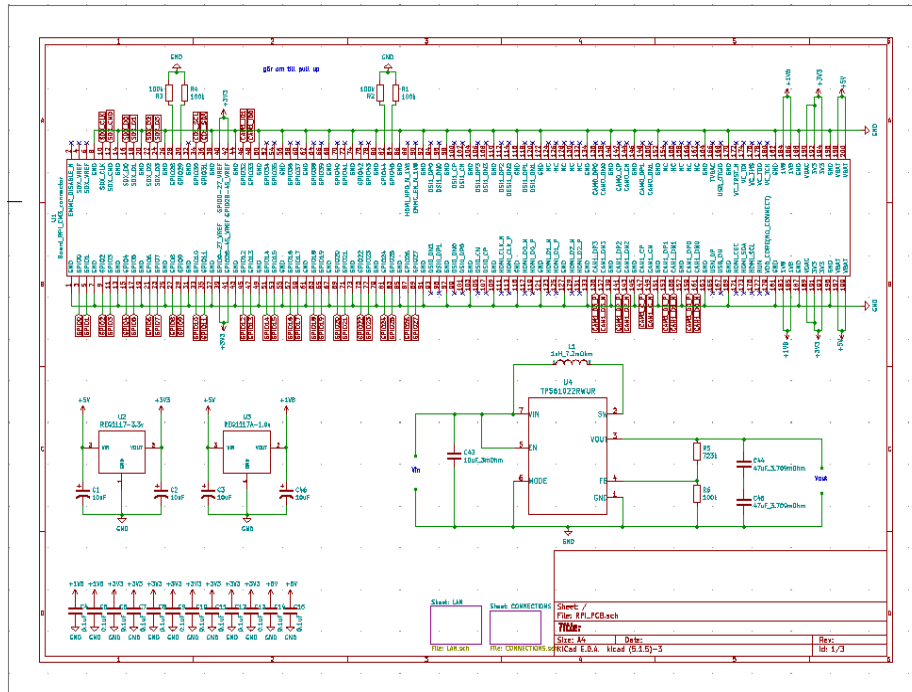




# CHALMERS



## Kretskort för drönare

Examensarbete inom högskoleingenjörprogrammet Elektroteknik

ZACK VESTER

Institutionen för Elektroteknik  
CHALMERS TEKNISKA HÖGSKOLA  
Göteborg, Sverige 2020

EXAMENSARBETE INOM ELEKTROTEKNIK 2020  
Kretskort för drönare

ZACK VESTER



Institutionen för Elektroteknik  
CHALMERS TEKNISKA HÖGSKOLA  
Göteborg, Sverige 2020

Kretskort för drönare  
ZACK VESTER  
© ZACK VESTER, 2020.

Handledare: Fredrik Falkman, Sjöräddningssällskapet  
Examinator: Thomas Rylander

EXAMENSARBETE INOM ELEKTROTEKNIK 2020  
Institutionen för Elektroteknik  
Chalmers tekniska högskola  
SE-412 96 Göteborg  
Telefon: +46 (0)31-772 1000

## **Sammanfattning**

Denna rapport beskriver design och konstruktion av ett kretskort till en fastvingedronare som ska användas av Sjöräddningssällskapet för sjöräddningsinsatser. När ett larm kommer till sjöräddningen ska en drönare skickas ut till olyckan. Drönaren ska sedan cirkulera över olycksplatsen och med kamera skicka information till sjöräddningsstationen så sjöräddarna kan förbereda sig bättre för räddningsinsatsen. Kretskortets kretsscheman och utformning kommer att designas i programmet KiCAD utifrån Sjöräddnings-sällskapets krav för drönare. Designen skickas sedan till ett företag för tillverkning.

## **Abstract**

This report describes the design and construction of a printed circuit board (PCB) for a fixed winged drone for the Sjöräddningssällskapet to use in their sea rescues. When a rescue alarm is received by Sjöräddningssällskapet, a drone is going to get sent out to the accident. The drone is then supposed to circulate above the area of the accident and send video feedback of the accident to the sea rescue crew so they can prepare better for the rescue operation. The PCB:s schematics and layout is going to be designed in Ki-CAD to match the specification given by Sjöräddningssällskapet. The design is then sent to a company for manufacturing.

# Innehåll

<b>1</b>	<b>Inledning</b>	<b>2</b>
1.1	Bakgrund . . . . .	2
1.2	Syfte och mål . . . . .	2
1.3	Avgränsningar . . . . .	3
1.4	Hållbar utveckling och etik . . . . .	3
<b>2</b>	<b>Systemdesign och konstruktion</b>	<b>4</b>
2.1	Övergripande system . . . . .	4
2.2	Hårdvara . . . . .	5
2.3	Kretskort . . . . .	7
<b>3</b>	<b>Resultat</b>	<b>15</b>
3.1	Konfigurering . . . . .	15
3.2	Test av lufthastigheten . . . . .	15
3.3	Test av gyroskopet . . . . .	17
3.4	Test av GPS, barometer och kamera . . . . .	19
<b>4</b>	<b>Diskussion</b>	<b>24</b>
<b>5</b>	<b>Slutsats</b>	<b>25</b>

## Förkortningar

GPS: Global Positioning System

USB: Universal Serial Bus

HDMI: High-Definition Multimedia Interface

RPI: Raspberry Pi Compute Module 3+ Lite

CAM: Computed-Aided Manufacturing

EDA: Electronic Development Environment

CSI: Camera Serial Interface

DSI: Display Serial Interface

I2C: Inter-Integrated Circuit

GND: Jord

V: Volt

A: Ampère

Tx signal: Sändarsignal

Rx signal: Mottagarsignal

# 1 Inledning

## 1.1 Bakgrund

Olyckor till sjöss förkommer året runt men det är extra många olyckor under månaderna juni och juli med runt 200, respektive 300 olyckor [1]. Flest olyckor involverar fritidsbåtar som råkar ut för propellerhaveri och grundstötningar, men även olyckor med brand, drunkning och sjuktransport ifrån fartyg inträffar också. Sjöräddningssällskapet är ett ideellt bolag som arbetar med sjöräddning i samarbete med Larmcentralen. De har 72 räddningsstationer, 2200 sjöräddare och 230 räddningsbåtar [2]. Inför en sjöräddning vill Sjöräddningssällskapet ha så mycket information som möjligt om olyckan för att kunna förbereda sin räddningsinsats så bra som möjligt. Till exempel kan en räddningsinsats kräva speciell utrustning eller sjukvårdare. Viktig information för förberedelser skulle man kunna få från videoströmning från en drönaren. Drönarens Global Positioning System (GPS) kan även hjälpa med att bestämma den korrekta positionen för olyckan, och om den olycksdrabbade båten rör sig så vet man i vilken riktning. Drönarna ska vara placerad utmed kusten för att snabbt kunna skickas ut vid en olycka. Detta ger en väldigt kort responstid på olyckan. Dessutom kan drönarna styras på distans. Man får även möjligheten att flyga väldigt nära farliga situationer utan att utsätta personal för risker.

