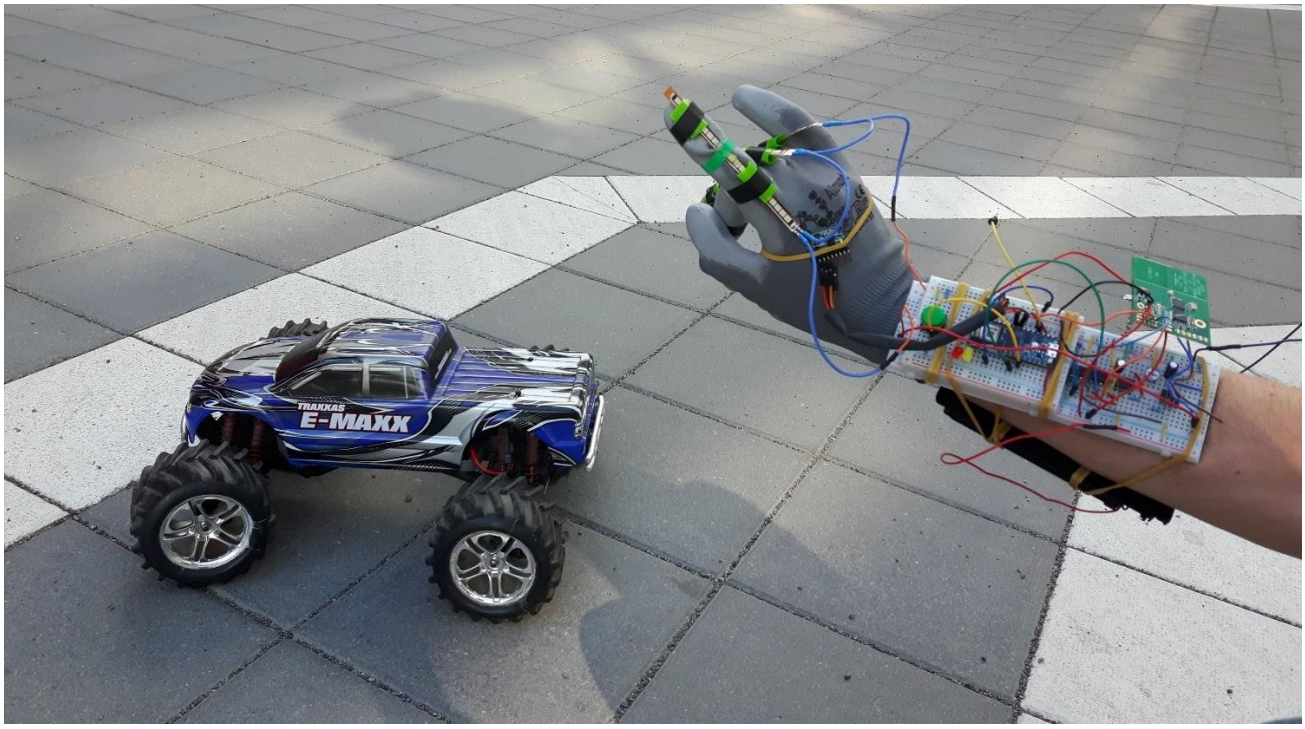
CHALMERS



Handgeststyrning av RC- fordon

Styrning av radiostyrd bil med hjälp av flexsensorer, IMU och mikrokontroller.

Examensarbete inom högskoleingenjörsprogrammet Elektroteknik (TIELL)

STEFAN ROS
ROBERT LUNDSTRÖM

INSTITUTIONEN FÖR ELEKTROTEKNIK

CHALMERS TEKNISKA HÖGSKOLA
Göteborg, Sverige 2021
www.chalmers.se

EXAMENSARBETE INOM ELEKTROTEKNIK 2021

Handgeststyrning av RC- fordon

Styrning av radiostyrd bil med hjälp av flexsensorer, IMU och mikrokontroller.

