fördelar och nackdelar för respektive motortyp (se tabell 2 samt 3).

Tabell 2. Stegmotorn jämförd med ett servosystem (Burris, 2018) (AMCI, 2018) (Lackey, 2017)

<i>Fördelar</i>	<i>Nackdelar</i>
-----------------	------------------

<ul style="list-style-type: none"> • Högt hållmoment 	<ul style="list-style-type: none"> • Lägre moment vid höga hastigheter
<ul style="list-style-type: none"> • Starkare moment vid låga hastigheter 	<ul style="list-style-type: none"> • Sämre på att accelerera en last
<ul style="list-style-type: none"> • Lägre kostnad 	<ul style="list-style-type: none"> • Ingen återkoppling på positionen
<ul style="list-style-type: none"> • Kan ej skadas vid mekanisk överbelastning 	<ul style="list-style-type: none"> • Högre värmebildning

Tabell 3. Servosystem jämfört med en stegmotor (Burris, 2018) (AMCI, 2018) (Lackey, 2017)

<i>Fördelar</i>	<i>Nackdelar</i>
<ul style="list-style-type: none"> • Bibehåller högt vridmoment vid höga hastigheter 	<ul style="list-style-type: none"> • Högre kostnad
<ul style="list-style-type: none"> • Bättre energieffektivitet 	<ul style="list-style-type: none"> • Ofta lägre vridmoment vid låga hastigheter
<ul style="list-style-type: none"> • Lägre värmebildning 	<ul style="list-style-type: none"> • Mer komplext styrsystem
<ul style="list-style-type: none"> • Återkoppling på positionen 	

Valet blev en stegmotor, främst på grund av prisskillnaden samt det faktum att hållmoment och vridmoment vid låga hastigheter är viktigt i vårt fall. Stegmotorns specifikationer kan ses i tabell 4. Återkoppling på positionen är normalt sett inte nödvändigt för stegmotorer, så länge de används inom specifikationerna (Lackey, 2017).

Specifikationerna för stegmotorn vi valde kan ses i tabell 4. Drivmodulen som vi införskaffade till stegmotorn heter TB6600. Dess uppgift är att styra huvudströmmen till stegmotorn med hjälp av styripulser från mikrokontrollern.

Stegmotorns har stegvinkeln 0.9° vilket betyder att ett varv är uppdelat i 400 steg. Denna noggrannhet räcker mer än väl för vårt ändamål. Vi utförde en mätning på hur stort moment som krävs för att få riggröret i rörelse. Det krävdes 1,25 kg med ett remhjul med radien 2,5 cm. Då kunde vi räkna ut vridmomentet enligt följande: $1,25 \text{ kg} * 9,8 \text{ m/s}^2 * 0,0225 \text{ m} \approx 0,28 \text{ Nm}$. Sedan kom vi fram till att

hastigheten 5 sekunder/varv, med andra ord 12 varv/minut är lämplig genom att testköra stegmotorn.

Vridmomentkurvan för vår motor är definierad endast efter 400 varv/minut, vilket är ett problem eftersom vi inte vet vad det riktiga vridmomentet är vid 12 varv/minut. Detta betyder att vi var tvungna att utgå ifrån att det är det samma som för 400 varv/minut, alltså 0,67 Nm. Med ett lika stort remhjul på stegmotorn skulle momentet då vara 0,67 Nm medan det beräknade vridmomentet för att få axeln i rörelse är 0,28 Nm. Vi noterade att vridmomentet testades på land och i praktiken krävs ett högre vridmoment. En marginal på cirka 2,4 gånger minimikravet verkade oberoende tillräcklig, så vi skaffade ett lika stort remhjul för stegmotorn.

Då stegmotorn var installerad prövades dess vridmoment i praktiken. Hållmomentet var tillräckligt kraftigt. Däremot visade det sig att det krävdes en relativt svag yttre störning på motorns riggrör då stegmotorn var i rörelse för att stegmotorn skulle tappa steg. Det enklaste sättet att lösa detta var att byta ut stegmotorns remhjul till ett mindre, vilket ledde till att vi skrev ut ett tre gånger nedskalad remhjul med hjälp av en 3d-skrivare. Eftersom nedväxlingen blev 1:3 är det nya hållmomentet $3 * 0,9 \text{ Nm} = 2,7 \text{ Nm}$ och rörelsemomentet $3 * 0,67 \text{ Nm} = 2 \text{ Nm}$. Detta visade sig i praktiska test vara tillräckligt, den enda nackdelen är att nedväxlingen måste tas i beaktande i programmeringen.

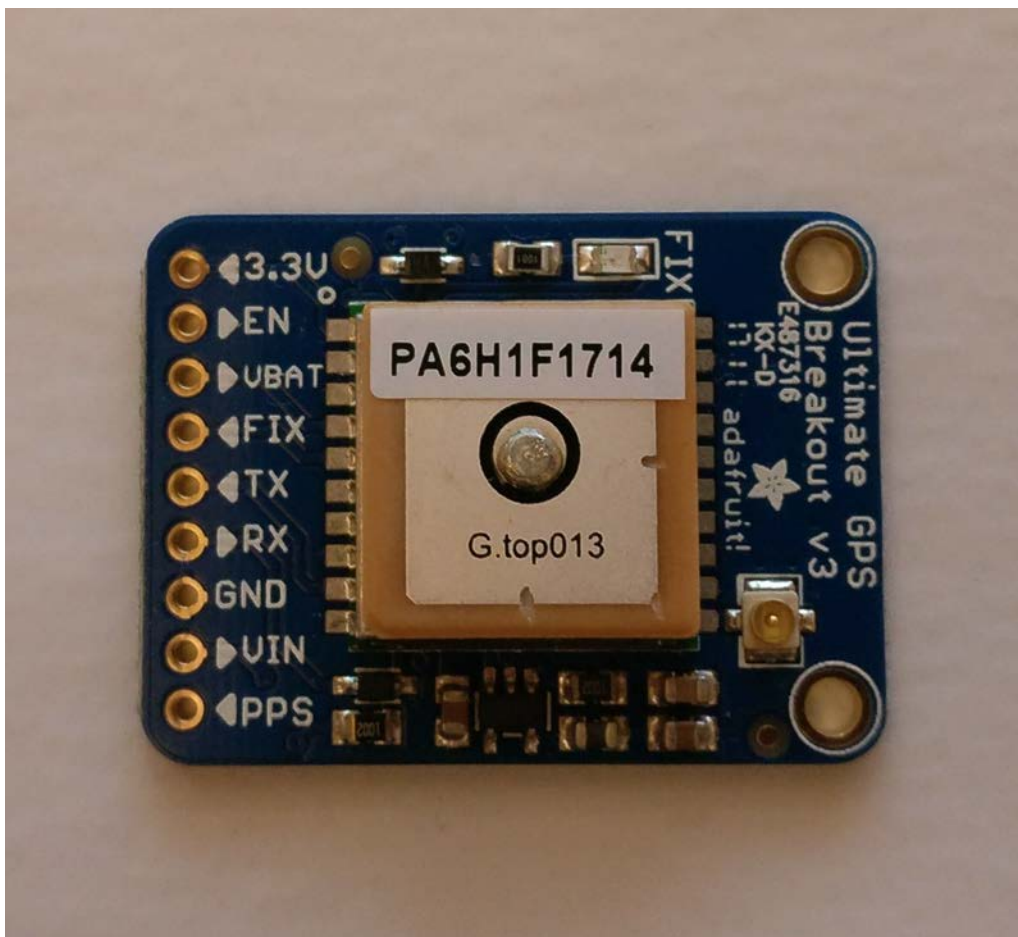
Tabell 4. Stegmotorns specifikationer. (Stepperonline, 2018)

Tillverkningsnummer	23HM20-0384S
Stegvinkel	0.9°
Stegnoggrannhet	5%
Hållmoment	0.9 Nm
Märkström/fas	0.38 A
Fasmotstånd	32 Ω
Spänning	12.1 V
Induktans	40 mH ± 20% (1 kHz)
Vikt	600 g

2.4 GPS-modul

Då själva konstruktionens grundfunktion är att kunna positionsreglera en fritidsbåt så behövs det uppenbarligen något som kan uppmäta den momentana positionen. För detta fordras en relativt exakt GPS-mottagare som mäter nuvarande position. På grund av detta var en av våra största frågeställningar i början att vilken GPS-modul ger den precision som krävs för detta projekt och inte ändrar allt för mycket koordinater då den är stationär, då det uppenbart inte är möjligt att styra en position med något som kastar med flera meter.

Efter att vi gått ner till grund och botten med en hel drös av moduler genom att läsa manualer och diverse forum fanns det en och annan nöt att knäcka, men när det väl kom till kritan valde vi Adafruit Ultimate GPS (se figur 4) eftersom vi inte vill satsa på fel modul.



Figur 4. Adafruit Ultimate GPS

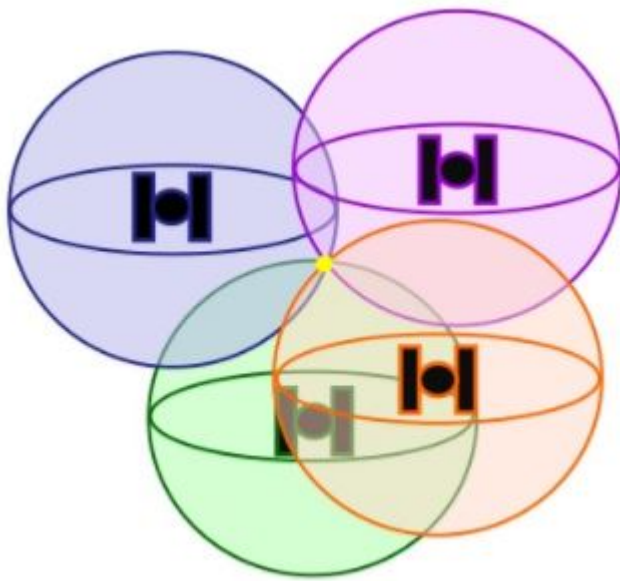
Anledningarna till att vi valde just denna modul som är baserad på MTK3339 chipsetet är åtskilliga, men de väsentliga var att det är 3,0-5,5 VDC spänningsförsörjning samt att den har

en positionsnoggrannhet på 3 m och med SBAS aktiverat 2,5 m. Uppdateringsfrekvensen kan dessutom ställas upp till 10 Hz vilket i vårt fall är mer än nog. Adafruit Ultimate GPS är även en väl använd och beprövad modul, vilket således innebär att det finns flera användningsexempel och fina färdiga bibliotek att använda sig av vid programmering. Som grädden på moset finns det en U-FL kontakt på kortet så att vi kan med en u.FL till SMA-adapter fästa en extern aktiv antenn till modulen för att få bättre mottagning.

Kommunikationen mellan GPS-modulen och mikrokontrollern sker med +3,3 V TTL nivå UART signaler. Datan som skickas från modulen till mikrokontrollern är i formatet NMEA 0183. (Adafruit, 2016) (GlobalTop Technology Inc., 2011)

2.4.1 Grunden för GPS

Själva teknologin bakom detta är rätt så komplex men konceptet är relativt simpelt. Varje GPS-satellit har atomklockor ombord som ger en exakt tid och är synkroniserade gentemot varandra. Varje satellit sänder kontinuerligt ut data som innehåller deras tid och position i form av radiovågor. GPS-mottagaren avkodar signalerna och beräknar distansen mellan satelliterna och mottagaren genom att veta hur länge det tog för signalerna att nå mottagaren och var satelliterna befann sig i det ögonblicket de sändes (Hoffman, 2016). Positionen beräknas sedan med hjälp av en metod som kallas trilateration. För detta krävs att man är inom räckvidd av minst tre GPS-satelliter. Trilateration är att man med hjälp av att veta distansen från satelliterna till mottagaren kan använda sig av klotets geometri för att bestämma positionen, där alla klot skär varandra (se figur 5) är positionen som man befinner sig vid. (MiTAC Digital Technology Corp, 2018)



Figur 5. Visuellt förklaring av trilateration (Saud, 2015)

2.5 PWM-modul

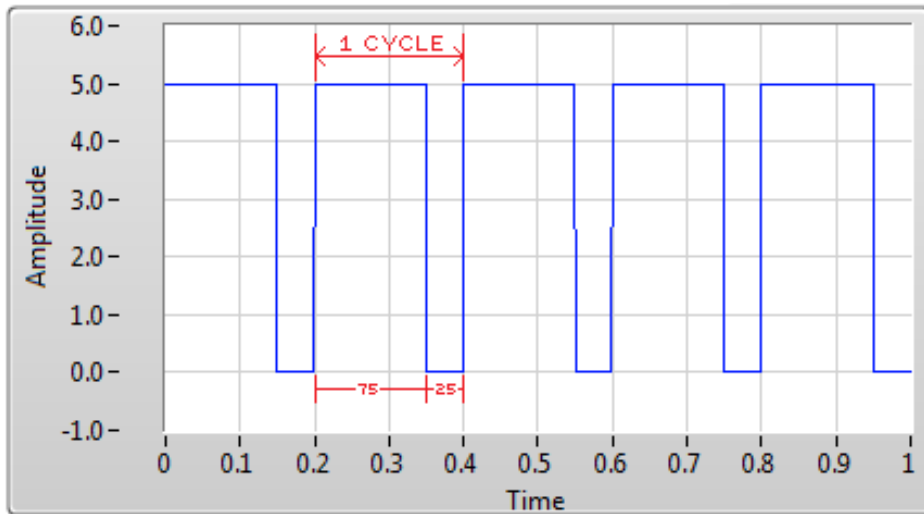
2.5.1 PWM

Båtmotorns rotationshastighet var från början reglerad med hjälp av motståndskretsar som aktiverades med en vridströmställare kopplad till utombordarhandtaget. Det fanns fem hastighetslägen för framkörning, tre lägen för back samt ett nolläge för avstängning av motorn.