## 1.2 Syfte och mål

Målet med detta projekt är att designa och testa om ett kretskort baserat på en Raspberry Pi Compute Module 3+ Lite (RPI) är en bra lösning till Sjöräddningssällskapet drönare. Flera av komponenterna som ska finnas i drönaren ska även implementeras på kortet för att minska utrustningens storlek och vikt i drönaren. Detta ska leda till ett kompakt system som ska kunna få plats i Sjöräddningssällskapet drönare och kunna driva hela systemet. Målet är att konstruera ett system som ska kunna bestämma sin egen position och kunna navigera från hemmabasen till olyckan med hjälp av autopiloten. Systemet ska även kunna strömma video från en kamera samt mäta drönarens hastighet, altitud och position. Systemet behöver följande komponenter:

- RPI

- Autopilot
- Strömmodul
- 4G-modem
- GPS
- Motorkontroller
- Kamera
- Servos för vingar
- Lampor
- Batterier
- Lufthastighetssensor

### **1.3 Avgränsningar**

Det konstruerade systemet kommer att testas med alla komponenter installerade men kommer ej att implementeras och testas i en flygande drönare. Kretskortet kommer att designas i KiCAD, vilket är ett program med öppen källkod. Majoriteten av komponenterna kommer att vara ytmonterade för att minska vikten och storleken på kretskortet så mycket som möjligt. Systemet kommer ej att testas med trådlös kommunikation mellan systemet och räddningsstationen. Lödning av komponenterna kommer att utföras av samma företag som tillverkar kretskortet.

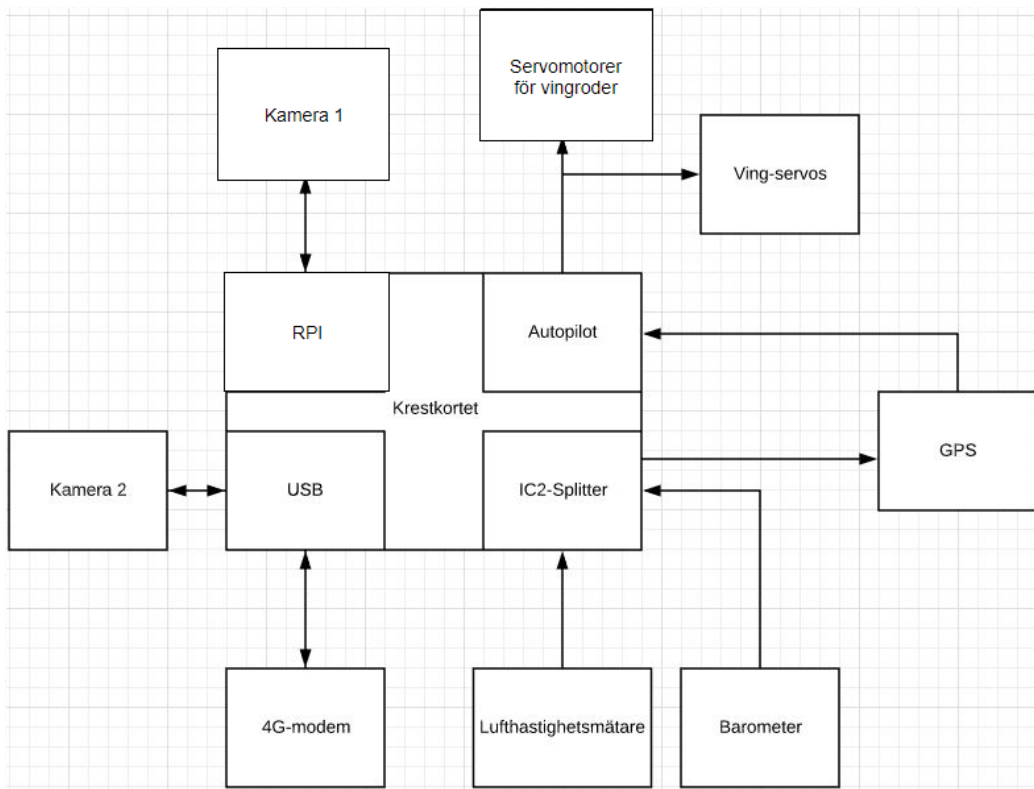
### **1.4 Hållbar utveckling och etik**

Eftersom uttryckningar kan planeras bättre kan man spara på resurser så som bränsle, utrustning och personal. Videoövervakning med hjälp av drönare är en etiskt komplicerad fråga, där nyttan med räddninginsatsen måste vägas mot möjlig intergretskränkning. Om personer kan identifieras på bilderna så faller automatiskt ett personuppgiftsansvar på den som filmar. Det är rimligt att videoövervakning inte används till och från olycksplatsen, utan endast vid olycksplatsen.

## 2 Systemdesign och konstruktion

### 2.1 Övergripande system

Systemet är konstruerat så att RPI:n utgör själva kärnan i systemet. Det är autopiloten som sköter all kommunikation mellan sensorerna och servomotorerna. Autopiloten kommunicerar också med RPI:n som har autopilotens programvara installerad. Man styr autopiloten genom RPI:n. Kamerorna och 4G-modemet är kopplade direkt till RPI:n genom Universal Serial Bus (USB) och Camera Serial Interface (CSI). Lufthastighetsensorn och barometern är kopplade till en I2C-splitter på kortet som är kopplad till GPS:n som i sin tur är kopplad till autopiloten. Illustration av systemet visas i figuren nedan.



Figur 1: Blockschema på hur systemet är uppbyggt och pilar på hur signalerna skickas mellan komponenterna.

## 2.2 Hårdvara

En av huvudkomponent för projektet är en RPI som kan ses som den centrala enheten i systemet. RPI:n är en minidator, men den har även ett I/O-gränssnitt. Detta ger funktionen att styra motorer och reläer samt få in data från givare och sensorer [3]. Det speciella med RPI:n är att hela datorn är ungefär  $8\text{cm} \times 4\text{cm} \times 5\text{mm}$ , men detta betyder också att man själv måste skapa ett kretskort med alla funktioner och komponenter man behöver. RPI:n har en BCM2837B0 processor med 1 GB ramminne. RPI:n har 200 stift med möjlighet att koppla in en USB, en High-Definition Multimedia Interface (HDMI), två CSI-kameror och två Display Serial Interface (DSI) displayer. För användning rekommenderas det att använda Linux eller Raspberry Pi Operating System (OS) som operativsystem. Det går även att använda Windows OS men bara för att bygga egna projekt. Detta beror på att Windows inte har förmågan att köra vanliga programvaror, som programvaran för autopiloten, eller ett Graphical User Interface (GUI). RPI:n har inte något internt minne, utan bara möjlighet att använda Secure Digital (SD) eller mikro-SD kort för lagring.

RPI:n har endast en USB men kretskortet behöver två stycken. Därför används en mikrokontroller LAN9512 vilket är en höghastighets-USB 2.0 hubb som förgrenar en USB till två. En USB-hubb har alltid en uppströmsport och två eller fler nedströmsportar. Uppströmsporten är kopplad från hubben till masterenhet och nedströmsportarna kopplar slavenheter till hubben. En hubb kan vara en bussförsörd hubb (passiv hubb), vilket innebär att hubben drar all ström från värdens USB. En självförsörjande hubb (aktiv hubb) är en hubb som får sin ström från en extern källa och kan då tillföra full strömstyrka till fler portar, upp till 500 mA [4].

Videokameran är en Raspberry Pi Kameramodul v2 som kopplas in i RPI:n genom CSI-porten.

Autopiloten är en mRo Control Zero F7 som monteras på kortet. Autopilotens signaler dras sedan ut på kortet till de olika kontakterna för sensorerna och servorna.

Kretskortet försörjs med 5 volt (V) från autopiloten vilket försörjs från strömmodulen från batterierna. Flera komponenter på kortet behöver 3.3 V eller 1.8 V, vilket implementeras med hjälp av två linjära regulatorer REG1117 som är baserade på stepdown-omriktare.