STEFAN ROS

ROBERT LUNDSTRÖM

CHALMERS

Institutionen för Elektroteknik
CHALMERS TEKNISKA HÖGSKOLA
Göteborg, Sverige 2021

Handgeststyrning av RC- fordon
Styrning av radiostyrd bil med hjälp av flexsensorer, IMU och mikrokontroller.

STEFAN ROS
ROBERT LUNDSTRÖM

© STEFAN ROS, ROBERT LUNDSTRÖM. 2021

Handledare: Jonas Williamsson, Consat Engineering AB

Examinator: Tomas McKelvey, Institutionen för elektroteknik

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

Omslag: Produktbild tagen under testkörning 2021-05-19

Sammanfattning

Teknik i dagens samhälle, såväl inom industrin som privat, tycks tillåta mer och mer omfamnande användarupplevelser genom användning av exempelvis virtuell verklighet och 3D- glasögon. I detta projekt konstrueras ett styrsystem som tillåter användaren att manövrera en radiostyrd bil med hjälp av handgester. Detta sker genom att ett antal sensorer monteras på en handske, vilka tillsammans med en framtagen elektronisk krets skickar signaler till ett styrkort från en befintlig handkontroll. Bilens hastighet regleras genom att böja på fingrarna och bilens riktning justeras genom att vinkla handen åt sidorna. Genom att luta handen bakåt kan användaren även bromsa eller backa fordonet.

Projektet pågick under vårterminen 2021 hos Consat Engineering AB och har genomförts delvis i Consats lokaler på Lindholmen och delvis hemifrån som följd av Covid-19 restriktioner.

Projektets syfte är att framställa en prototyp av denna styrenhet samt att undersöka vilka typer av sensorer som lämpligast kan användas för att registrera en hands rotationsvinkel i x- och y-led samt fingrarnas böjning. Prototypen utvecklas och testat på kopplingsplatta, men den kompletta kretsen designas även för mönsterkortstillverkning.

Arbetet är avgränsat till framtagning av just den styrenhet som ska ersätta originalfjärrkontrollens fysiska styrreglage. Närliggande teknik så som mottagande system, trådlös dataöverföring och antennteknik undersöks således inte. Vidare är framtagen prototyp anpassad för -och testad på just ett specifikt fordon (av modell Traxxas E-maxx), med tillhörande fjärrkontroll.

Styrenheten konstrueras som två delsystem: Ett system som känner av fingrarnas böjning med hjälp av flexsensorer och operationsförstärkare och ett system som registrerar handens rotationer med hjälp av en accelerometer ansluten till ett Arduino utvecklingskort. Båda delsystem knyts sedan samman genom anslutning till originalfjärrkontrollens tidigare styrreglages anslutningspunkter.

Arbetet resulterade i en styrenhet som funktionsmässigt motsvarar originalfjärrkontrollen, men samtidigt tillåter inlevelserik framförning av det radiostyrda fordonet. Dock behövs vidare arbete för att säkra enhetens pålitlighet, då mönsterkortskonstruktionen inte hann färdigställas i tid och styrenheten således endast testats monterad på ett kopplingsdäck.

Nyckelord: RC- fordon, radiostyrt fordon, Arduino, rotationssensor, flexsensor, operationsförstärkare, MEMS, IMU, accelerometer, gyroskop, drifting, mikrokontroller

Abstract

This report is written in Swedish.

Technology nowadays seems to offer more and more immersive experiences through for instance Virtual Reality and 3D technology, both in private- and industrial applications. In this project, a control system is constructed that allows the user to manoeuvre a radio-controlled car using hand gestures. This is done by mounting a number of sensors on a glove, which together with a developed electronic circuit sends signals to a control board, reused from an existing remote control. The car’s speed is regulated by bending the fingers, while the car's direction is adjusted by angling the hand to the sides. By tilting the hand backwards, the user can also brake or reverse the vehicle.

The project was conducted during the spring semester of 2021 at Consat Engineering AB and has been carried out partly in Consat's premises on Lindholmen and partly remotely from home as result of Covid-19 restrictions.

The purpose of this project is to produce a prototype of the above control unit and to investigate which types of sensors are most suitable to register the bending of a hand’s fingers and its rotational angle in the x- and y-plane. The prototype is developed and tested on a prototyping breadboard, but the circuit is also designed for printed circuit board manufacturing.