Detta system lämpar sig inte för vårt ändamål, eftersom vi vill kunna reglera hastighet samt färdriktning elektroniskt från mikroprocessorn. För att få en stabilare positions hållning behöver vi även kunna reglera hastigheten steglöst. Vi kom fram till att pulsbreddsmodulering, PWM (pulse width modulation), skulle vara den lämpligaste lösningen. Detta eftersom det skulle möjliggöra steglös reglering samt även anses vara en energisnål metod.

Pulsbreddsmodulering används främst till att reglera effektmatningen till likspänningsmotorer genom att slå på och av spänningsmatningen med en så hög frekvens att motorn inte kan urskilja det från en konstant spänningsmatning. Det är alltså den så kallade tillslagstiden som

regleras, vilken bestämmer hur lång tid spänningsmatningen är aktiverad under en viss cykeltid. Utspänningen kommer då att beté sig som en konstant spänning med varierande amplitud. Exempelvis 75% tillslagstid med 5 V ger en utspänning på $0.75 \cdot 5 \text{ V} = 3.75 \text{ V}$ (se figur 6). (National Instruments Corporation, 2016)



Figur 6. Exempel på 75% tillslagstid vid 5 V och 0.2 sekunders cykeltid ($f = 1/T = 1/0.2 = 5 \text{ Hz}$).
(National Instruments Corporation, 2016)

2.5.2 Planering & tillverkning

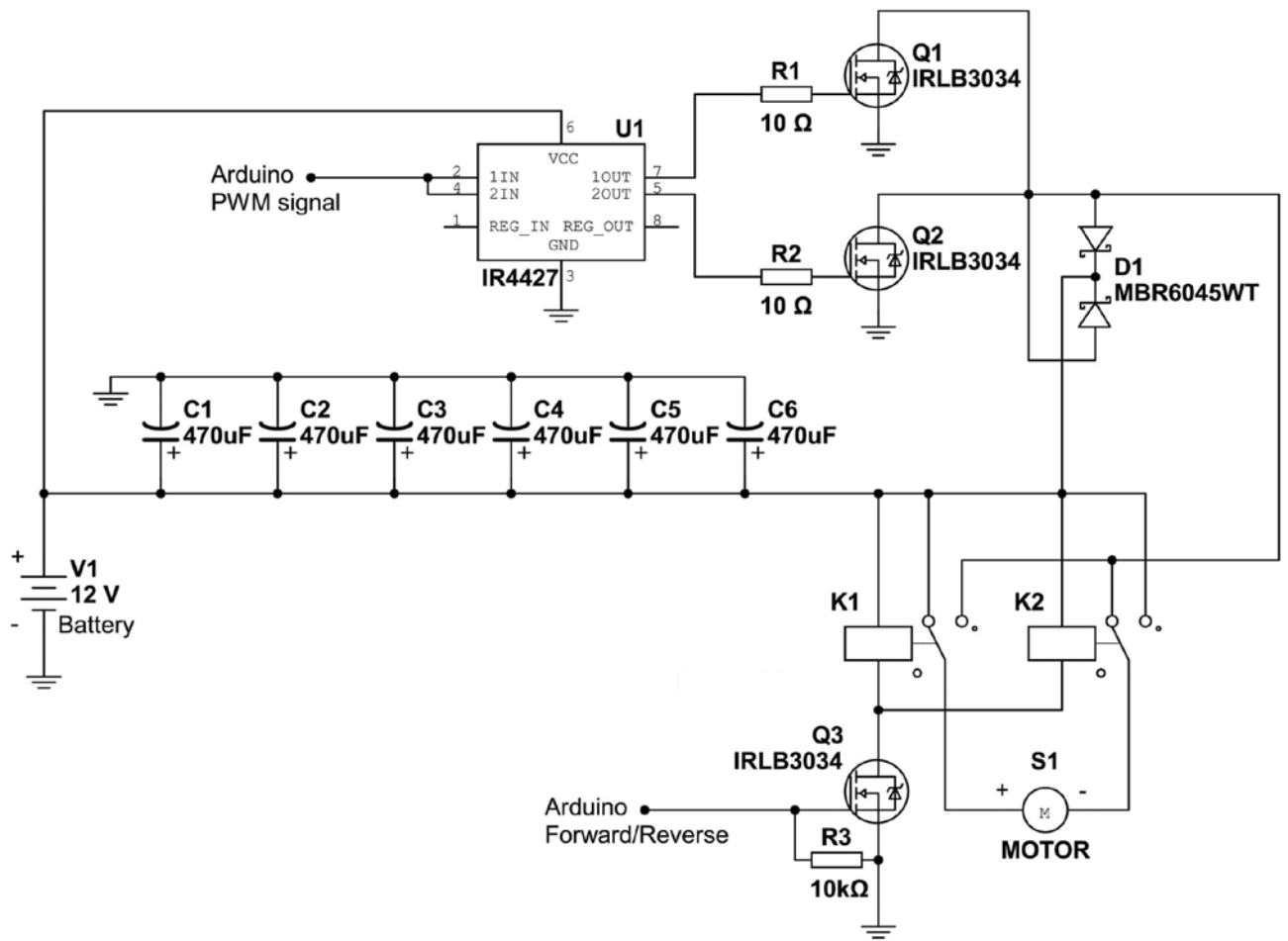
Vi valde att konstruera en egen PWM-modul anpassat för vårt ändamål. Som tidigare nämnt ville vi kunna reglera båtmotorns effekt samt rotationsriktning från mikrokontrollern. Därför hamnade vi konstruera en modul som kan använda PWM från mikrokontrollern till att styra huvudströmmen till motorn. Modulen måste också kunna använda en digital utgång från mikrokontrollern till att ändra rotationsriktningen.

För att reglera tillslagstiden på pulsbreddsmoduleringen använder vi N-kanals MOSFET av anrikningstyp med en drivenhet inkopplad på gaten. Med hjälp MOSFET:arna kan den relativt höga strömmen till motorn (max 30 A vid full effekt) styras med en låg ström till gaten. Drivenhetens uppgift är att öka spänningen och strömmen till gaten, så att MOSFET:arna leder fullständigt. Detta betyder att tiden som MOSFET:en är i övergångsskedet mellan ledande och icke-ledande minimeras, vilket i sin tur leder till mindre energiförlust samt värmeutveckling (Kalaff, 2014). För att ytterligare minska på värmebildningen använder vi två identiska MOSFET:ar parallellt för samma ändamål som därav delar på effekten.

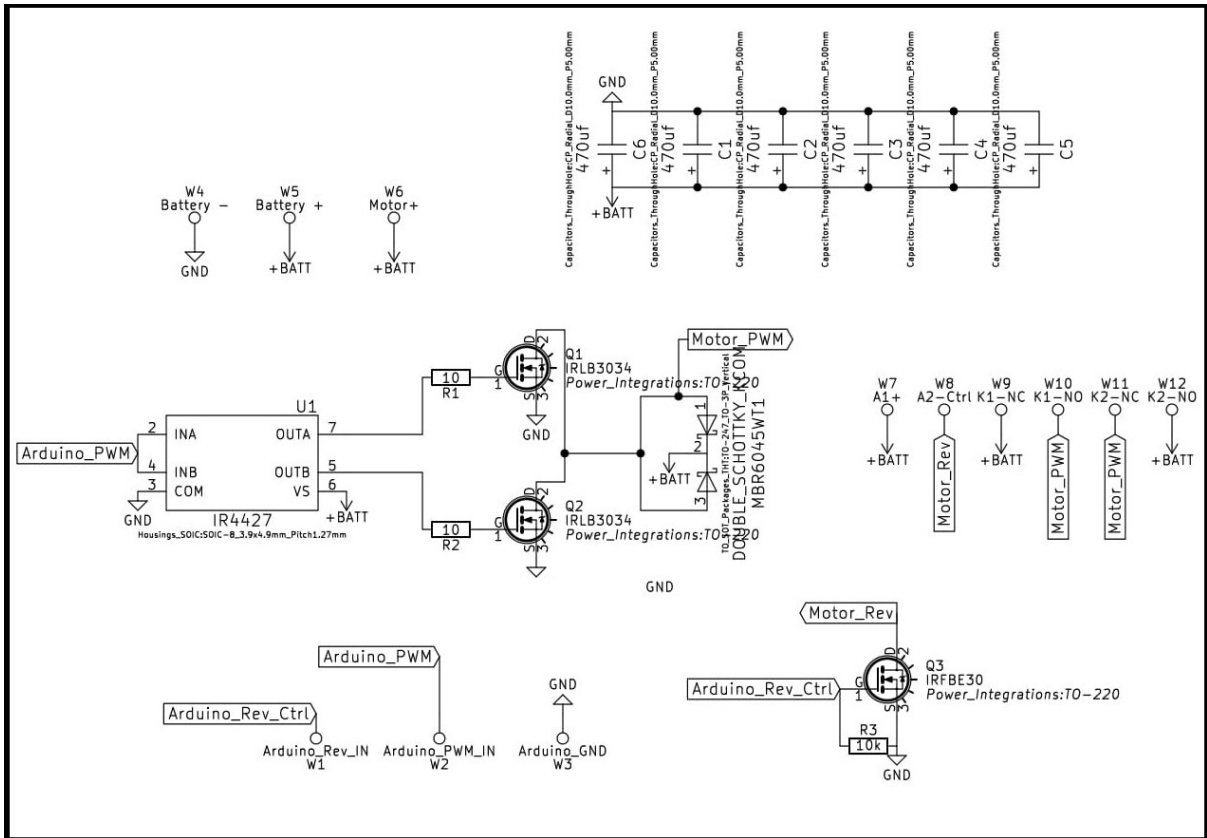
Spänningspikar och störningar som uppstår vid pulsbreddsmoduleringen av motorn dämpar vi med hjälp av en schottkybarriär samt en kondensatorbank. (Torrens, 2011)

Rotationsriktningen på elmotorn styrs med hjälp av två växelreläer som växlar om polariteten till motorn. Reläerna får sin matningsspänning via en MOSFET, vars gate aktiveras då de får en styrsignal från en digital utgång från mikrokontrollern. För att säkerställa att MOSFET:en hålls icke-ledande då den inte är aktiverad kopplades ett motstånd mellan gaten och jord, en så kallad pull-down resistor.