Linjära regulatorer använder sig av bipolära transistorer eller fälteffekttransistorer. En linjär spänningsregulator höjer och sänker utspänningen jäm-

fört med inspänningen. Linjära regulatorer valdes på grund av att skillnaden mellan spänningarna är så liten och de är även väldigt kompakta och lätta att implementera jämfört med en växlande regulator. Man får även lägre brusspänning av linjära regulatorer [5].

Man har valt att använda en lufttryckssensor för att mäta hastigheten istället för en GPS. Denna sensor har ett så kallat Pitot-rör längst fram på planet som är riktat mot luftströmmen när man flyger. Autopiloten räknar sedan ut hastigheten beroende på övertrycket som uppstår i sensorn [6].

Höjden över havet beräknas med hjälp av utsignalen från en barometer vilken mäter lufttrycket. Ett annat alternativ är att använda en nedåtriktad radar. Detta görs genom att skicka en elektromagnetisk våg från drönaren och mäta tiden det tar för vågen att färdas till marken och tillbaka till drönaren. Denna tid kan sedan användas för att räkna ut hur högt upp drönaren är ovanför marken [7].

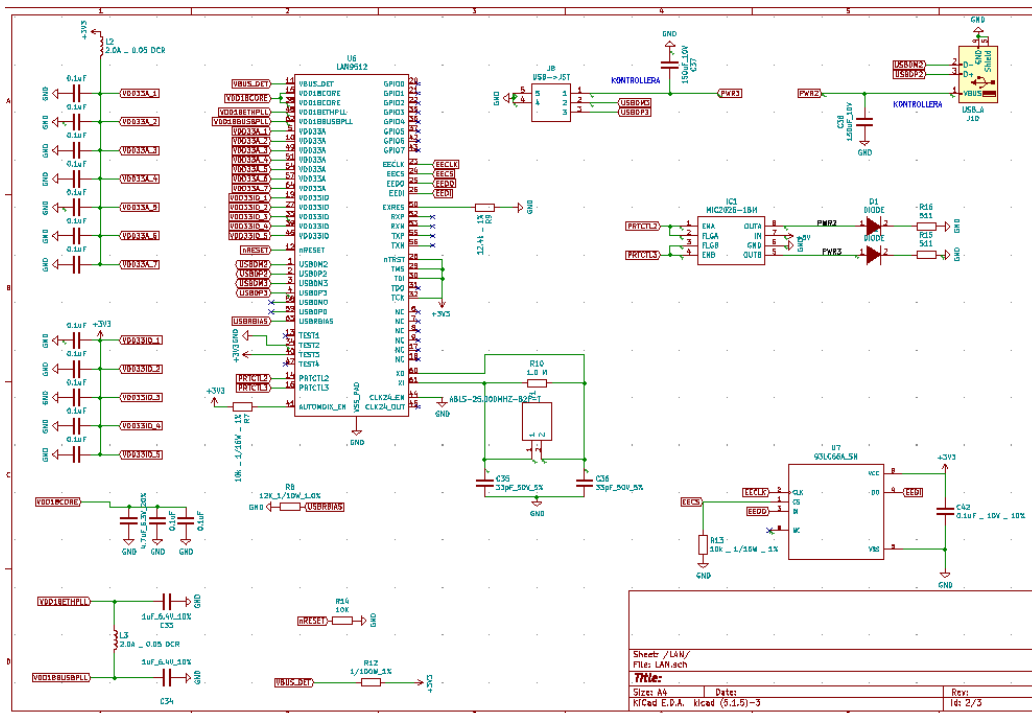
Barometern, lufttryckssensorn och GPS:n skickar informationen till autopiloten. Informationen skickas sedan till programvaran QGroundControl som körs på RPI:n. Autopiloten skickar signaler till propellermotorn och servomotorerna för att flyga drönaren.

Flyghastighetssensorn, barometern och GPS:n ska alla kommunicera med autopiloten. För att få det att fungera så används en I2C splitter. En I2C buss behöver två signaler: klocksignal och datasignal. Klocksignalen är alltid genererad av bussvärden. Slavkomponenterna kan tvinga klocksignalen låg för att hindra bussvärden att skicka mer information, eller för att få mer tid att förbereda data som ska skickas till värden. I2C använder sig av öppen drain vilket innebär att den kan göra signalerna låga men ej höga. Därför behövs det en pull-up resistor för varje signal för att återställa signalen när den inte används [8].

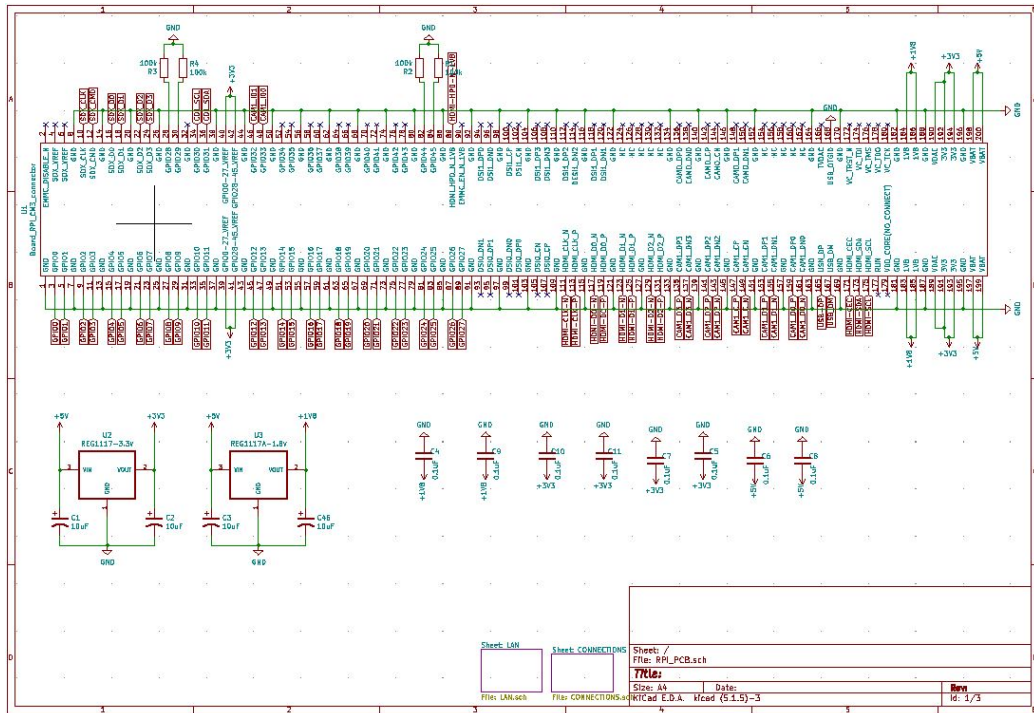
## 2.3 Kretskort

KiCAD kan klassas som Electronic Design Automation (EDA) baserad på öppen källkod där man designar kretsscheman och kretskortutformning och generering för filerna för Computed-Aided Manufacturing (CAM) [9].