Note that this project is limited to the development of the control unit that will replace the original remote control's physical control elements. Related technologies such as the receiving system, wireless data transmission and antenna-technology are thus not investigated. Furthermore, the developed prototype has been adapted for -and tested on one specific vehicle only, namely one of model Traxxas E-maxx, along with its associated remote control.

The control unit is divided into two subsystems: One system that senses the flexion of the fingers using flex sensors and operational amplifiers and one system that registers the hand's rotations using an accelerometer, connected to an Arduino development board. These subsystems are then linked through the connection of the original remote control’s control board connection points for the previously used steering elements.

The work resulted in a controller unit that operatively corresponds to the original remote control, whilst allowing immersive controlling of the radio-controlled vehicle. However, further work is needed to secure the unit's reliability, as the printed circuit board design did not get finished on time and the system has therefore only been tested while mounted on a breadboard.

Keywords: RC vehicle, radio-controlled car, Arduino, rotational sensor, flex sensor, operational amplifier, MEMS, IMU, accelerometer, gyroscope, drifting, microcontroller

Förord

Denna rapport är en del av ett examensarbete vid institutionen för elektroteknik på Chalmers Tekniska Högskola. Examensarbetet som omfattade 15 högskolepoäng per student. Arbetet genomfördes under 20 veckor under vårterminen 2021 och var en avslutande del av det treåriga högskoleingenjörsprogrammet elektroteknik (TIELL) om 180 högskolepoäng.

Vi vill passa på att rikta ett stort tack till Consat Engineering AB som låtit oss genomföra detta arbete på avdelning Embedded Design på Lindholmen, trots begränsningar och risker gällande Covid-19 pandemin. Consat har förutom tillgång till lokaler även bistått med både radiostyrd bil samt material och utrustning för konstruktion, programmering och testning. Ett speciellt stort tack vill vi tilldela Jonas Williamsson, som i rollen som handledare för oss på Consat, stöttat och motiverat oss genom projektet och gett oss ett mycket gott omhändertagande under våren. Tack även till Ragnar Hallberg, Ole Andreas Utsomo och övriga Consat- ingenjörer som visat intresse för oss och vårt projekt och låtit oss störa er i ert arbete.

Vi vill även tacka Tomas McKelvey, som tagit sig an rollen att vara examinator för detta examensarbete.

Stefan vill också tacka styrelsen och medlemmarna i Lindholmens Makerspace för stöttning, inspiration och kamratskap och för alla utvecklingsmöjligheter de erbjudit honom, både tekniskt och socialt, under hans tid på Chalmers. Slutligen vill han även tacka Robert för ett fantastiskt samarbete, enormt engagemang, och stark uthållighet genom långa arbetsdagar, tekniska motgångar och logistiska problem orsakade av Covid-19.

Robert vill även tacka sin familj. Stort tack för allt stöd under dessa tre år, ni har peppat mig i motgångar och glatts åt mina framgångar. Till sist vill jag även tacka Stefan för ett gediget samarbete, en stor portion tålamod och ett härligt lugn. Hoppas vi kommer att jobba ihop i framtiden.