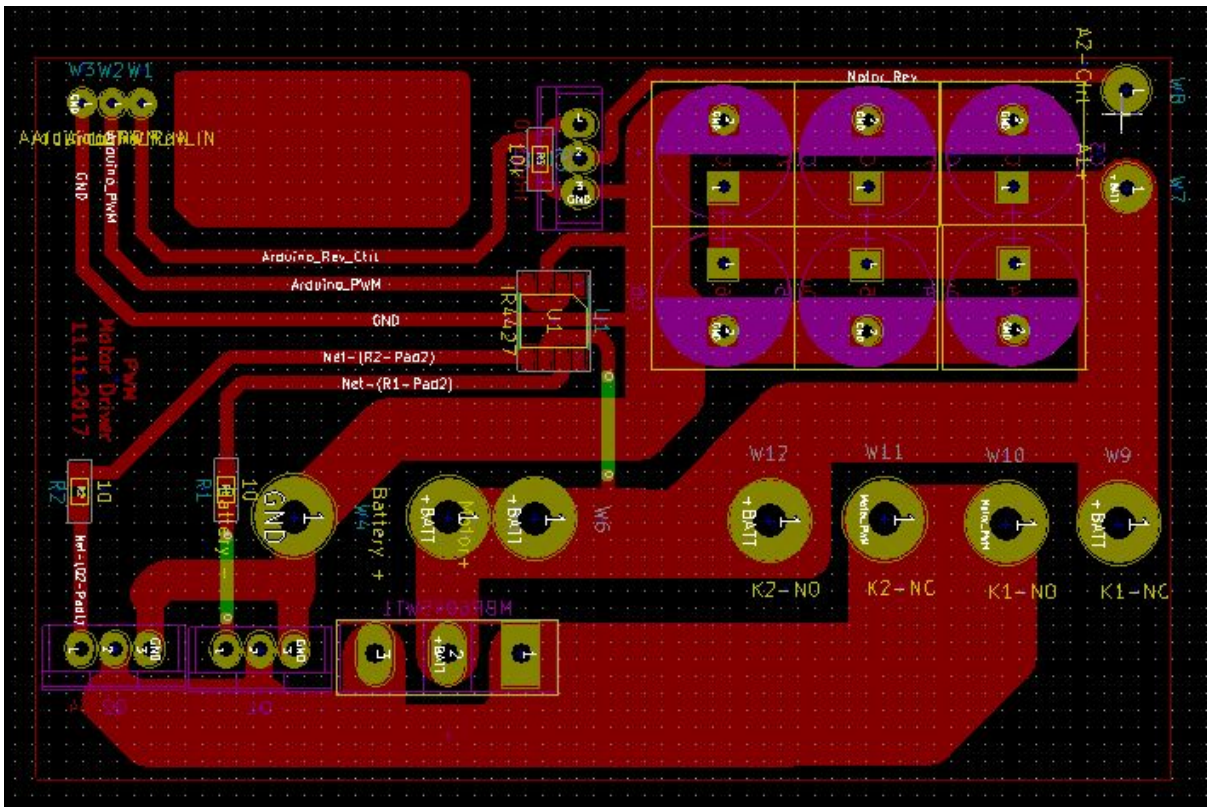
Då vi valt komponenterna gjorde vi ett kopplingsschema på PWM-modulen i Multisim (se figur 7). Sedan sattes kretskortstillverkningen igång. Vi fick en rekommendation om att tillämpa programvaran KiCad. Det är en programvara med öppen källkod avsedd för design av mönsterkort (PCB) samt kopplingsscheman (KiCad, 2018). När man designar ett mönsterkort i KiCad, är första steget att placera ut alla komponenter på en ritning (se figur 8). Då man gjort detta lägger man in komponenternas "footprints", det vill säga vilka standardiserade mått de har mellan sina ben. När detta är insatt kan programmet generera en så kallad Gerberfil av ritningen, som används vid tillverkning av mönsterkort (se figur 9). Programmet placerar ut komponenternas botten med hjälplinjer som visar hur de bör kopplas ihop. Sedan var det relativt simpelt att designa kopparspårens form.



Figur 7. Huvudschema på PWM-modulen ritat i NI Multisim.



Figur 8. Steg 1: Schemat i KiCad.

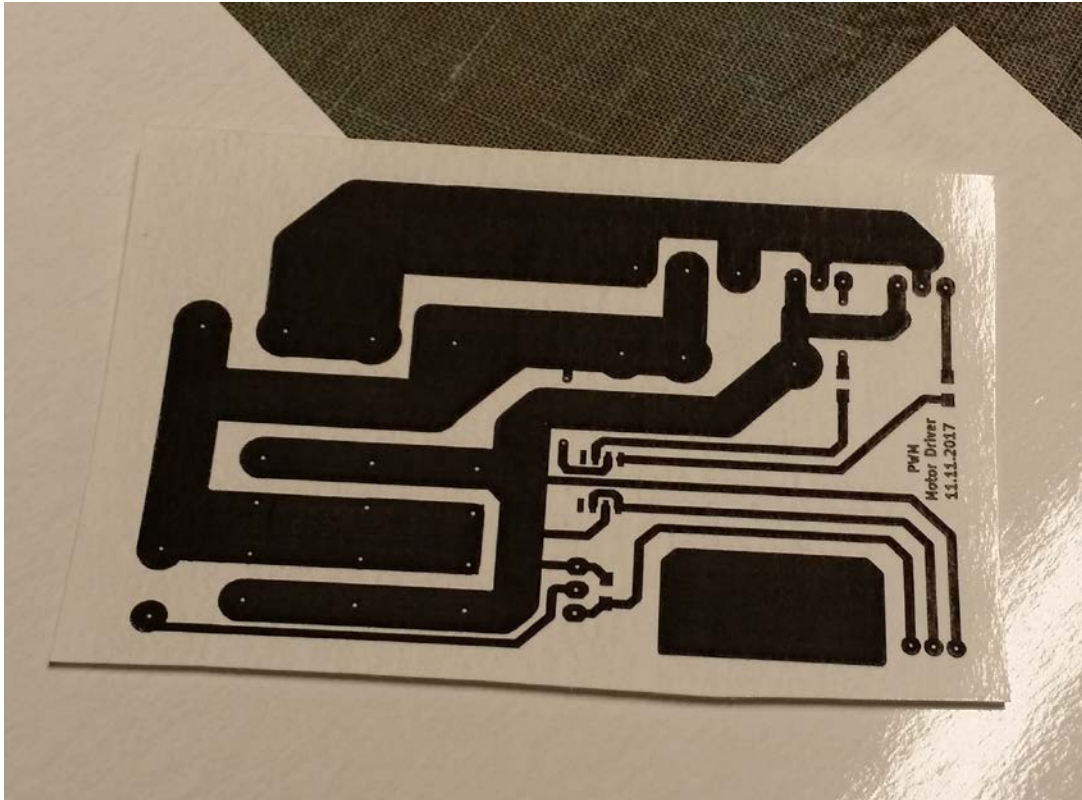


Figur 9. Steg 2: Gerberfilen i KiCad.

Gerberfilen skrevs ut i spegelvänt format på ett blankt papper med en svartvit laserskrivare (se figur 10). Följande steg var att lägga utskriften mot kopparplattan och stryka den, så att utskriftsfärgen brann fast på kopparytan (se figur 11). Pappret kunde sedan lösas upp i vatten och färgen blev kvar på kortet. Då vi kommit så här långt var det dags att framkalla mönsterkortet. Kortet lades i ett bad med 25% natriumpersulfat ($\text{Na}_2\text{S}_2\text{O}_8$) och fick ligga där tills kopparen frätts bort på alla ställen förutom där utskriften fastnat (se figur 12). Vattnet värmdes till ca 40 grader, för att gynna framkallningsprocessen. Det tog cirka 20 minuter tills kopparytan frättes bort och kortet kunde tas upp ur badet. Det färdiga mönsterkortet kan ses i figur 13.

Hål borrades sedan genom mönsterkortet för komponenternas ben. Sedan återstod endast fastlödning av komponenterna samt kablar för inkoppling. Reläerna för rotationsriktningsstyrningen kopplades även in. Slutresultatet kan ses i figur 14 och 15.

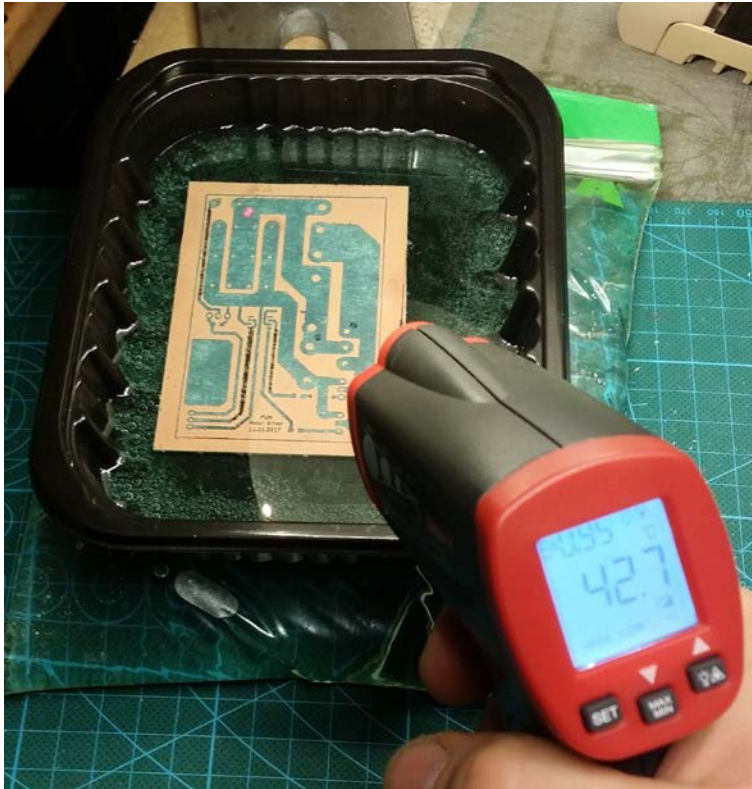
Till slut provkördes den färdiga PWM-modulen med 20 kHz. Först programmerades mikrokontrollern. När det var klart kopplades en potentiometer (10 k Ω) och en brytare fast i mikrokontrollern på en analog respektive digital ingång. Sedan kopplades två digitala utgångar från mikrokontrollern till PWM-modulen, samt 12 V matningsspänning från en ackumulator. Med hjälp av programmet kunde motorns rotationshastighet nu regleras med potentiometern och rotationsriktningen kunde bytas genom att aktivera brytaren. Modulen fungerade med andra ord som vi hade planerat.



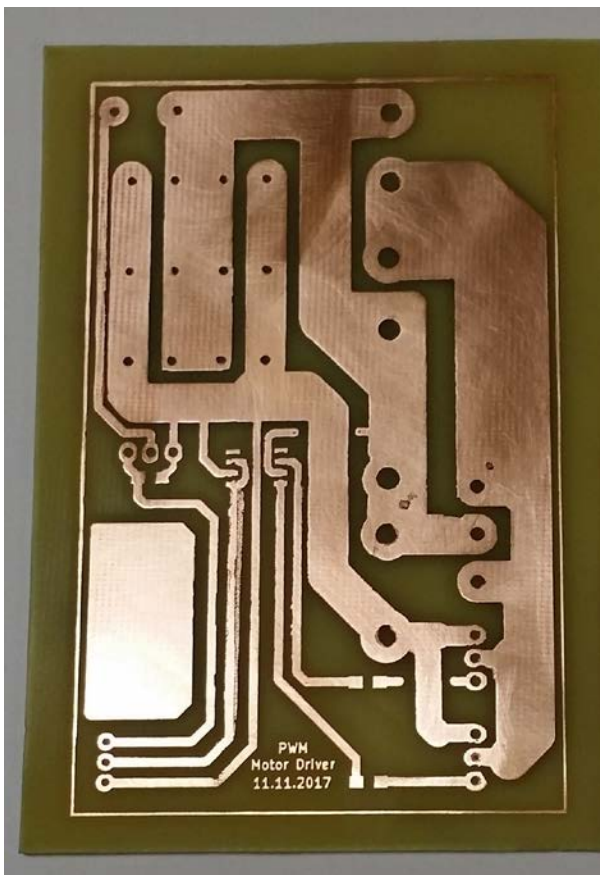
Figur 10. Gerberfilen utprintad spegelvänd på blankt papper.



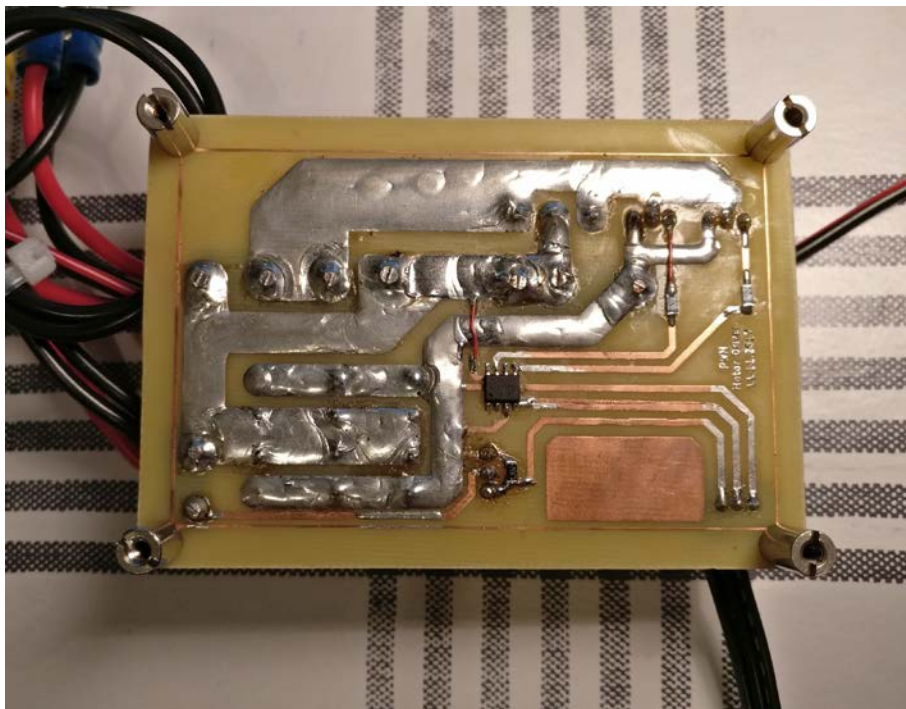
Figur 11. Utskriften stryks fast på kopparplattan.