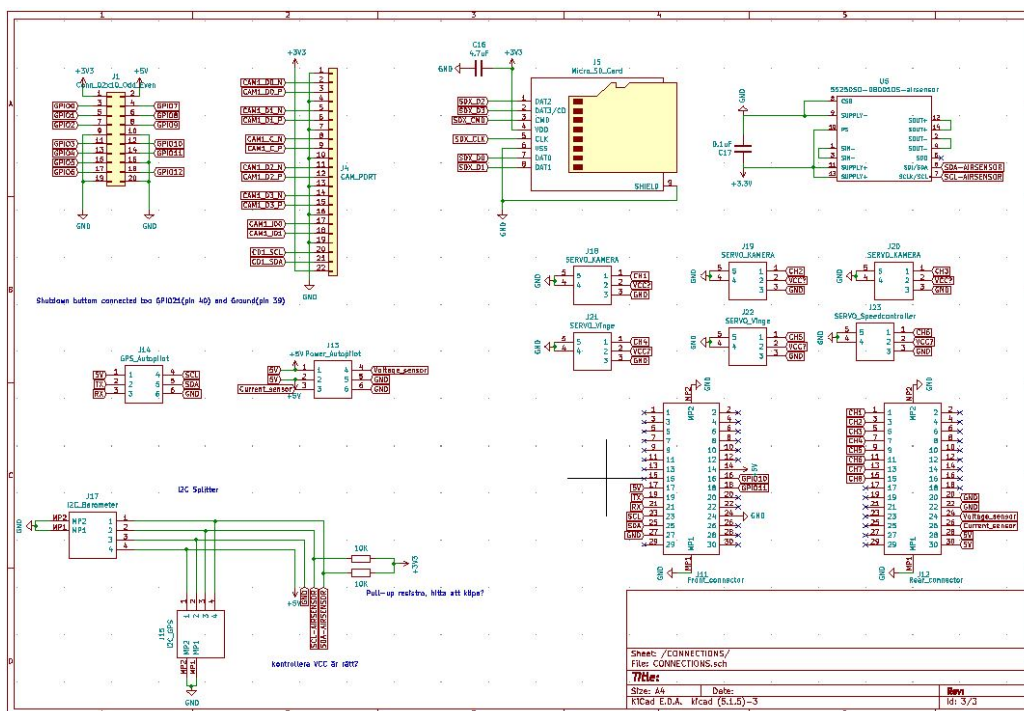
För att skapa kretsscheman så behövs kretsinformationen i KiCAD:s bibliotek för alla komponenter man ska använda. De vanliga komponenterna som resistorer och kondensatorer finns oftast redan i biblioteket. Finns inte komponenterna så måste man göra egna ritningar i programmet med hjälp av informationen man får i databladet. Vissa komponenter kan hämtas online och läggas till i biblioteket. I kretsschemat ska alla komponenter kopplas ihop enligt önskad konstruktion. Informationen om hur detta ska göras finns oftast i databladet för komponenten. I figur 2-4 nedan visas kretsscheman för systemet. Så som figur 2-4 visar så används det många kondensatorer och detta är för att förhindra spänningsfall vilket kan skada de andra komponenterna.



Figur 2: Kretsschema med USB-hubben och tillhörande komponenter.



Figur 3: Krettschema för RPi:n och spänningsregulatorerna.



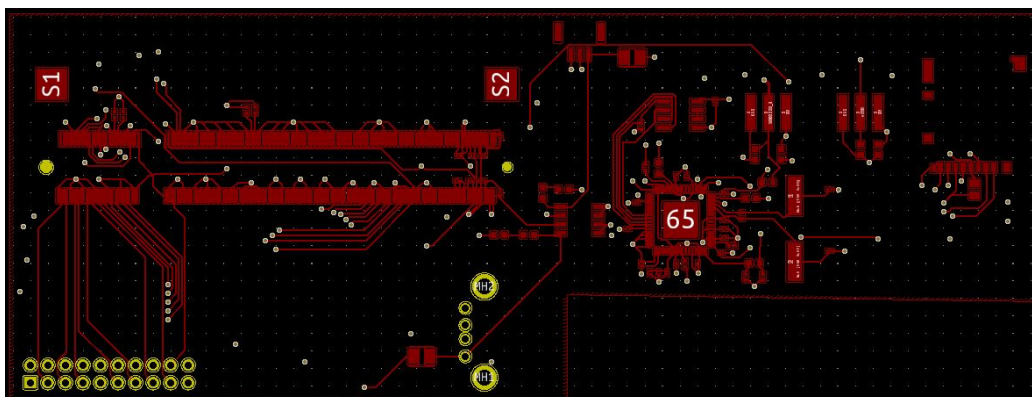
Figur 4: Kretsschema för de olika kontakterna för komponenterna som ska kopplas in.

Kretskortet är designat runt RPI:n som är monterad med en DDR2 SO-DIMM RAM kontakt, vilket är samma kontakt som används för minnen av typen DDR2. RPI:n har 200 olika stift för att kunna tillhandahålla de alla olika funktioner som den erbjuder. Stiften GPIO 0-45 har funktionen att vara antingen in- eller ut-portar eller ett avbrott. Alla GPIO:s förutom GPIO 28, 29, 44, 45 har svaga pull-up eller pull-down resistorer aktiverade när enheten startas. Därför kopplas det pull-down resistorer till dessa fyra stift för att se till de aldrig hamnar i ett flytande tillstånd vid uppstart. GPIO 0-27 drivs av spänningsbank 0 och GPIO 28-45 drivs av spänningsbank 1. Dessa spänningsbanker kan ge en spänning intervallet 1.8-3.3 V beroende på behov.

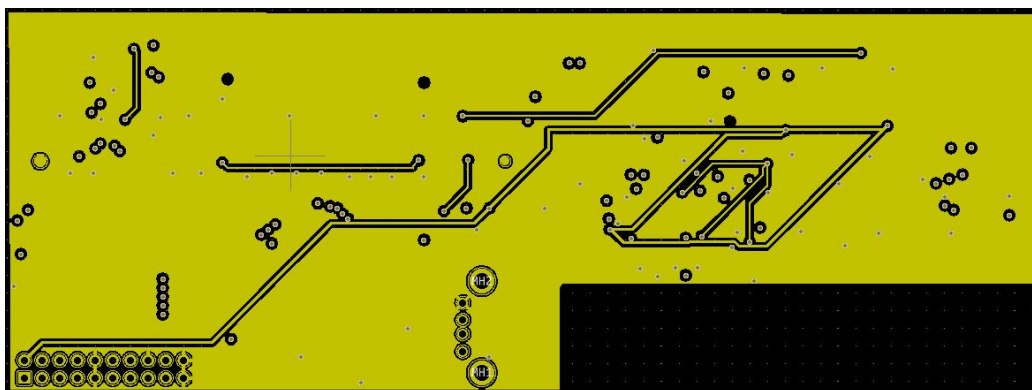
Spänningen och strömmen som krävs för att driva kortet fås från autopiloten som levererar 5 V och 2.5 Ampère (A). Som det nämndes tidigare i rapporten så kräver även kortet 3.3 V och 1.8 V för att fungera. Dessa spänningsnivåer tillhandahålls av två regulatorer.

När kretsschemat för kortet är klart så är nästa steg att ta fram krets-

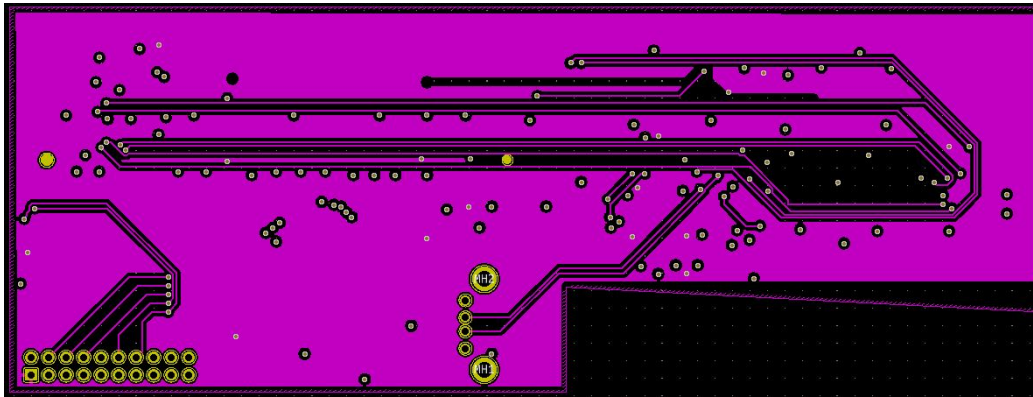
designen för komponentplaceringen. Kretsdesignen innehåller också dragning av de ledare som kopplar ihop komponenterna. Kretsdesignen visas i figur 5-8.