Stefan Ros & Robert Lundström, Göteborg 2021

INNEHÅLLSFÖRTECKNING

1 Inledning 2
1.1 Bakgrund 2
1.2 Syfte 2
1.3 Avgränsningar 2
1.4 Precisering av projektmål 3
2 Teknisk bakgrund 4
2.1 Sensorer 4
2.1.1 Accelerometer 4
2.1.2 Gyroskop 5
2.1.3 IMU 5
2.1.4 Tiltswitch 6
2.1.5 Flexsensor 7
2.2 Operationsförstärkare 7
2.3 Multiplexer 8
2.4 Arduino 8
2.4.1 Arduino Nano 33 IoT specifikationer 9
2.5 I^2C 10
2.6 Hysteres 11
2.7 Spänningsdelning 11
3 Metod 12
4 Genomförande 13
4.1 Förberedande arbete 13
4.1.1 Hårdvaruanalys 13
4.1.2 Val av sensorer 16
4.1.3 Val av mikrokontroller 17
4.1.4 Val av operationsförstärkare 18
4.1.5 Val av övriga komponenter 18
4.2 Mjukvara 18
4.2.1 I^2C 18
4.2.2 Kalibrering 19
4.2.3 Förskjutning 19
4.2.4 Filter 21
4.2.5 Hysteres 22
4.3 Hårdvarukonstruktion 23
4.3.1 Flexkretsen 23
4.3.2 Rotationskretsen 27
4.3.3 Sensorenheten 27
4.4 Montering 29
5 Slutsats 31
5.1 Användningsförfarande 33
6 Diskussion 34
6.1 Komponentbeställningar 34
6.2 Regleringskänslighet 34

6.3 Problemlösning 34
6.3.1 Operationsförstärkare 35
6.3.2 Mikrokontroller och programmeringsval 35
6.3.3 Driftning 36
6.4 Förbättringsförslag 36
6.4.1 Montering och PCB 36
6.4.2 Finjustering av hastighetsreglage 37
6.4.3 Överspänningar till styrkretsen 37
6.4.4 Systembugg vid handrotation 37
6.4.5 Tvärt riktningsomslag 37
6.5 Etiska och ekologiska aspekter 38
6.5.1 Hobbyanpassat projekt 38
Referenser 39
Figurförteckning 41
Tabellförteckning 42
Bilaga A. Materiallista ..... Bilagor, sid. 1
Bilaga B. Kretsschema ..... Bilagor, sid. 2
Bilaga C. PCB- ritning ..... Bilagor, sid. 3
Bilaga D. Programkod ..... Bilagor, sid. 5

TERMINOLOGI

- DAC – Digital-to-analog Converter. Omvandlar en serie av digitala signaler till en analog sådan.
- DMP - Digital Motion Processing. En matematisk funktion som kombinerar rådata från accelerometer och gyroskop. Funktionen är till för att avlasta mikrontrollen vid tunga och tidskrävande uträkningar.
- DoF – Degrees of Freedom. Anger kring hur många axlar en sensormodul mäter orientering.
- IDE – Integrated Development Environment. En utvecklingsmiljö som är framtagen för att underlätta kodning. Kan inkludera moment så som att kompilera och felsöka.
- IMU - Inertial Measurement Unit. En enhet som med hjälp av en accelerometer, ett gyroskop och ibland även en magnetometer, mäter rörelser.
- MEMS - Micro Electro-Mechanical System. En teknologi som blandar elektroniska och mekaniska komponenter, där de mekaniska komponenterna kan vara rörliga delar som beroende på rörelsen ger olika elektroniska utfall.
- MCU – MicroController Unit. Programmerbar mikrokontroller.
- PCB – Mönsterkort (från engelskans ”Printed Circuit Board”) som är ett maskinproducerat kretskort med tryckta ledningsbanor och kopplingspunkter på vilken elektriska komponenter monteras.
- RC – Radiostyrd (Från engelskans ”Radio Controlled”), exempelvis RC-bil innebär ”Radiostyrd bil”.

1 INLEDNING

I detta avsnitt ges information i form av bakgrund, syfte, mål, avgränsningar och frågeställningar.

1.1 Bakgrund

Många tekniska aktiviteter innefattar idag mer omslutande interaktion mellan användaren och tekniken, genom exempelvis förstapersonsvy (FPV) vid drönarflygning, 3D-glasögon och virtuell verklighet (VR) i TV- spel och industrier av olika slag. I detta projekt konstrueras en sensorkrets som kan användas för en mer inlevelserik styrning av fjärrstyrda enheter genom att tillåta användaren kontrollera ett fjärrstyrt fordon med hjälp av handgester. Kretsen konstrueras för användning med en radiostyrd bil (RC- bil), men tekniken kan även tillämpas inom många andra områden som fjärrstyrning av drönare, TV- spel och andra nöjesprodukter, styrning av robotar för hantering av farligt gods eller som ersättning eller komplettering av andra industriella styrreglage för ökad säkerhet och ergonomi. Tekniken skulle även i förlängningen kunna användas i forskningssyfte, tillsammans med andra sensorsystem för exempelvis undersökning av en persons rörelsemönster under specifika arbetsuppgifter.