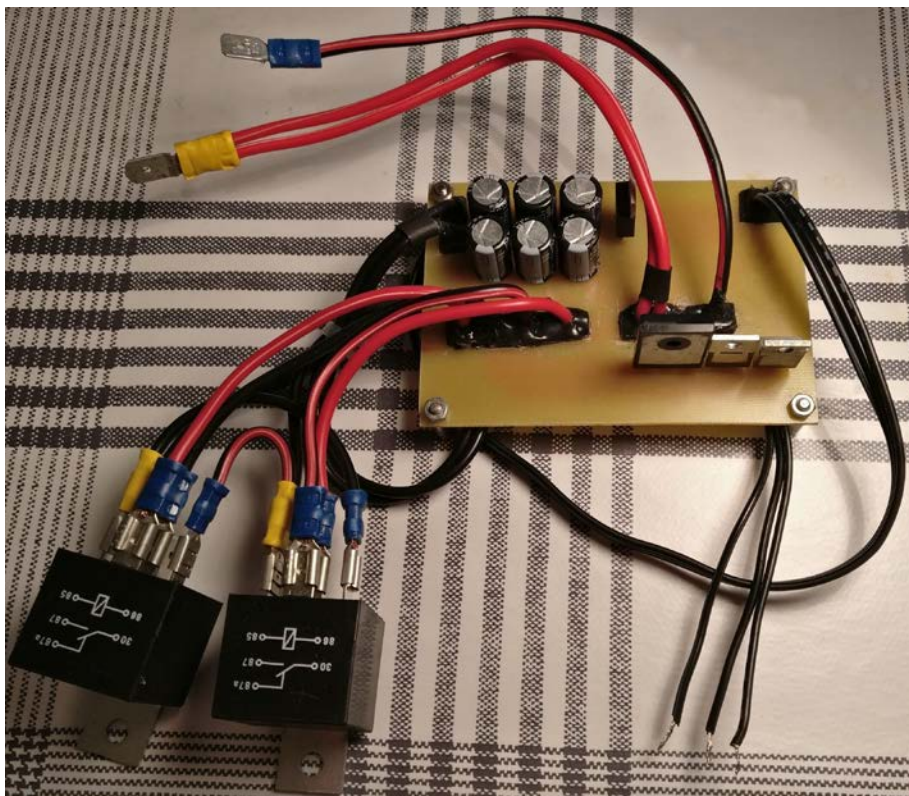
Figur 12. Framkallning.



Figur 13. Det framkallade mönsterkortet.



Figur 14. PWM-modulens baksida.



Figur 15. PWM-modulens framsida.

2.5.3 Komponentlista för PWM-modul

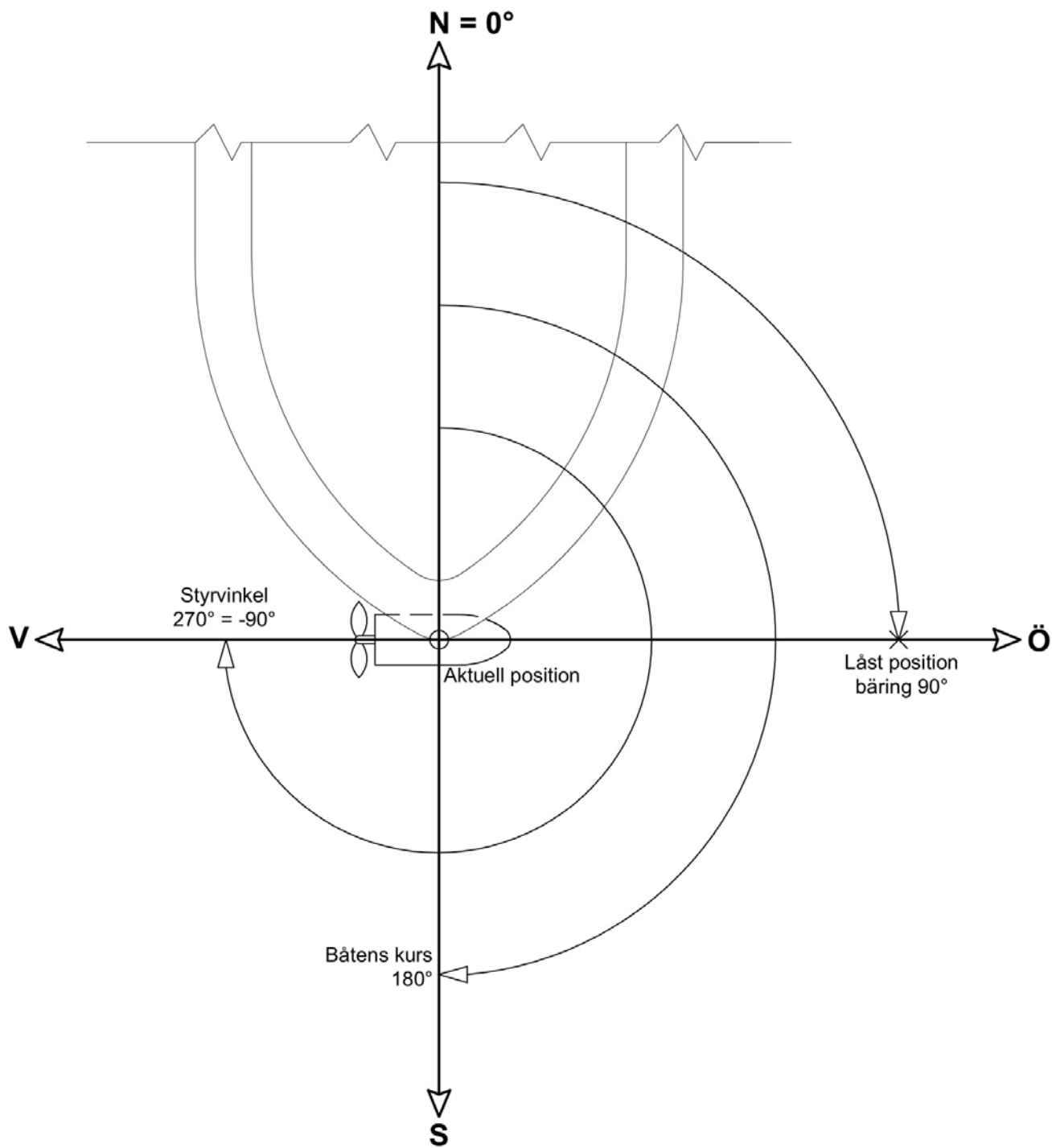
- 6 x 470 μ F kondensator
- 3 x IRLB3034 MOSFET
- 2 x växelrelä 12 V, 30/40 A
- 2 x 10 Ω motstånd
- IR4427 MOSFET drivenhet
- MBR6045WT schottkydiod
- 10 k Ω motstånd

2.6 Elektronisk kompassmodul

Den elektroniska kompassens uppgift i vårt system är att definiera båtens kurs. Då kursen samt bäringen från den rådande positionen till den låsta positionen är beräknad kan sedermera elutombordarens styrvinkel beräknas. Därmed utgör kompassen en nyckelroll eftersom det utan denna skulle det vara omöjligt att veta hur motorn skall riktas in så att båten kör mot den låsta punkten. (se figur 16).

Kompassmodulen som tillämpas i detta projekt är CMPS11 som tillverkas av Devantech. Modulen har färdigt inprogrammerad lutningskompensation och beräkningen sköts av en inkorporerad 16-bitars processor. Lutningsfelet kompenseras med hjälp av ett Kalmanfilter som tar i beaktande mätvärden från en treaxlad gyrometer samt treaxlad accelerometer. Vinkelvärdet från magnetometern korrigeras sedan med det beräknade felet kontinuerligt. Kommunikationen mellan kompassmodulen och mikroprocessorn kan ske med antingen UART eller I2C protokoll. (HobbyTronics, 2018) Vi valde att använda UART på grund av att vi har mer erfarenhet av det eftersom det även tillämpas av GPS-modulen i detta projekt.

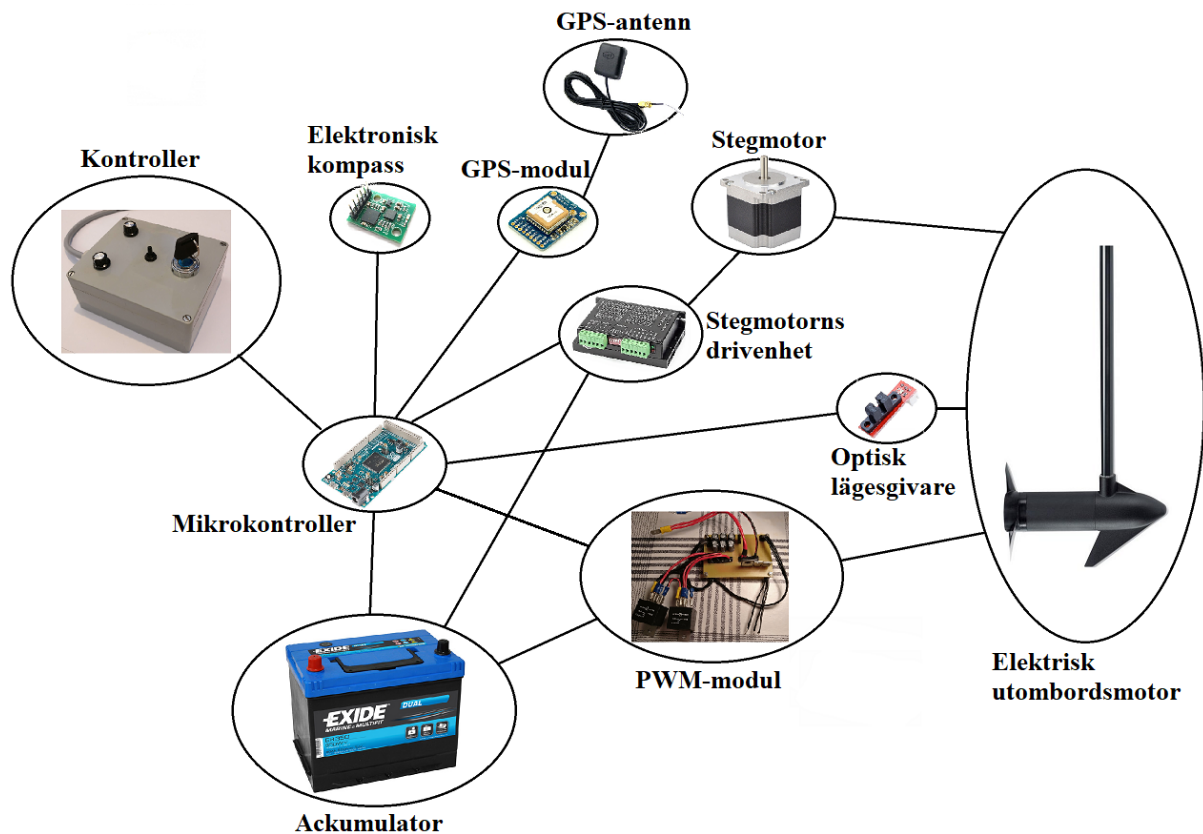
Vid testning framkom att kompassmodulens prestanda från fabriken inte var tillräckligt god och således utfördes en kalibreringsprocedur enligt anvisningarna från tillverkaren. Kalibreringen förbättrade prestandan till en tillfredsställande nivå.



Figur 16. Definition av vinklar. Båt och motor sett ovanifrån (schematiserat).

3. SYSTEMUPPBYGGNAD

För att få en helhetsbild av systemet gjorde vi en överblicksbild (se figur 17).