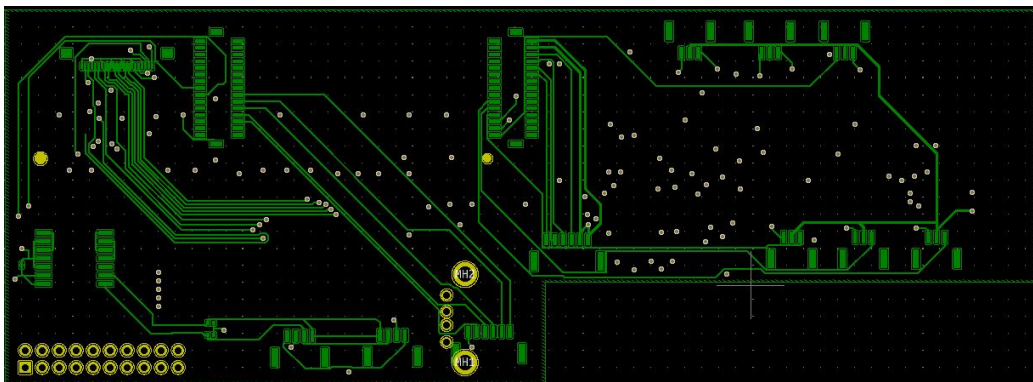
Figur 5: Översta lagret med RPI, USB-hubb och regulatorerna.



Figur 6: GND-lagret.



Figur 7: 3.3V-lagret.



Figur 8: Understa lagret med det olika kontakterna.

Kortet har ett plan för jord och ett plan för matningsspännig och två plan för komponenterna. Det är bra att ha ett eget plan för jord eftersom det ger viss skärmning mot elektromagnetiska fält som kan störa signaler.

För att man ska kunna få fram en utformning måste det även finnas vad som kallas footprints för varje komponent. Footprints är en symbol med de exakta måtten för komponenten, och detta måste finnas så att ledarna är rätt placerade när kortet ska skapas.

När man drar ledarna använder man alla fyra lager för att inte korsa ledarna med varandra. Vias är ett hål mellan dessa fyra lager som gör att man kan leda en signal från ett lager till ett annat. Spänningsnivån 3.3V och jord användes väldigt ofta över hela kortet och kan orsaka korsningar av ledare och

resultera i väldigt komplicerat dragning av ledare. Eftersom ett fyra-lagerkort användes så gjordes ena lagret i mitten till ett 3.3V-lager. Detta innebär att spänningen 3.3V är direkt tillgänglig över hela detta kopparlager. Istället för att dra en ledare från 3.3V regulatoren så räcker det att man sätter en vias ner till 3.3V lagret precis vid komponenten som behöver 3.3V och behöver då bara dra spåren en väldigt kort sträcka. Samma sak gjordes med GND för det andra lagret i mitten. För komponenterna som inte är ytmonterade, utan stiften går igenom hela kortet, kan man koppla dessa stift direkt till önskat lager utan vias och utan spår. Man kan även dra signalledare i 3.3V-lagret och jordlagret, men dessa ledare avgränsas från de delar av kopparlagrena som är tilldelade 3.3V eller jord. När man drar spår i 3.3V-lagret och jordlagret måste man vara försiktig för att förhindra att komponenter eventuellt kan bli avgränsade eller isolerade ifrån dessa potentialnivåer.

Spåren för USB-signalerna DP och DM måste vara parallellt dragna. Detta för att informationen mellan USB:ens två signaler ska anlända samtidigt. Är ett spår längre än det andra kommer den informationen att dröja eftersom tiden för en signal att propagera längs spåret från ena änden till den andra är lika med spårets längd delat med signalens utbredningshastighet. När man ritat ut ledarna är det också viktigt att tänka på att undvika 90-graderskrökar eftersom 90-graderskrökar reflekterar tillbaka en betydande del av de högfrekventa signalerna. Sådana reflektioner kan reduceras betydligt genom att till exempel kombinera två på varandra följande 45-graderskrökar så som figur 9 visar, eller andra mjukare kurvformer.



Figur 9: Två olika krökar: en 90-raderskrök visas till vänster; och två på varandra följande 45-graderskrökar visas till höger [9].

Det är viktigt när man drar spåren mellan komponenterna att de är tillräckligt breda för den ström som ska flyta genom ledarna. Resistansen  $R$  för likström ges av  $R = l/(\sigma A)$  där  $l$  är ledarens längd,  $\sigma$  är ledningsförmågan och  $A$  är den tvärsnittsarea som strömmen flyter genom. För en ledare med ett rektangulärt tvärsnitt ges dess area av  $A = bh$  där  $b$  är tvärsnittets bredd och  $h$  är dess höjd. Ju bredare spår desto mindre effekt  $P$  och värme utvecklas enligt formeln  $P = RI^2$  givet en konstanten ström  $I$ . Den utvecklade värmen transporteras från ledaren till den omgivande luften och substratet vilket kyler ledaren. Då värmeutvecklingen vid kontinuerlig (och konstant) drift balanseras av kylning blir temperaturen konstant i tiden, efter att transienterna har klingat av. Det är viktigt att temperaturen inte blir för hög och för detta använder man en kalkylator [10] som ger en bredd som spåret minst måste ha.

När kortet är färdigtställt så måste man spara ritningen i ett speciellt format som kallas Gerber-format. Detta är ett format som beskriver ritningarna för de automatiska maskinerna som sköter tillverkningen av kretskortet. Dessa maskiner fungerar på ett likande sätt som 3D-skrivare. Om man även vill att de som tillverkar kretskortet ska löda dit alla komponenter så måste en Bill of Materials (BOM) fil tillhandahållas. Det är en fil med alla komponenter

ter på kortet, Manufacturer Number (MNF) och antal av varje komponent. Bild på BOM-filen för det här projektet visas nedan.