1.2 Syfte

Projektet går ut på att framställa en produktprototyp av en sensorförsedd handske med tillhörande styrkrets, med vilken man ska kunna styra ett RC- fordon trådlöst. Styrkretsen ska konstrueras så att denne kan kopplas in i en befintlig fjärrkontroll och ersätta dess traditionella styrreglage. Under projektets gång ska även implementering av rotationssensorer undersökas och utvärderas för vilken eller vilka sensortyper som lämpar sig bäst för den färdiga produkten.

1.3 Avgränsningar

Vid genomförandet av detta projekt kommer en redan befintlig RC- bil av modell Traxxas E-Maxx (se Figur 1 nedan) och tillhörande fjärrkontroll Traxxas TQ-6516 (Figur 2) användas. Trots att prototypen teoretiskt kan användas till många andra fordon och ändamål, kommer den producerade enheten att anpassas för just denna bil och fjärrkontroll.

Figur 1. Använd RC- bil, Traxxas E-Maxx.

Figur 2. Traxxas TQ-6516 fjärrkontroll

Styrkretsen som utvecklas under arbetets gång ska endast ersätta den befintliga fjärrkontrollens styrreglage. Elektriska kretsar och komponenter för tolkning -och överföring av styrsignaler

mellan styrreglage och mottagare (bilen) likväl som systemets inbyggda antenn, återanvänds från redan befintlig fjärrkontroll. Dessa kommer inte undersökas i någon större utsträckning annat än för anslutning till den utvecklade prototypen. Undersökning av RF-teknik (RadioFrekvens) och optimering av trådlös kommunikation undersöks således inte i detta arbete.

Även de tester som utförs på det färdiga systemet kommer till viss del begränsas av använd mottagare, fordon och originalfjärrkontroll. Exempelvis kommer inte sensorupplösning kunna testas och optimeras fullt ut då känsligheten hos befintlig sändare, mottagare och/eller fordonets ställdon kan vara lägre än precisionen hos den konstruerade prototypen och dess komponenter.

Samtliga beräkningar och använd kretsteori i denna rapport utgår från ideala förhållanden och den färdiga prototypen testas endast under normala förhållanden med avseende på temperatur, väder och annan yttre påverkan. Små förändringar och avvikelser som följer av exempelvis temperaturpåverkan bortses därmed ifrån.

Om inget annat förtydligas, avses likspänning (DC) vid alla spänningsmätningar i denna rapport. Vid eventuell användning eller mätning av växelspänning eller pulsbreddsmodulering, kommer detta att tydliggöras i texten.

1.4 Precisering av projektmål

Projektet har delats in i följande delmål:

- Undersöka möjligheten att ersätta befintlig fjärrkontrolls styrreglage med sensorer.
- Utvärdera vilken typ av sensorer som lämpligen kan användas för detektion av handgester.
- Konstruera en elektrisk krets som med hjälp av valda sensorer och en mikrokontroller kan återskapa motsvarande signaler från de tidigare styrreglagen.
- Designa ett mönsterkort (PCB) för ovanstående krets.
- Implementera den skapade kretsen genom att fysiskt ersätta de befintliga styrreglagen i fjärrkontrollen.