Figur 17. Överblick av systemet.

Mikrokontrollern får 12 V matningsspänning från akkumulatorm. Den elektriska utombordsmotorn får sin matning från akkumulatorm via PWM-modulen, vars tillslagstid styrs med PWM från en av mikrokontrollerns digitala utgångar. Utombordsmotorn's vinkel styrs med remtransmission (1:3) från stegmotorn som får sin matningsspänning från drivenheten. Drivenheten får i sin tur 12 V matningsspänning från akkumulatorm medan mikrokontrollern ger pulssignaler och en riktningssignal via två digitala utgångar som styr stegmotorn. Eftersom stegmotorn inte har någon position definierad från början finns den optiska givaren som används då programmet körs igång i en programfunktion som söker upp och definierar nollläget för stegmotorn. Med kontrollern kan man med nyckelströmställaren välja mellan tre lägen: avstängt, manuellt eller ankarläge. I manuellt läge kan man med

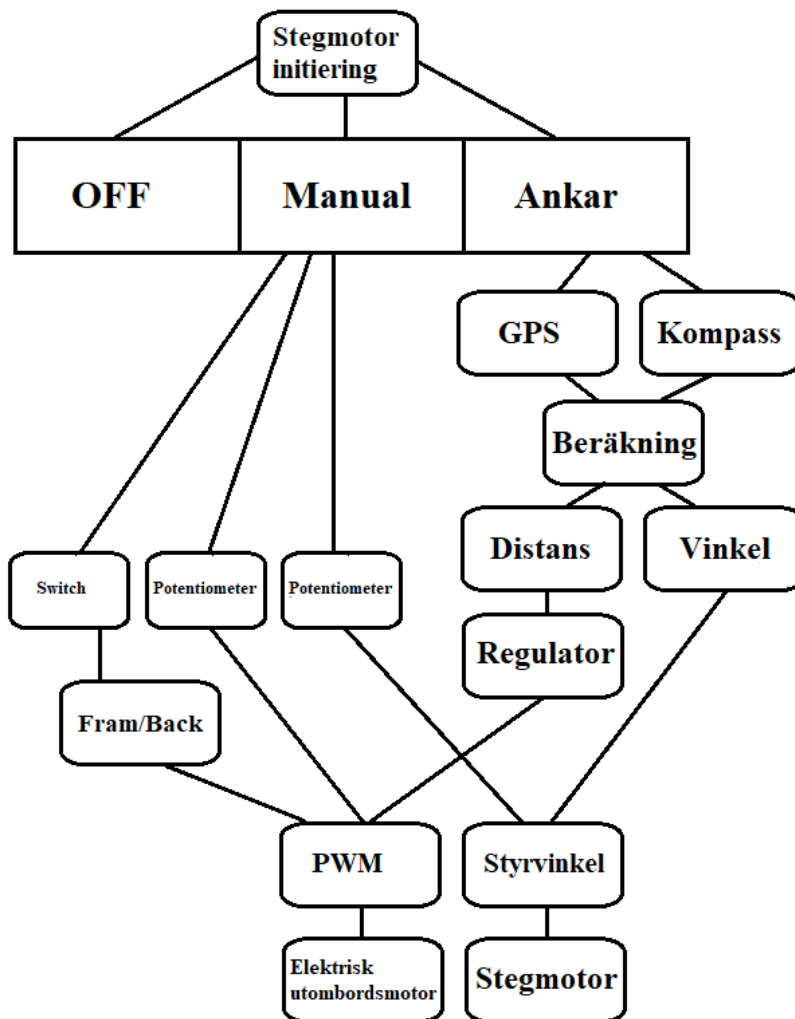
potentiometrarna styra vinkelläge respektive gaspådrag till utombordsmotorn. Fram/back kan väljas med en vippbrytare.

Med hjälp av positionsdata från GPS-modulen samt båtens riktning från den elektroniska kompassen beräknas styrvinkel och gaspådrag i Arduinon. Dessa beräknas och regleras automatiskt i ankarläget.

4. PROGRAMMERING

Programvaran har som tidigare nämnts två driftlägen och hårdvarumässigt är det en nyckelbrytare som bestämmer vilken av dessa lägen som är aktiverat. Respektive läge har sedan sina specifika funktioner som läser in värden, utför beräkningar och sedermera påverkar hårdvaran via mikrokontrollerns stift.

Innan programmering inleddes gjorde vi ett enkelt blockschema som vi kunde använda som riktlinje vid programmeringen (se figur 18).



Figur 18. Riktlinje av programmet.

Snuttar av huvudfunktionerna i programkoden kommer att förklaras och redovisas i detta kapitelns underrubriker. Vi kommer inte att behandla hela koden på en gång eftersom det inte skulle medföra någon glädje för läsarna, utan vi satsar istället på de relevanta delarna.

4.2 Avstängt läge

Då nyckelbrytaren vrids i läge 0 körs kodsektionen för avstängt läge igång. Då är det tänkt att ingenting skall ske, därmed ställs gaspådraget till noll. Flaggorna som talar om ifall initieringen av stegmotorn skett samt att positionen är låst nollställs.

```
// OFF-LÄGE -----  
----  
if (KeySwitchState1 == LOW && KeySwitchState2 == HIGH && KeySwitchState3 ==  
HIGH) {  
  
  Throttle(0); // Stänger av gaspådraget  
  Homing = 0; // Nollställer flaggan för att nolläget hittats  
  Locked = 0; // Nollställer flaggan för att positionen är låst  
}
```

4.2 Manuellt läge

Om nyckelbrytaren vrids i läge 1 aktiveras manuellt läge. Det första som då sker är att en funktion vars uppgift är att söka upp och definiera nolläget för stegmotorn körs igång.

Störningar på de analoga ingångarna som potentiometrarna är kopplade till visade sig vara ett stort problem då de var så kraftiga att de fick stegmotorn att bli orolig. Problemet försökte först lösas hårdvarumässigt med hjälp av lågpasfilter. Lågpasfiltren hjälpte till med att få bort en del av störningarna men inte fullständigt. Slutligen hittades ett färdigt programmerat bibliotek för mikroprocessorn som mjukvarumässigt filtrerar ingångarna, kallat ResponsiveAnalogRead (Clarke, 2016). Biblioteket visade sig vara mycket effektivt och i kombination med lågpasfiltret återstod inte några störningsproblem.

```
// MANUELLT LÄGE -----  
----  
else if (KeySwitchState1 == HIGH && KeySwitchState2 == LOW && KeySwitchState3  
== HIGH) {  
  
  SteeringPot.update(); // Uppdaterar ResponsiveAnalogRead  
  ThrottlePot.update();  
  
  Locked = 0;  
  Home(); // Kallar på funktionen för sökning av nolläge  
  
  // Styrvinkel
```

```

int SteeringValue = SteeringPot.getValue();
int SteeringValueMapped = map(SteeringValue, 0, 1022, -180, 180); //
Konverterar potentiometers värde till grader
SteeringAngle(SteeringValueMapped);

// Gaspådrag
int ThrottleValue = ThrottlePot.getValue();
int ThrottleValMapped = map(ThrottleValue, 0, 1022, 0, 100);
Throttle(ThrottleValMapped); // Skickar potentiometers värde till funktionen
för gaspådrag

// Färdriktning
Direction(digitalRead(DirectionSwitch)); // Skickar switchens läge till
funktionen för fram/back
}

```

4.2.1 Initiering av stegmotorns position

På grund av att riggröret kan vara i vilken position som helst vid uppstarten av programmet måste ett initieringsprogram köras vid uppstart. Programmet kör stegmotorn motsols ända tills den optiska gränslägesbrytaren aktiveras. Då brytaren gett ett utslag definieras positionen som noll. Därefter körs motorn medsols ett halvt varv eftersom potentiometern för styrningen skall verka i området -180° till 180° .

```

// Funktion för sökning av nolläge
void Home() {
  if (Homing == 0) { // Körs endast på nytt om brytaren har varit i off-läge
    stepper.setMaxSpeed(150.0); // Stegmotorns tillfälliga hastighet för
nollägessökningen
    stepper.setAcceleration(100.0); // Stegmotorns tillfälliga acceleration för
nollägessökningen
    while (!digitalRead(HomeSwitch)) { // Kör stegmotorn tills den optiska
gränslägesbrytaren aktiveras
      stepper.moveTo(HomingInit);
      stepper.run();
      HomingInit--; // motsols
      delay(5);
    }
    delay(500);
    stepper.setCurrentPosition(0); // Tillfälligt nolläge
    stepper.runToNewPosition(600); // Kör tillbaka  $400 * 3/2 = 600$  steg =  $0.5$  varv
=  $180$  deg
    stepper.setCurrentPosition(600); // Slutligt nolläge.  $1200/2 = 600$ .
    delay(500);
    Homing = 1;
  }
}
}

```

4.2.2 Styrvinkel, gaspådrag samt fram/back

Funktionen för styrvinkeln kör stegmotorn till den önskade positionen. Den tar in ett börvärde mellan -180° och 180° och konverterar detta till motsvarande steg från nollläget. Detta värde skickas sedan till en funktion i stegmotorns bibliotek som kör stegmotorn. Eftersom motorns position redan är definierad kan funktionen `moveTo` räkna ut hur många steg som krävs från den rådande positionen samt åt vilket håll den bör gå för att nå den önskade positionen.

```
// Funktion för styrvinkel (-180 -> 180)
void SteeringAngle(int angle) {
  int SteeringAngleToSteps = map(angle, -180, 180, 0, 1200);
  stepper.moveTo(SteeringAngleToSteps);
  stepper.setSpeed(70);
  stepper.runSpeedToPosition();
}
```

Gaspådragets funktion genererar med hjälp av PWM en fyrkantsvåg med 8-bitars resolution på tillslagstiden. Det skulle gå att få högre resolution men i vårt fall räcker 256 steg mer än väl. Frekvensen sattes till 20 kHz som är utanför hörområdet för att undvika störande högfrekvent ljud.

```
// Funktion för gaspådrag. Kan ta emot ett värde mellan 0 och 100%
void Throttle(int ThrottleVal) { // Genererar en fyrkantsvåg med en
  // specifik duty-cycle mellan 0 och 255 (8-bit).
  int DutyCycle = (255./100.)*ThrottleVal;
  // Frekvensen ändrad till 20kHz i Arduinos variant.h-fil:
  analogWrite(PWMPin, DutyCycle); // #define PWM_FREQUENCY 20000
} // #define TC_FREQUENCY 20000
```

Fram/back aktiverar en digital utgång som går till PWM-modulen. Spänningen styr en MOSFET som aktiverar ett relä. Reläet växlar i sin tur om polariteten på utombordsmotorn.

```
// Funktion för färdriktning, 0 = Back, 1 = Framåt
void Direction(int dir) {
  if (dir == 0) {
    digitalWrite(DirectionPin, HIGH);
  } else {
    digitalWrite(DirectionPin, LOW);
  }
}
```


4.3 Ankarläge

Förutsatt att nycken vrids i läge 2 körs kodsnutten för ankarläge. Såvida stegmotorn inte redan initierats körs initieringsprogrammet igång. Därpå utförs diverse beräkningar med de nuvarande koordinaterna, de låsta koordinaterna samt kompassens värde med specifika intervaller för att försöka hålla båten på plats.

```
// LÅST LÄGE -----  
-----  
else if (KeySwitchState1 == HIGH && KeySwitchState2 == LOW && KeySwitchState3  
== LOW) {  
  
    Home();  
    GPS_lock();  
  
    double CurrentLatRad = GPSlat*(PI/180); // Grader -> radianer för att kunna  
användas i beräkningsfunktionerna  
    double CurrentLongRad = GPSlong*(PI/180);  
    double LockedLatRad = LockedLatitude*(PI/180);  
    double LockedLongRad = LockedLongitude*(PI/180);  
  
    if(currentMillis - previousMillis2 > DistanceCalcInterval) {  
        previousMillis2 = currentMillis;  
  
        GPScalcDistance(CurrentLatRad, CurrentLongRad, LockedLatRad, LockedLongRad);  
    }  
  
    if(currentMillis - previousMillis3 > GPSbearingCalcInterval) {  
        previousMillis3 = currentMillis;  
  
        GPScalcBearing(CurrentLatRad, CurrentLongRad, LockedLatRad, LockedLongRad);  
    }  
  
    if(currentMillis - previousMillis4 > SteeringAngleCalcInterval) {  
        previousMillis4 = currentMillis;  
  
        CalculatedAngle = CalcSteeringAngle(GPSbearingDeg, CompassHeadingDeg);  
    }  
  
    SteeringAngle(CalculatedAngle); // Skickar den senast beräknade styrvinkeln  
till fkt för styrvinkel  
    ThrottleReg(DistanceFiltered); // Skickar det senast beräknade avståndet till  
fkt för gaspådrag i låst läge  
    }  
}
```

4.3.1 Låsning av GPS-koordinater

Denna funktion tar de rådande koordinaterna och definierar dessa som de låsta. Då båten börjar förflytta sig börjar dessa koordinater avvika från varandra eftersom de låsta koordinaterna är statiska medan de rådande koordinaterna ändras.

```
// Funktion för låsning av GPS-koordinat
void GPS_lock() {
    if (Locked == 0) {
        LockedLatitude = GPSlat;
        LockedLongitude = GPSlong;
        Locked = 1;
    }
}
```

4.3.2 Distans, GPS-bäring samt styrvinkelberäkning

Avståndet från de låsta koordinaterna till de rådande beräknas med Haversineformeln (Wikipedia, 2018):

$$h = \sqrt{\text{hav}(\varphi_2 - \varphi_1) + \cos(\varphi_1) * \cos(\varphi_2) * \text{hav}(\lambda_2 - \lambda_1)}$$
$$= \sqrt{\sin^2\left(\frac{\varphi_2 - \varphi_1}{2}\right) + \cos(\varphi_1) * \cos(\varphi_2) * \sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}$$
$$d = R * \text{hav}^{-1}(h) = 2R * \arcsin(\sqrt{h})$$

d = Avståndet mellan position 1 och position 2.