ref	value	MNF number	Quantity
R1, R2, R3, R4	100KOHM	RC0402FR-07100KL	4
C4, C5, C6, C7, C8, C9, C10, C11, C12, C13, C14, C17	0.1uF	GRM033R61A104KE15D	12
C1, C2, C3 ,46	10uF	GRM188R60J106KE47D	4
C16	4,7uF	LMK107BJ475KA-T	1
R5_PULL-UP1, R6_PULL-UP1	10KOHM	CR0603-FX-1002EAS	2
C18, C19, C20, C21, C22, C23, C25, C26, C27, C28 ,C29, C31, C32	0.1uF	GRM033R61A104KE15D	13
C30	4.7uF_6.3V_20%	GRM155R60J475ME47D	1
C33, C34	1uF_6.4V_10%	C0402C105K9PACTU	2
C35, C36	33pF_50V_5%	GRM0335C1H330JA01D	2
R13	10k_1/16W_1%	CRGCQ0402F10K	1
C42	0.1uF_10V_10%	GRM033R61A104KE15D	1
R12	1/100W_1%	MCEE100STR10MHN	1
C37, C38	150uF_10V	TLJB157M010R0500	2
J10	USB	87520-0010BLF	1
J8	USB->JST	SM03B-GHS-TB(LF)(SN)	1
R10	1.0MOHM	RC0402FR-071ML	1
Y1	HC49US	ABLS-25.000MHZ-B2F-T	1
R9	12.4KOHM	AC0402FR-0712K4L	1
R8	12KOHM_1/10W	CRGCQ0603F12K	1
R7	10K_1/16W	CRGCQ0402F10K	1
L2, L3	spole	BLM18PG121SN1D	2
U1	DDR3	1473005-4	1
U2	REG1117-3.3V	REG1117-3.3	1
U3	REG1117A-1.8V	REG1117A-1.8	1
J3	GPIO PINS	832-80-010-10-001101	1
J4	CAM-PORT	54548-2271	1
J5	MICRO-SD	1051620101	1
J11, J12	AUTOPILOT CONNECT	2035663007	2
J13, J14	6-CONNECT	SM06B-GHS-TB(LF)(SN)	2
J8 ,J15, J17	4-CONNECT	SM04B-GHS-TB(LF)(SN)-	3
U5	AIR-SENSOR	5525DSO-DB001DS	1
U6	93LC66A_5N	93LC66A/SN	1
U7	LNA9512	LAN9512-JZX	1
J18, J19, J20, J21, J22, J23	3-CONNECT	SM03B-GHS-TB(LF)(SN)	6
R15, R16	511	MCR03EZPFX5110	2
D1, D2	Diode	SML-D12FWT86	2
IC1	MIC2026-1BM	MIC2026-1BM	1

Figur 10: BOM med komponenternas beteckning, komponentens tillverkningsnummer och hur många av varje komponent som ska användas.

## 3 Resultat

### 3.1 Konfigurering

RPI:n installeras i IO Board V3 och man ställer in den i programmeringsläge. Man kan nu göra modulens första uppstart och börja konfigurera den. Operativsystemet Raspberry Pi OS valdes och installerades på RPI:n. Eftersom ingen skärm kommer vara inkopplad till kortet så behöver man kunna koppla den till en annan dator. Man kan även använda sig av Virtual Network Computing (VNC) vilket är en programvara man laddar ner så man kan styra RPI:n med en annan dator eller en mobilelefon [11].

Om man vill använda sig av en kamera som är ansluten till en kameraport så måste den funktionen även aktiveras för RPI:n innan man kan koppla in en kamera. Detta görs också med hjälp av IO Board V3. Kamerakontakten har fyra stycken kontrollsignaler vilka ges av fyra stycken GPIO-signaler, som i sin tur är programmerbara. Man behöver också ett så kallat device tree, vilket aktiverar GPIO-signalerna så att dessa kopplas till kamerans kontrollsignaler. Man kompilar sedan sitt device tree vilket skriver över standardlistan [11].

För att ta bilder på ett smidigt sätt skapades ett program i Python vilket med en while-loop gjorde att kameran tog en bild per halvminut.

Autopiloten behöver en programvara för att fungera. Här valdes programmet QGroundControll vilket styr kommunikationen mellan autopiloten, sensorer och servorna.

### 3.2 Test av lufthastigheten

Testning av lufthastighetssensorn utfördes genom att använda en bil. Sensorn hölls utanför passagerarsidans sidoruta då bilen kördes. Mätvärdet från lufthastighetsensorn jämfördes sedan med den uppmätta hastigheten bilens mätare visar.



Figur 11: Systemet hölls utanför bilen för att registrera lufthastigheten. Pitot-röret riktas mot bilens körriktning.

Endast hastigheter upp till 30 km/h kunde testas. Resultaten visas i tabellen nedan.

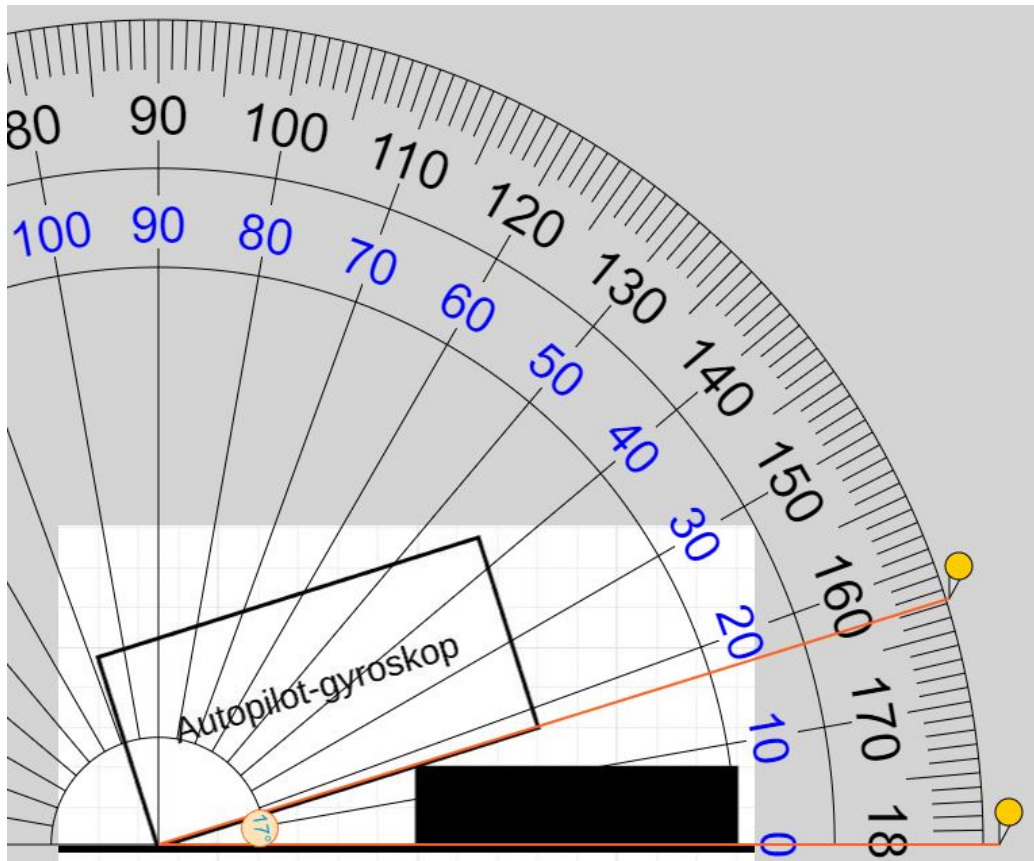
Sensorn (km/h)	Bilen (km/h)	Relativ avvikelse (%)
11.9	10	19
19.8	20	1.0
31.6	30	5.3

Mätvärdena stämmer ganska väl överens och den absoluta avvikelsen är mindre än 2km/h. En tänkbar orsak kan vara att vinden utomhus påverkar hastighetsmätningen. Även aerodynamiken runt bilen kan påverka luftflödet ytterligare och därmed också mätvärdena för sensorn. Troligen är sensorns relativa noggrannhet bättre för högre hastigheter, vilket stöds av de preliminära mätningarna i tabellen ovan.

### 3.3 Test av gyroskopet

Här gjordes en mätning för att testa för gyroskopet i autopiloten. Detta gjordes genom att en kloss lades under ena änden på autopiloten och sedan togs en bild från sidan. Därefter användes en digital vinkelmätare för att uppskatta vinklen och jämföra den med den vinkel som autopiloten visar.