Följande kravspecifikation ska eftersträvas under projektets gång:

- En radiostyrd bil ska kunna kontrolleras med hjälp av den framtagna prototypen, tillsammans med befintliga kretsar enligt projektets syfte och avgränsningar.
- Prototypen ska använda befintlig fjärrkontrolls spänningskälla och kunna kopplas samman direkt med befintlig krets.
- Bilen ska styras enligt följande handgester:
    - Böjning av fingrarna, med handen horisontellt mot marken – Bilen kör framåt.
    - Böjning av fingrarna, med handen lutandes bakåt – Bilen bromsar/kör bakåt.
    - Utsträckning av fingrarna – Bilen rullar fritt.
    - Lutning av handen åt höger – Bilen svänger åt höger.
    - Lutning av handen åt vänster – Bilen svänger åt vänster.
    - Mellan dessa extremlägen ska hastighet och riktning justeras i steg som upplevs som likvärdiga med motsvarande justering av originalreglagen.
- Externa sensorer ska användas för detektion av handrörelser.
- Manuell finjustering av styrsignal ska vara tillgänglig under användning.
- Kalibrering av enhetens utgångsorientering ska vara lättillgängligt vid drift.

- Uppstart av enheten ska ske på ett sådant sätt att bilen inte beter sig okontrollerat vid aktivering eller innan användaren är redo att ta kontroll över den.
- Vid framtagning av kretsdesign, ska möjligheter för framtida vidareutveckling, utbyggnad och anpassning av produkten beaktas.
- Användning av färdigproducerade moduler och utvecklingskort ska undvikas i möjlig mån och motiveras vid tillämpning.

2 TEKNISK BAKGRUND

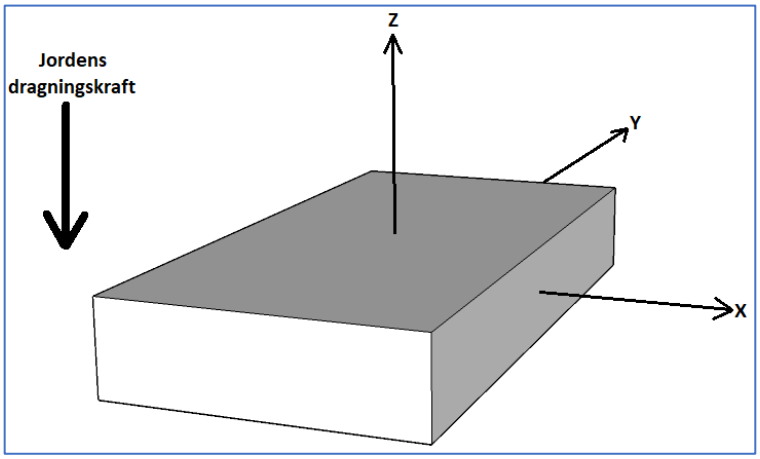
I detta kapitel redogörs för den hårdvara och tillämpningsteori som används i projektet.

2.1 Sensorer

I detta kapitel redogörs kortfattat principen bakom accelerometer, gyroskop samt tiltswitch. Detta kapitel tar sin utgångpunkt i att accelerometern och gyroskopet är utav typen micro-electro-mechanical systems (MEMS). MEMS-sensorer fungerar på ett likande sätt som mekaniska sensorer, men i ett mycket mindre format. Sensorerna mäter och producerar sedan rådata och det är denna rådata som sedan används för att beräkna vinklar och position. Sensorerna returnerar ett mätvärde (rådata) som sedan konverteras till önskad storhet så som vinklar och hastigheter.

2.1.1 Accelerometer

En accelerometer är en sensor som mäter accelerationen i förhållande till tyngdkraften för en axel. De finns i utföranden som mäter för 1, 2 eller 3 axlar. Tyngdaccelerationen (g) har enheten m/s^2 och 1 g är ungefär 9,80 m/s^2 beroende på plats på jorden. En sensor som ligger plant och stilla är endast påverkad av tyngdaccelerationen. Den tyngdkraft som en 3-axlad accelerometer då skulle mäta är således 1 g för z-axeln och 0 g för y- och x-axeln, se Figur 3.



Figur 3: Illustrationsbild över en MEMS accelerometer.

En viktig egenskap hos accelerometern är dess känslighet. MEMS accelerometrar går att ställa in på olika kännslighetsområden och vanliga intervall för området är ±2g, ±4g, ±8g och ±16g, där 0g motsvarar ±90°. Vilken rådata som sedan produceras och tillgängliggörs varierar mellan olika accelerometrar [1]. MPU6050 chipet har en 16-bitars upplösning och rådata anges i LSB/g, vilket är ett mått på accelerationen för en axel i förhållande till jordens dragningskraft.