R = Jordens radie.

φ_1 = Latitud i radianer för position 1.

φ_2 = Latitud i radianer för position 2.

λ_1 = Longitud i radianer för position 1.

λ_2 = Longitud i radianer för position 2.

Vi funderade på att filtrera koordinaterna för att minska felvisning. Utgående från precisionstestet konstaterades dock att ett filter inte hjälper eftersom felen inte uppstår i en slumpmässig riktning utan består av flertal felvisningar på en specifikt punkt. Dessutom

tenderar felet att långsamt söka sig i en viss riktning. Vi valde att istället filtrera den uträknade distansen med ett exponentialfilter enligt formeln (Stanley, 2017):

$$y(k) = \alpha * y(k - 1) + (1 - \alpha) * x(k)$$

$y(k)$ = Filtrerat värde

$x(k)$ = Ofiltrerat värde

$y(k - 1)$ = Föregående filtrerat värde

α = Filtreringskonstant, $0 < \alpha < 1$

En låg filtreringskonstant ger en låg filtrering och en stor filtreringskonstant ger en hög filtrering.

Den filtrerade distansen skickas sedan till en funktion som reglerar gaspådraget.

Effektiviteten av filtreringen bör utvärderas mer i praktiken.

```
// Funktion som beräknar avståndet mellan låsta koordinater och nuvarande
koordinater
void GPScalcDistance(double LatCurrentRad, double LongCurrentRad, double
LatLockedRad, double LongLockedRad) {

    h = sin((LatCurrentRad-LatLockedRad)/2)*sin((LatCurrentRad-
LatLockedRad)/2)+cos(LatLockedRad)*cos(LatCurrentRad)*sin((LongCurrentRad-
LongLockedRad)/2)*sin((LongCurrentRad-LongLockedRad)/2);
    Distance = 2*R*asin(sqrt(h));

    DistanceFiltered = alpha*DistancePrev+(1-alpha)*Distance; // Exponentialfilter
    DistancePrev = DistanceFiltered;
}
```

Bäringen mellan de låsta positionen och de rådande positionen beräknas med (Veness, 2017):

$$\theta = \text{atan2}(\sin(\lambda_2 - \lambda_1) * \cos\varphi_2, \cos\varphi_1 * \sin\varphi_2 - \sin\varphi_1 * \cos\varphi_2 * \cos(\lambda_2 - \lambda_1))$$

φ_1 = Latitud i radianer för position 1.

φ_2 = Latitud i radianer för position 2.

λ_1 = Longitud i radianer för position 1.

λ_2 = Longitud i radianer för position 2.

Den beräknade bäringen skickas tillsammans med kompassens värde till en funktion som beräknar styrvinkeln.

```
// Funktion som beräknar bäring mellan låsta koordinater och nuvarande
koordinater
void GPScalcBearing(double LatCurrentRad, double LongCurrentRad, double
LatLockedRad, double LongLockedRad) {

    b1 = sin(LongCurrentRad-LongLockedRad)*cos(LatCurrentRad);
    b2 = cos(LatLockedRad)*sin(LatCurrentRad)-
sin(LatLockedRad)*cos(LatCurrentRad)*cos(LongCurrentRad-LongLockedRad);

    GPSbearing = atan2(b1,b2);
    GPSbearingQuadrantRad = fmod(GPSbearing+2*PI,2*PI);
    GPSbearingDeg = GPSbearingQuadrantRad*(180/PI);
}

```

Styrvinkeln fås genom att subtrahera kursen från bäringen till den låsta positionen. Eftersom motorn kan röras maximalt 180° i respektive riktning från nollpunkten måste vi implementera följande villkor i programvaran:

- Styrvinkel = GPS-bäring - kurs
- Beräknad styrvinkel > 180°: korrigerad styrvinkel = beräknad styrvinkel - 360°
- Beräknad styrvinkel < -180°: korrigerad styrvinkel = beräknad styrvinkel + 360°

```
// Funktion som beräknar elutombordarens styrvinkel (-180 -> 180)
double CalcSteeringAngle(double GPSbearing, double CompassHeading) {

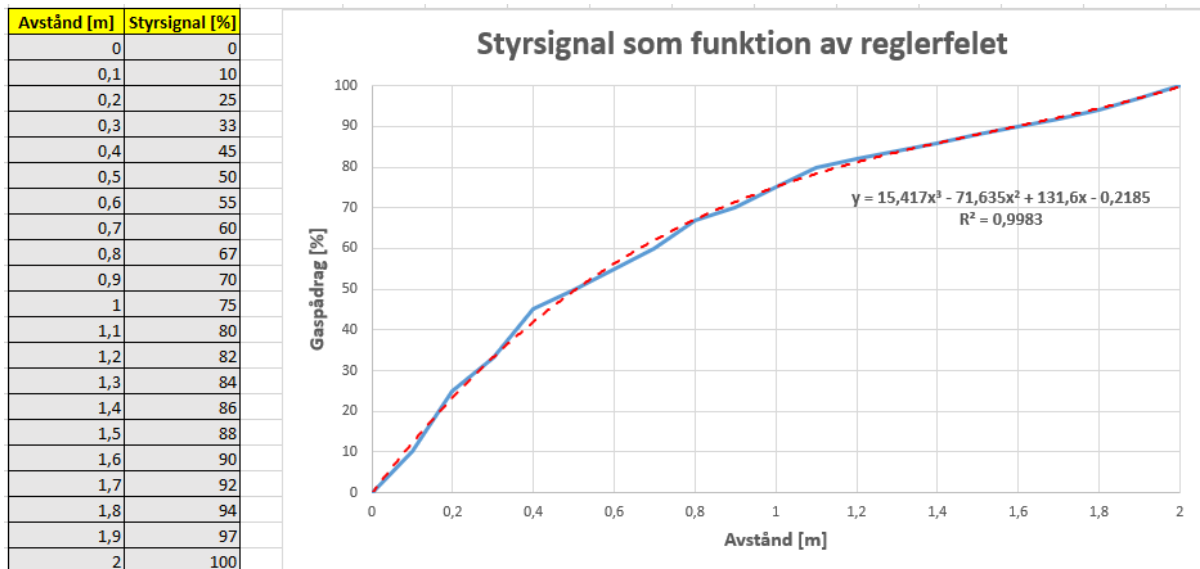
    double CalcAngle = GPSbearing - CompassHeading;
    if (CalcAngle > 180) {
        angle = (GPSbearing-CompassHeading)-360;
    } else if (CalcAngle < -180) {
        angle = (GPSbearing-CompassHeading)+360;
    } else {
        angle = GPSbearing - CompassHeading;
    }

    return angle;
}
}

```

4.3.3 Reglering av gaspådrag

Gaspådraget i ankarläget är anpassat enligt påhittade procentvärden för ett specifikt avstånd enligt värden som vi tyckte verkade rimliga. Sedan kunde minsta kvadratmetoden tillämpas för att bestämma en funktion för gaspådraget (se figur 19).



Figur 19. Kurvanpassning till tredje gradens polynom i Excel.

Vi valde att ha ett dödområde på 2 m från den låsta positionen som gaspådraget inte är aktivt på grund av GPS:ens felmarginal på upp till ett par meter samt det faktum att det enligt oss är obefogat att motorn gaspådrag aktiveras så fort båten förflyttas en aning.

```
// Funktion för beräkning av gaspådrag i låst läge
void ThrottleReg(double DistanceCur) {

    if (DistanceCur < 2) {
        DistanceReg = 0;
    } else {
        DistanceReg = DistanceCur-DeadZone;
    }
    u = 15.417*pow(DistanceReg, 3) - 71.635*pow(DistanceReg, 2) +
    131.6*DistanceReg - 0.2185; //kurvanpassningen

// Begränsar styrsignalen mellan 0-100%
    if (u <= 0) {
        ControlSignal = 0;
    } else if (u >= 100) {
        ControlSignal = 100;
    } else {
        ControlSignal = u;
    }
}
```

4.3.4 Kompass

Kompassens vinkel korrigeras med en deklination som kan specificeras i början av programmet. Eftersom vinkeln ges i 8-bitarsformat och beräkningarna utgår från 0-360° måste vinkeln först skalas om och sedan utförs deklinationskorrigeringen med hjälp av en modulusoperation. Orsaken till att man inte direkt kan subtrahera eller addera deklinationen från eller till den rådande vinkeln är att svaret skulle bli utanför området 0-360°.

```
// Kompass
// Frågar och läser in 8-bitars vinkeln
Serial2.write(CMPS_GET_ANGLE8);
CompassAngle8 = Serial2.read();
CompassHeadingMapped = map(CompassAngle8, 0, 255, 0, 360);
CompassHeadingDeg = (CompassHeadingMapped-(Declination)+360)%360; //
Korrigerar kompassen för magnetisk deklination
}
```