Gyroskop vinkel (grader)	Mätt vinkel (grader)	Relativ avvikelse (%)
4.3	4.1	4.9
9.9	10.4	4.8
14.5	14.4	0.7
20.4	20.7	1.4
26.4	26.0	1.5
31.2	30.6	2.0



Figur 12: Illustration på hur testet med gyroskopet genomfördes. Gyroskopet lades i en vinkel och jämförde den manuella mätningen och vinkeln ifrån autopiloten.

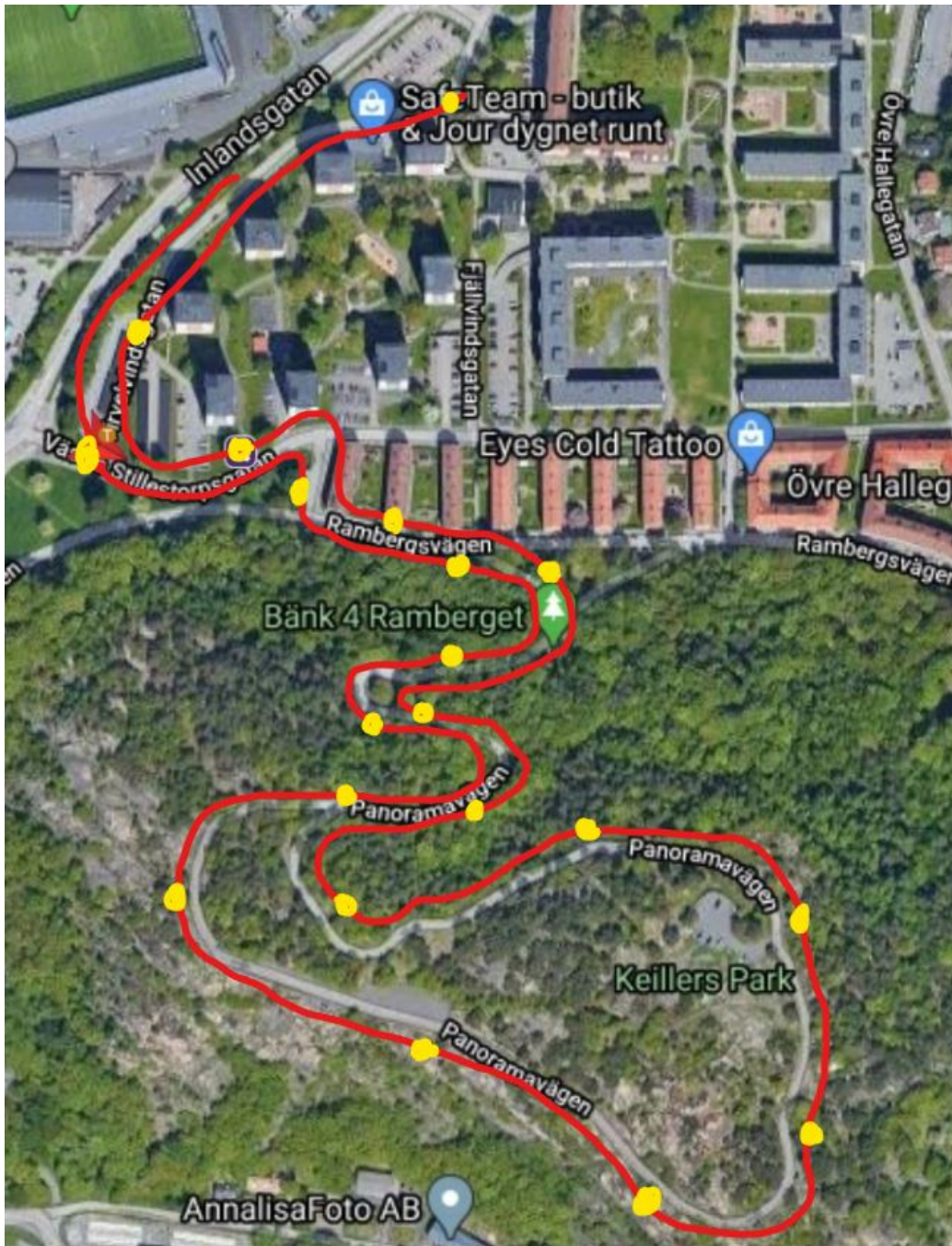
Jämförelsen mellan autopiloten och de manuella mätningarna visar god överensstämmelse med en absolut avvikelse som är mindre än 1 grad för de testade vinklarna. Man kan konstatera att gyroskopet fungerar bra, givet att de relativt avvikelseerna är mindre än 5%

### **3.4 Test av GPS, barometer och kamera**

Detta test gjordes för att testa GPS:n, barometern och kameran. Systemet med komponenterna som testades sattes upp i en bil som visas i figur 13 nedan. Bilen körde i ungefär 20 km/h på en förbestämd rutt på 4.1 km som visas i figur 14 nedan. Under denna rutt så tas en bild med hjälp av kameran och en skärmdump av informationen ifrån autopiloten varje halvminut vilket visas i figur 15. Tabellerna av uppåtlutning och altitud längs med rutten visas i figur 16-17. Efter testet så kontrollerades GPS punkterna genom att gå till varje position som matchar bilderna från kameran och jämföra GPS positionen ifrån mätningen med den verkliga positionen. Under denna kontroll kollades också om vägens lutning var positiv eller negativ, vilket också jämfördes med mätningarna. Mätningarna ifrån GPS:n visar samma position som kontrollmätningarna gjorde och mätningarna för lutningen stämmer med hur vägen ser ut.



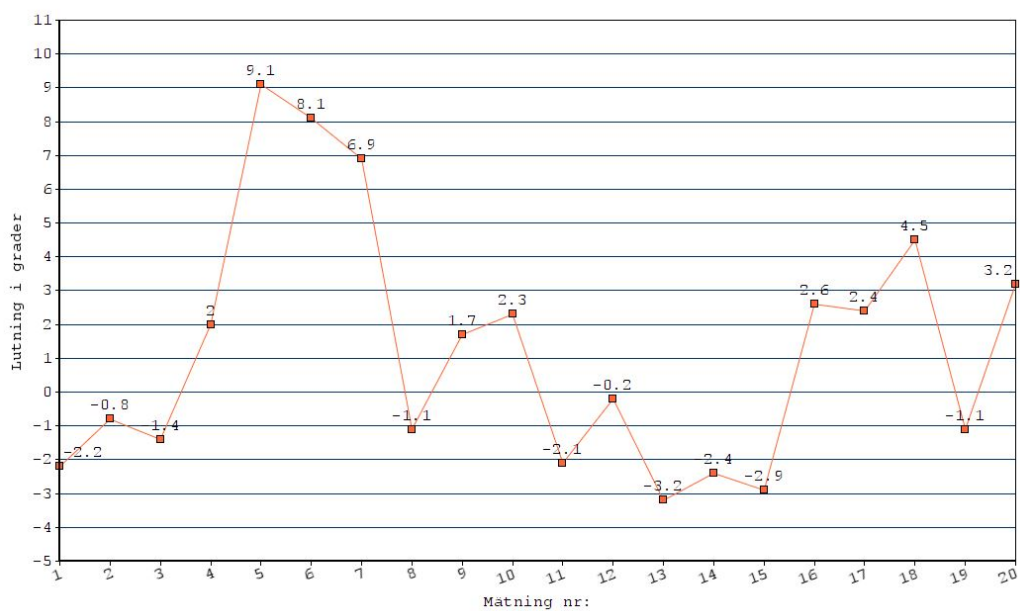
Figur 13: Bild på hur systemet var monterat i bilen.