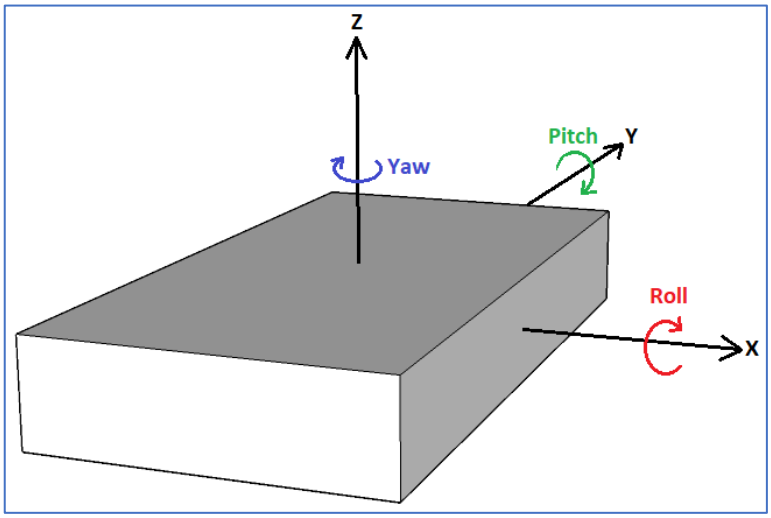
16-bitar ger 2^{16} = 65 536 och om tex känslighetområdet ±4g väljs, ger detta en kännslighetsfaktor på:

$$känslighetsfaktor = \frac{LSB}{g} = \frac{65536}{8} = 8192\ LSB/g \qquad (1)$$

Där g = 8 kommer ifrån intervallet -4 till + 4. Dessa värden kan sedan med hjälp av olika ekvationer användas för att beräkna vinklar.

2.1.2 Gyroskop

Ett gyroskop är en sensor mäter rotationen runt en axel. De finns i utföranden som mäter för 1, 2 eller 3 axlar, där vinklarna kallas Yaw, Pitch och Roll, se Figur 4. Dessa uttryck används bland annat inom flygbranchen och kallas där på svenska girvinkel, stigvinkel respektive rollvinkel.



Figur 4: Illustrationsbild över ett MEMS gyroskop.

Gyroskopet mäter vinkelhastighet för respektive axel genom att detektera och kombinera rörelsen i två riktningar och resultatet av detta presenteras i from av vinkelhastighet uttryckt i grader per sekund (°/s), där typiska värden kan vara ±250 °/s, ±500 °/s, ±1000 °/s, ±2000 °/s [2]. Kännslighetsfaktorn för gyroskopet i MPU6050-chipet baseras på 16-bitars upplösning. Om tex kännslighetsområdet ±250 °/s välj, ger detta en kännslighetsfaktor på:

$$känslighetsfaktor = \frac{LSB}{°/s} = \frac{65536}{500} = 131\ LSB/(°/s) \qquad (2)$$

Kännslighetsfaktorn och vinkelhastighet används sedan vid beräkningar för att bestämma respektive vinkel.

2.1.3 IMU

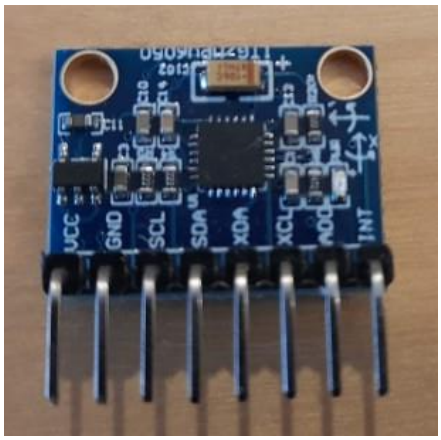
MPU6050 är en IMU (Inertial Measurement Unit) bestående av en 6 DoF (Degrees-of-Freedom) sensormodul som drivs av en egen processor. 6 DoF innebär att modulen inkluderar en 3-axlad accelerometer och ett 3-axlat gyroskop. Båda dessa är av typen MEMS. Modulen är även utrustad med en temperatursensor. Rådata från respektive axel går att nå genom olika specifika register i processorn. Processorn inkluderar en DMP och modulen kommunicerar via I^2C protokoll. Modulen använder sig av en 7 bitars adress, där de 6 mest signifikanta bitarna är statiska, vilket innebär att de inte går att ändra. Den minst signifikanta biten i adressen går att