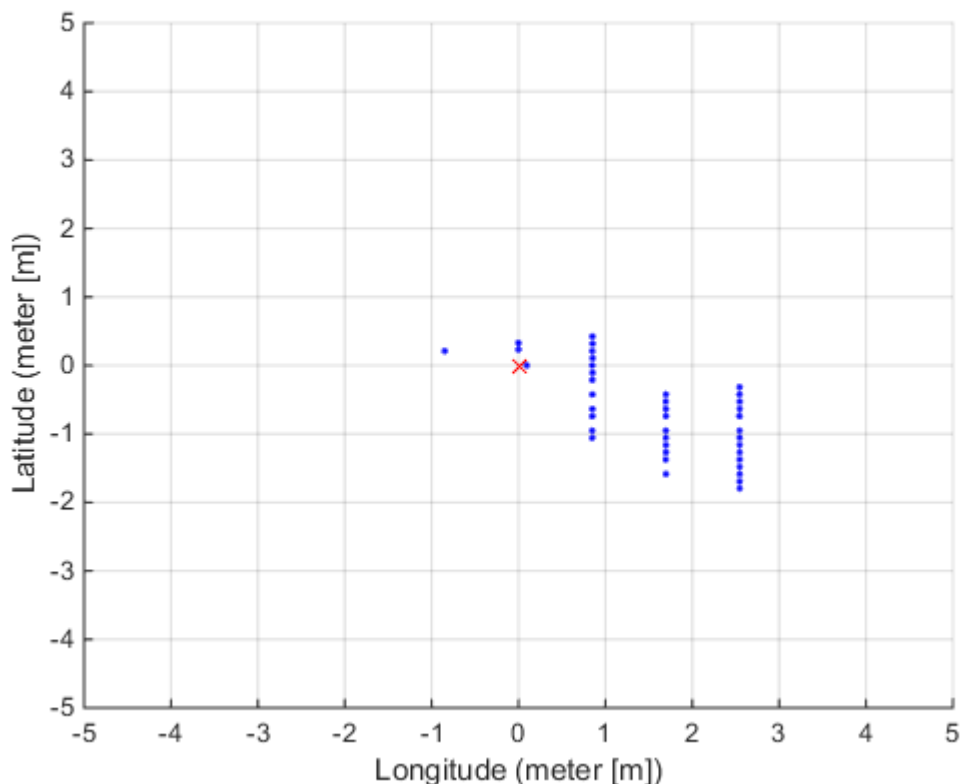
5. PRAKTISKA EXPERIMENT

5.1 Precision av GPS-modul

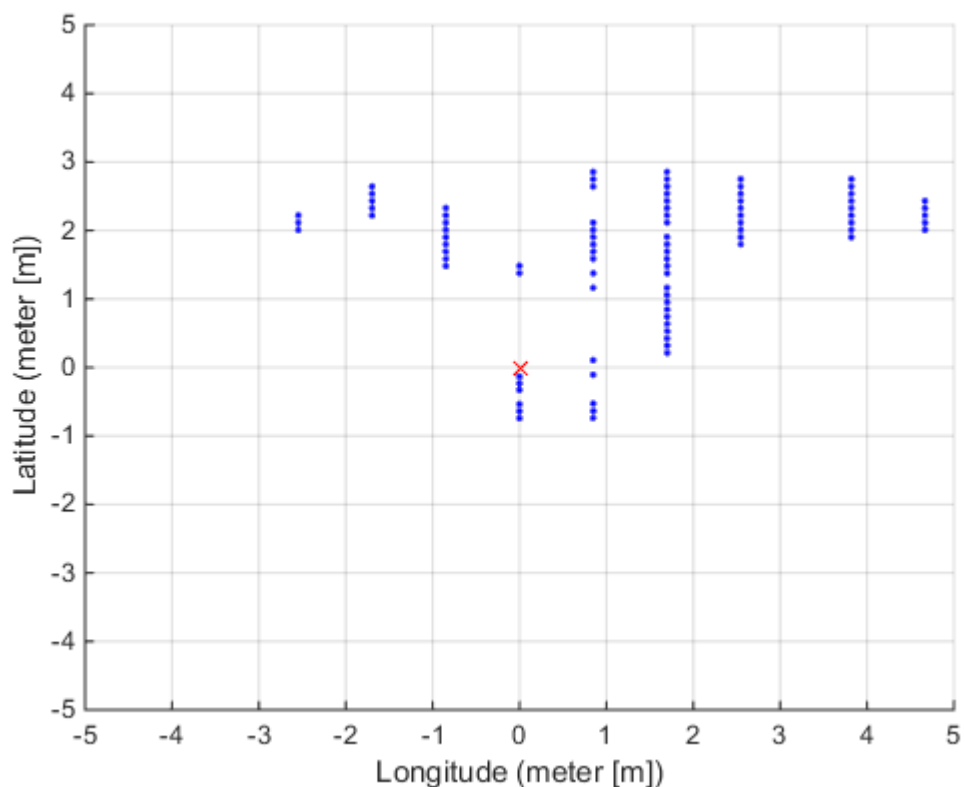
Då vi fått modulen så ville vi självklart göra praktiska experiment med den och se ifall det dom lovat i manualen verkligen stämde. Praktiska experimentet utförde vi genom att med mikrokontrollern logga med 1 Hz uppdateringsfrekvens i en halv timmes tid koordinater då modulen var stationärt positionerad, därefter sparade vi koordinaterna som en Excel-fil och plottade med matlab ett diagram för att kunna avläsa samt utforska visuellt vårt data.

Då vi gjorde detta ville vi även undersöka ifall det verkligen fanns någon skillnad mellan att aktivera SBAS (Satellite-based augmentation system) vilket fanns som en möjlighet i biblioteket vi använde oss av. Detta gjordes direkt efter på exakt samma position och samma program förutom att SBAS nu var avaktiverat.

I figur 20 samt figur 21 (se nedan) är röda krysset första positionen som loggats och därmed placeras i origo, blåa punkterna är de resterande 1799 mätvärdens positionsavvikelse i avseende på det första mätvärdet.



Figur 20. Mätresultat med SBAS aktiverat.



Figur 21. Mätresultat med SBAS avaktiverat.

Från graferna ovan kan man tolka att då SBAS är aktiverat är spridningen betydligt mindre, vilket gör det till ett enkelt val att använda sig av det i programmet.

5.2 Torrövning

På grund av isläget kunde vi inte utföra testkörningar till sjöss i det skede då programvaran var redo att utvärderas. Därmed konstruerade vi en oestetisk men fungerande kärra för att simulera vår båt (se figur 22). Kärnan skuffades sedan runt på torra land för att se hur systemet reagerar då man låst en position och därefter förflyttar kärnan till en ny position.

Propelleraxelns tätning är konstruerad så att den smörjs och kyls av omkringvarande vatten. Därför hade vi motorn nersänkt i en vattenbalja under testkörningen för att tätningen inte skall haverera på grund av friktion.

Gasreglaget visade sig arbeta tillfredsställande. Den reglerades proportionellt mot distansen från den låsta positionen. Därmed fungerar distansformeln samt pulsbreddsmoduleringen som planerat.



Figur 22. Kärran.

Reglaget av styrvinkeln arbetade däremot under all kritik. Visserligen ändrades vinkeln då man förflyttades, men inte som den borde. Därmed blev vår nästa etapp i projektet att ta reda på varför genom att undersöka programvaran och göra utföra ytterligare testkörningar.

Orsaken till detta problem visade sig vara i koden vi skrivit till kompassen. Eftersom vi använde oss av en 8-bits upplösning på vinkeln från kompassen, så var ett varv definierat som 0-255. Eftersom beräkningarna utgår ifrån att ett varv är 0-360° så leder detta uppenbarligen till problem med hur riggen svänger. Det här var en sak vi visste om vid programmeringen men som vi bara helt enkelt glömt bort att skriva i programmet. Efter att vi ändrade detta i koden fungerade systemet som förväntat.

Dock går det inte att dra slutsatser ur detta utan att testa systemet i praktiken. Enda detta visade var att principen är rätt i teorin, men regleringen av gaspådraget kommer troligtvis behöva en hel del justeringar med hjälp av experiment eftersom vi anser att endast reglera gaspådraget proportionellt mot distansen inte troligtvis är den bästa lösningen.

5.3 Utvärdering till sjöss

I slutfasen av projektet hittade vi öppet vatten, där vi kunde testköra de olika lägena (se figur 23). Båten sjösattes och därpå monterades motorn i fören. Testet inleddes med utvärdering av det manuella driftlägets funktionsduglighet. Det konstaterades att funktionsdugligheten var god, dock var styrningen emellertid i viss omfattning onödigt komplicerad på grund av att styrområdet är ett helt varv. För att testet inte skulle bli jävigt värvade vi en extern opartisk testpilot. Testpiloten konstaterade att det är sannolikt att man med ett par timmars övning skulle bemästra styrningen.

Under utvärderingen av ankarläget framkom några komplikationer. Regleringen av gaspådraget som vi anpassat som en funktion av avståndet från den låsta positionen visade sig i praktiken vara för aggressiv. Ett annat problem var att ifall gaspådraget kommer igång före vinkeln hunnit ändra klart blir regleringen aningen instabil. Dessa är saker är svåra att förutsäga utan praktiska experiment men nu hade vi fått den information som krävs för att veta vilka förbättringar som kan implementeras i programvaran. Förslag till förbättringar tas upp i ett eget kapitel.



Figur 23. Testkörning

6. FÖRBÄTTRINGSFÖRSLAG

I det praktiska experimentet visade sig anpassningen som vi gjort för gaspådraget i ankarläge vara för aggressiv och bör således planas ut för att få en stabilare gasreglering. Denna lösning är inte skriven i sten, utan ifall en annan regleringsmetod visar sig vara lämpligare kommer den naturligtvis att tas i bruk.

Ett annat problem var att om börvärdesförändringen på styrvinkeln är stor så hinner positionsregleringen bli instabil på grund av att gaspådraget är aktivt under tiden som vinkeln ändras. En lösning till detta skulle eventuellt vara att i programmet använda ett villkor som säger att gaspådraget inte aktiveras då stegmotorn för styrvinkeln körs. Risken med detta är dock att regleringen blir trög. Dessa lösningsförslag måste givetvis prövas fram i praktiken för att utvärdera om dessa förbättringar gör systemet robustare, så som det är tänkt i teorin.

En sak som vi övervägde redan i början av planeringen för projektet var att ifall vi skall göra kontrollern trådlös eller med kabel. Eftersom själva funktionen inte påverkas utav detta valde vi att använda kabel i och med att en trådlös kontroll inte hade fallit inom våra tidsramar för projektet. Det här är dock en sak vi kommer att implementera på egen tid efteråt, eftersom det förbättrar handhavandet då man inte har någon kabel i vägen som man kan trassla sig i.

Angående kontrollern (se figur 24) konstaterades under testkörningen att för att få en bättre känsla då man styr och gasar i manuella läget kunde potentiometerns ratt vara större till formatet. För att underlätta styrningen kunde det även prövas med att minska styrområdet från -180° - 180° till -90° - 90° eftersom back redan finns implementerat i koden samt på kontrollern.



Figur 24. Kontrollern.

Vi valde ett styrområde från -180° till 180° så att riggröret inte kan snurra flera varv i någondera rotationsriktning. Detta på grund av att kablarna som går ner till elmotorn inte skall tvinnas ihop sig och således leda till att stegmotorn hoppar över steg på grund av det ökade motståndet. En lösning till hur riggröret skulle kunna snurra fritt kunde evalueras eftersom detta skulle leda till en snabbare vinkelreglering. Till exempel om börvärdet ändras från -179° till 179° måste nu riggröret snurra nästan ett helt varv tillbaka istället för att ta den kortare vägen.

7. KOSTNADSKALKYL

En av arbetets grundtankar var att hålla utgifterna på en rimlig nivå. En kostnadsberäkning utfördes därmed i excel (se tabell 4) för att i slutet kunna utvärdera om målet uppfyllts. I

beräkningen tas hänsyn till alla komponenter samt diverse annat material. Även det som existerade från tidigare i våra lager som exempelvis elsnurran, vissa plåtar, bultar, muttrar brickor och skruvar. Dessa delars kostnader har uppskattats så noggrant som möjligt med rådande marknadsvärde för att hålla beräkningen korrekt.

Tabell 4. Kostnadskalkyl i Excel

Produkt	Beskrivning	Pris (frakt inkluderat)
Elektrisk utombordsmotor	Minn Kota Endura C2 30	199,00 €
Mikrokontroller	DUE R3 SAM3X8E 32-bit ARM Cortex-M3 Arduino DUE R3 Control Board	11,51 €
GPS-modul	Adafruit Ultimate GPS Breakout - 66 channel w/10 Hz updates	47,00 €
U.FL till SMA Adapter	Hot Antenna 25cm SMA IPX1.13 IPX I-PEX UFL To SMA Female RF Adapter Cord Cable	0,85 €
Antenn	GPS Active Antenna Aerial Male Plug Connector Adapter with 1m Cable	7,75 €
Stegmotor	DE Ship Nema 23 Bipolar 0.9deg(400 steps/rev) 12V 0.38A 0.9Nm Small Current	28,35 €
Drivmodul för stegmotor	Upgraded 4A TB6600 Single Axis Stepper Motor Driver Controller For 57BYG250H Mot	7,19 €
Remhjul	3M 3mm Pitch 48 Teeth Timing Belt Pulley AF-type 17mm	16,26 €
Kuggrem	345mm	20,00 €
Elektronisk kompass	CMPS11 - Tilt Compensated Compass Module	29,05 €
2 x Potentiometer	10kΩ	0,60 €
Nyckelbrytare	Universal 6 Wire Motorcycle Ignition Key Switch 3 Pos 2 Keys Pit Quad Dirt Bike	10,02 €
PWM-modul	Komponenter	20,00 €
2 x Lagerenheter	30MM LUCF206K	35,20 €
Övrigt	Plåtar, rör, bultar, muttrar, brickor, skruvar, plast, kablar etc.	80,00 €
	Totalt	512,78 €

Då de billigaste befintliga produkterna säljs i Finland kring 1500 € kan man med gott samvete påstå att målet höll. Dock är de befintliga produkterna mer estetiska och har vissa andra funktioner som vi inte har implementerat i vårt program, men som med extra tid även skulle gå att implementera i vårt system. Det som också måste komma ihåg är att vår produkt endast är en prototyp vars huvudsyfte är att fungera och inte att se vacker ut. Om produkten skulle tillverkas för försäljning så skulle självklart utseendet spela en större roll vid planeringen.

Det man dock inte får glömma är de otaliga arbetstimmar som lagts på detta arbete. Om man räknar med dessa arbetstimmar och arbetstimmar som kommer att läggas på finjustering och testkörning av arbetet ännu efter att det officiella examensarbetet avslutats så kan det konstateras att det helt klart inte är värt att bygga systemet själv bara för att komma billigare undan.