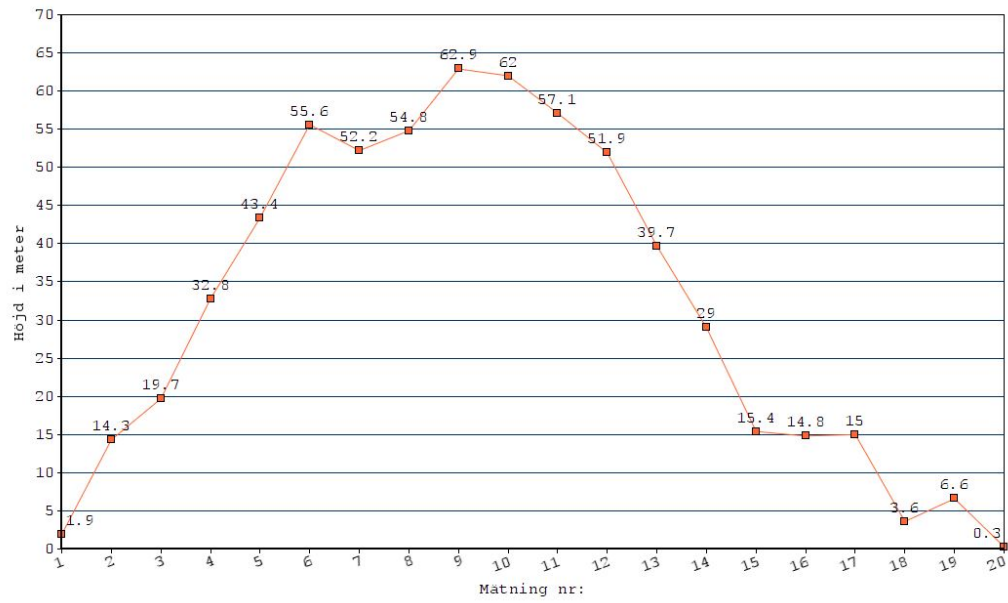
Figur 14: Det röda strecket markerar rundan som kördes med systemet, och de gula punkterna är där mätdata togs ut.



Figur 15: Bild på GPS:n och en bild ifrån kameran vid en specifik mätpunkt.



Figur 16: Mätning av lutningen i grader för systemet. Negativ lutning svarar mot nedförsbacke och positiv lutning svarar mot uppförsbacke.



Figur 17: Mätning av höjden i meter med hjälp av systemet. Höjden är relativ till startpunkten för den körda ruten.

## 4 Diskussion

Barometern mäter lufttrycket vilket används för att bestämma drönarens höjd över havet. Denna mätmetod skulle kunna kompletteras med en nedåtriktad radar, vilken mäter avståndet till marken eller vattnet som finns under drönaren. Detta alternativ ger även drönaren förmågan att detektera föremål under drönaren. En möjlig nackdel kan vara att sådana föremål felaktigt tolkas som ett markeko.

Valet av mikrokontroller för USB:en kunde göras annorlunda eftersom den nu har möjlighet att lägga till en ethernet också. Eftersom detta inte behövdes så skulle en mindre mikrokontroller användas vilket reducerar systemets vikt och pris.

Val av komponenter och var man kunde köpa dessa tog oväntat mycket arbetstid, vilket inte fanns med i planeringen från början eftersom att det inte förväntades vara något problem. Den extra arbetstiden beror på att man inte alltid kan köpa en komponenten om man inte köper en större mängd. Vissa komponenter saknade riktlinjer för hur de skulle kopplas och därmed behövdes extra tid för att läsa på om dessa komponenter för att veta exakt hur de fungerade och att kunna implementera dem i designen.

Tillverkningen av kretskortet försenades avsevärt av företaget på grund av Covid-19, vilket gjorde att jag inte kunde följa min tidsplanering i slutet.

## 5 Slutsats

Den här rapporten beskriver design, konstruktion och testning av ett kretskort till en fastvingedronare, vilken används av Sjöräddningssällskapet vid sjöräddning till havs. Kretskortet ska kunna navigera drönaren till olycksplatsen. Därifrån ska det kunna skicka en videoström till sjöräddningsstationen, där videoströmmen visar olycksplatsen. Kretskortet ska även kunna mäta drönarens position, lufthastighet, lutning och höjd. Kretskortets funktionalitet har testats på land och mätvärdena har jämförts med referensmätningar, där jämförelsen visar god överensstämmelse.

Målet som sattes upp för projektet har uppnåtts med ett fungerande system för en drönare. Man skulle kunna jobba vidare med att optimera kortet så det tar upp mindre plats i drönaren. Integration av modem direkt på kortet hade kunna spara plats för systemet, men då måste man eventuellt skärma modem mot störningar från kortet. Systemet skulle kunna utvecklas för att ge sjöräddningens personal möjlighet att kommunicera med de nödställda personerna vid olycksplatsen direkt via drönaren. Ekonomiska faktorer togs inte med i arbetet när kortet designades. De flesta komponenterna är billiga men kostnaderna kan bli onödigt höga om man ska producera många kort. Tillverkningskostnaden kan troligen reduceras genom ytterligare designarbete. För att förbättra testresultaten skulle man kunna bygga en dedikerad teststation för systemet för noggrannare mätningar. När konstruktionen för kortet gjordes så hade det hjälpt om vissa signaler var kopplade till stift för att underlätta direkt mätning av olika signaler.

## Referenser

- [1] Sjöfartsverket, "Statistik för sjö- och flygräddning 2019."  
URL: <https://www.sjofartsverket.se/sv/Sjofart/Sjo--och-flygraddning/Administration/Statistik/>  
Last accessed: 2020-11-13.
- [2] Sjöräddningssällskapet, "Frivilliga sjöräddare sedan 1907."  
URL: <https://www.sjoraddning.se/om-oss/>  
Last accessed: 2020-06-20.
- [3] RaspberryPi, "Compute module 3+."  
URL: <https://www.raspberrypi.org/products/compute-module-3-plus/>  
Last accessed: 2020-05-12.
- [4] M. T. In, "Lan9512."  
URL: <https://www.microchip.com/wwwproducts/en/LAN9512datasheet-toggle/>  
Last accessed: 2020-05-12.
- [5] A. In, "What is a voltage regulator."  
URL: <https://www.autodesk.com/products/eagle/blog/what-is-a-voltage-regulator/>  
Last accessed: 2020-06-20.
- [6] A. D. Team, "Using an airspeed sens."  
URL: <https://ardupilot.org/plane/docs/airspeed.html>  
Last accessed: 2020-06-20.
- [7] T. E. of Encyclopaedia Britannic, "Altimeter."  
URL: <https://www.britannica.com/technology/altimeter>  
Last accessed: 2020-06-20.
- [8] S. Electronic, "I2c."  
URL: <https://learn.sparkfun.com/tutorials/i2c/all>  
Last accessed: 2020-06-17.
- [9] C. Basics, "How to design a pcb layout,."  
URL: <https://www.circuitbasics.com/make-custom-pcb/>  
Last accessed: 2020-06-17.

- [10] A. Circuits, ‘‘Printed circuit board width tool,.’’  
URL: <https://www.4pcb.com/trace-width-calculator.html>  
Last accessed: 2020-05-23.
- [11] RaspberryPi, ‘‘Attaching a raspberry pi camera module to  
the compute moduleio board.’’  
URL: [https://www.raspberrypi.org/documentation/hardware/computemodule/  
cmio-camera.md](https://www.raspberrypi.org/documentation/hardware/computemodule/cmio-camera.md)  
Last accessed: 2020-05-23.