8. SLUTSATS

Det som kan konstateras i efterhand är att eftersom vintern varit kall, så har isläget varit som det varit. Därmed har vi inte haft möjlighet att utföra praktiska experiment till sjöss i den utsträckning som vi önskat för att få ett system fritt från småfel. Systemet fungerade dock förvånansvärt bra med tanke på hur lite vi haft möjlighet till att testa olika lösningar. Det krävs dock en del finslipning och experiment ännu, dessa experiment och finslipningar kommer dock ske efter examination.

För övrigt är vi riktigt nöjda med det vi åstadkommit på den tid som planerats för slutarbetet. Vi är realistiska och förstår att andra likvärdiga produkter som finns på marknaden är planerade och byggda i företag med flera anställda som haft helt andra resurser och verktyg till befogande än vad vi haft. Därmed vore det orimligt att förvänta sig en helt perfekt produkt på denna tid.

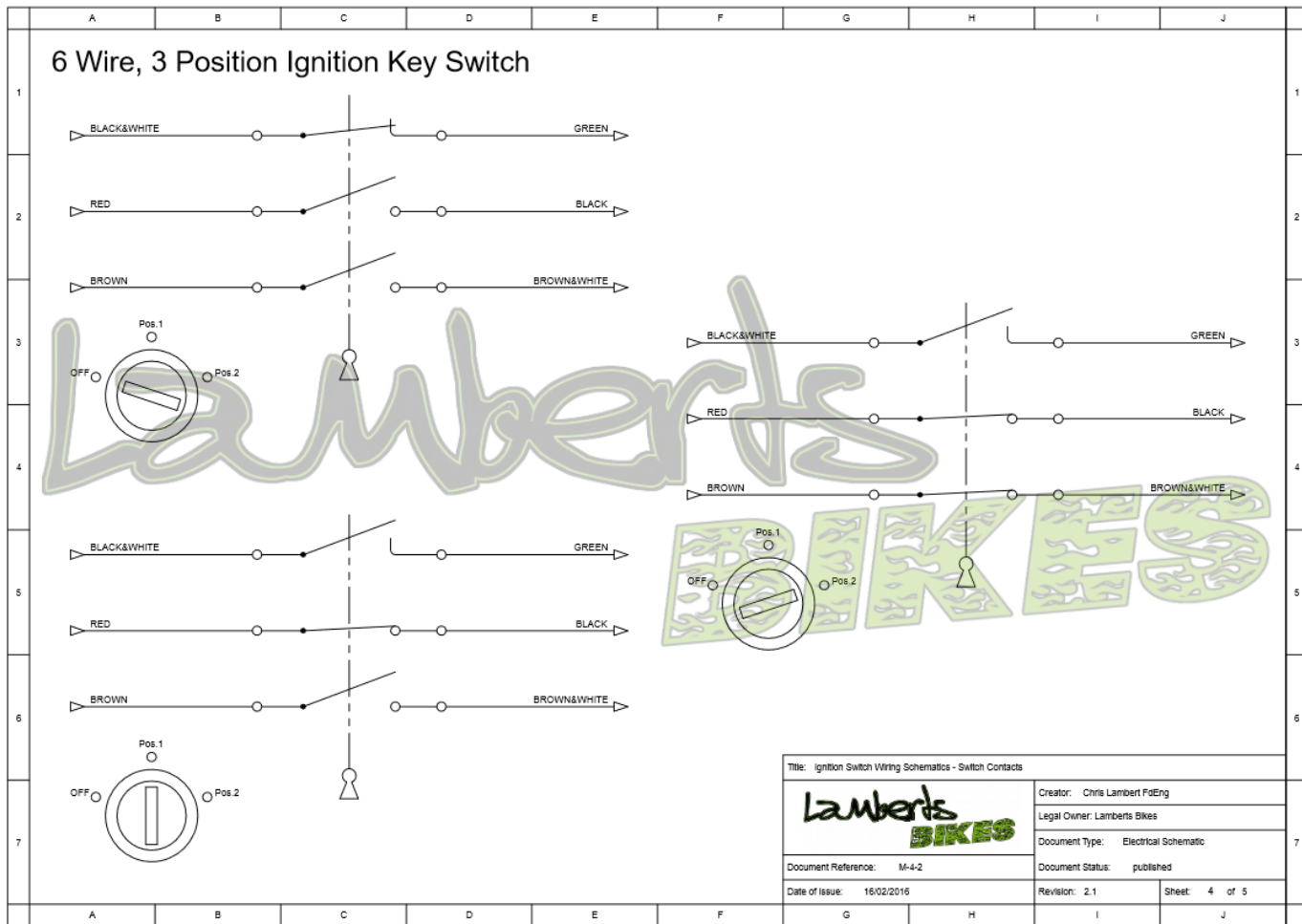
KÄLLFÖRTECKNING

- Adafruit. (2016). Adafruit Ultimate GPS. *Adafruit Industries, Learn. Adafruit. Com*. Retrieved from <https://www.adafruit.com/product/746>
- AMCI. (2018). Stepper vs Servo. Retrieved 2018, from <https://www.amci.com/industrial-automation-resources/plc-automation-tutorials/stepper-vs-servo/>
- Arduino. (2018). ARDUINO DUE. Retrieved 2018, from <https://store.arduino.cc/arduino-due>
- Burris, M. (2018). Stepper Motors vs. Servo Motors - Selecting a Motor. Retrieved 2018, from <https://www.lifewire.com/stepper-motor-vs-servo-motors-selecting-a-motor-818841>
- Clarke, D. (2016). *ResponsiveAnalogRead*. Github. Retrieved from <https://github.com/dxinteractive/ResponsiveAnalogRead>
- Distrelec Group Inc. (2018). Mikro-ohjainkortti, Due, A000062, Atmel SAM3X8E ARM Cortex-M3 CPU, A000062, Arduino. Retrieved 2018, from <https://www.elfadistrelec.fi/fi/mikro-ohjainkortti-due-a000062-atmel-sam3x8e-arm-cortex-m3-cpu-arduino-a000062/p/11038918?q=arduino+due&page=2&origPos=3&origPageSize=50&simi=88.63>
- GlobalTop Technology Inc. (2011). FGPMMPA6H GPS Standalone Module Data Sheet. Retrieved 2018, from <https://cdn-shop.adafruit.com/datasheets/GlobalTop-FGPMMPA6H-Datasheet-V0A.pdf>
- HobbyTronics. (2018). CMPS11 - Tilt Compensated Compass Module. Retrieved 2018, from <http://www.hobbytronics.co.uk/cms11-tilt-compass>
- Hoffman, C. (2016). How GPS Actually Works. Retrieved 2018, from <https://www.howtogeek.com/137862/htg-explains-how-gps-actually-works/>
- Kalaff, L. (2014). How to Implement a MOSFET with a Gate Driver. Retrieved from <http://www.egr.msu.edu/classes/ece480/capstone/fall14/group08/Documents/How%20to%20Implement%20a%20MOSFET%20with%20a%20Gate%20Driver-1.pdf>
- KiCad. (2018). KiCad. Retrieved from <https://sv.wikipedia.org/wiki/Kicad>
- Lackey, B. (2017). Servo Motor vs Stepper Motor: Which is right for your application? Retrieved 2018, from <https://www.motionsolutions.com/servo-motor-vs-stepper-motor-right-application/>

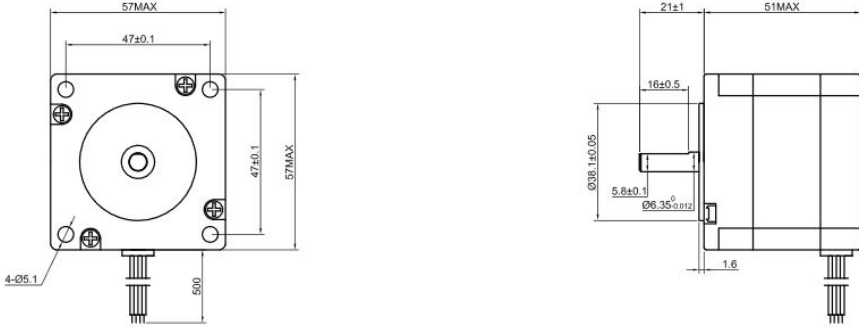
- Lambert, C. (2016). 6 Wire Motorcycle Ignition Switch Wiring Diagram. Retrieved 2018, from <https://lambertsbikes.com/motorcycle-parts/parts-data-library/motorcycle-ignition-switch/6-wire-motorcycle-ignition-switch-wiring-diagram/>
- MiTAC Digital Technology Corp. (2018). What is trilateration? Retrieved 2018, from <http://www.mio.com/technology-trilateration.htm>
- National Instruments Corporation. (2016). What is a Pulse Width Modulation (PWM) Signal and What is it Used For? Retrieved from <http://digital.ni.com/public.nsf/allkb/2AC662081E01E1AE86257F5D00011B6D>
- Navimare. (2018). Minn kota Endura C2 30 Sähköerämoottori. Retrieved 2018, from https://www.navimare.fi/product_details.php?p=28
- Saud, G. (2015). Global positioning system (gps). Retrieved 2018, from <https://www.slideshare.net/gokulsaud/global-positioning-system-gps-51942739>
- Stanley, G. (2017). Exponential Filter. Retrieved 2018, from <https://gregstanleyandassociates.com/whitepapers/FaultDiagnosis/Filtering/Exponential-Filter/exponential-filter.htm>
- Stepperonline. (2018). Nema 23 Bipolar 0.9deg 0.9Nm (127.5oz.in) 0.38A 12.1V 57x57x51mm 4 Wires. Retrieved 2018, from <https://www.omc-stepperonline.com/hybrid-stepper-motor/nema-23-bipolar-09deg-09nm-1275ozin-038a-121v-57x57x51mm-4-wires-23hm20-0384s.html>
- Torrens, R. (2011). 4QD-TEC: Electronics Circuits Reference Archive PWM speed control. Retrieved from <http://www.4qdtec.com/pwm-01.html>
- Veness, C. (2017). Calculate distance, bearing and more between Latitude/Longitude points. Retrieved 2018, from <https://www.movable-type.co.uk/scripts/latlong.html>
- Wikipedia. (2018, March 22). Haversine formula. Retrieved April 23, 2018, from https://en.wikipedia.org/w/index.php?title=Haversine_formula&oldid=831914278

BILAGOR

Bilaga 1 - Ritning på nyckelströmställaren (Lambert, 2016).



Bilaga 2 - Stegmotorns datablad & vridmomentkurva (Stepperonline, 2018).

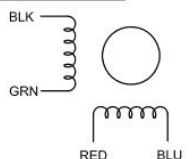



SPECIFICATION	CONNECTION	BIPOLAR
VOTAGE(VOC)		12.10
AMPS/PHASE		0.38
RESISTANCE/PHASE(Ohms)@25°C		32.00±10%
INDUCTANCE/PHASE(mH)@1KHz		110.00±20%
HOLDING TORQUE(Nm)[lb-in]		0.90[7.97]
STEP ANGLE(°)		0.90
STEP ACCURACY(NON-ACCUM)		±5.00%
ROTOR INERTIA(g-cm²)		240.00
WEIGHT(Kg)[lb]		0.60[1.32]
TEMPERATURE RISE:MAX.80°C (MOTOR STANDSTILL;FOR 2PHASE ENERGIZED)		
AMBIENT TEMPERATURE -10°C~50°C[14°F~122°F]		
INSULATION RESISTANCE 100 Mohm (UNDER NORMAL TEMPERATURE AND HUMIDITY)		
INSULATION CLASS B 130°C[266°F]		
DIELECTRIC STRENGTH 500VAC FOR 1MIN.(BETWEEN THE MOTOR COILS AND THE MOTOR CASE)		
AMBIENT HUMIDITY MAX.85%(NO CONDENSATION)		

TYPE OF CONNECTION (EXTERN)		MOTOR	
PIN NO	BIPOLAR	LEADS	WINDING
1	A —	BLK	A
2	A\ —	GRN	
3	B —	RED	B
4	B\ —	BLU	

FULL STEP 2 PHASE-Ex. WHEN FACING MOUNTING END (X)

STEP	A	B	A\	B\	
1	+	+	-	-	CCW
2	-	+	+	-	↓
3	-	-	+	+	
4	+	-	-	+	CW





APVD	
CHKD	
DRN	
SCALE	SIGNATURE DATE

STEPPER MOTOR

23HM20-0384S

23HM20-0384S PULL OUT TORQUE (0.38A/PHASE 24V CONSTANT CURRENT FULL STEP)