påverka, vilket ger en möjlighet att välja mellan två olika adresser, nämligen 0x68 och 0x69 [3, sid 15]. Fullständig pinout för MPU6050 går att se i Tabell 1 och en bild av modulen går att se i Figur 5.

Tabell 1. Pinout och kortbeskrivning av respektive pinne för modulen MPU6050.

Stiftnamn	**Beskrivning**
VCC	VCC 3.0 – 5.0V
GND	GND
SCL	Serial Clock. Överför klocksignaler för att synkronisera sändningen mellan olika enheter. Pull-Up resistor är integrerad på kortet.
SDA	Serial Data. Överför data, antigen från slave till master eller från master till slave. Pull-Up resistor är integrerad på kortet.
XDA	Auxiliary Serial Data. Används för att koppla samman externa enheter med MPU6050.
XCL	Auxiliary Serial Clock. Används för att synkronisera överföringen mellan MPU6050 och externa enheter.
AD0	Address. AD0 = ’0’ ger modulen adressen 0x68. AD0 = ’1’ ger modulen adressen 0x69.
INT	Interrupt. Sätts hög för att indikera att MPU6050 har data att sända.

Då ingen extern kringutrustning är kopplad till modulen, DMP är ur bruk och adressen 0x68 används, är minsta koppling mellan sensormodulen och en mikrokontroller 2 trådar, en för SCL och en för SDA. Modulen måste även spänningssättas samt kopplas mot jord [3, sid 8].



Figur 5. En MPU6050.

2.1.4 Tiltswitch

Tiltswitch-sensorn är, som namnet antyder, en sensor som fungerar som en switch. Den ger en utsignal som är ’på’ eller ’av’, beroende på sensorns vinkel. Ett vanligt sätt att konstruera tiltswitchen på är att placera en eller flera kulor av metall i ett rör. När röret når en viss vinkel sluter brytaren kretsen genom att metallkulan kommer i kontakt med ledande material i röret och en ström kan således flyta genom kretsen. När kulan inte sluter kretsen, är denna bruten och stoppar strömmens genomflöde. Denna konstruktion gör att sensorn inte lämpar sig så bra för vibrationsrika miljöer. Dessa sensorer finns i en mängd olika utförande vad gäller vilken vinkel som ska fungera som brytvinkel [4].

2.1.5 Flexsensor

En flexsensor fungerar som en lång, platt potentiometer som ändrar resistansvärde linjärt i förhållande till dess böjning. Detta fungerar genom att ena sidan av sensorn har ett ledande material, i vilket avståndet mellan dess ledande partiklar ökar när sensorn böjs. Ju större detta avstånd blir, desto mindre ledande och därmed mer resistivt, blir materialet [5].

2.2 Operationsförstärkare

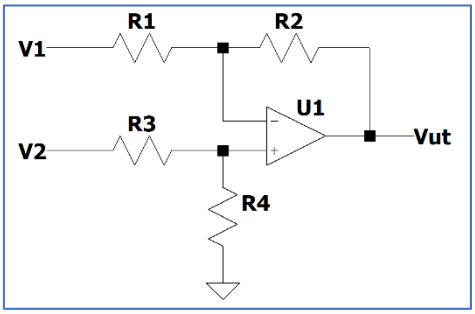
Den maximala utspänningen hos en operationsförstärkare (OP-förstärkare, OP) beror bland annat på dess matningsspänning. Historiskt sett är det vanligast med OP-förstärkare som kräver två matningsspänningar, V_+ och V_-, där V_+ är en positiv spänning, medan V_- är en lika stor spänning, men med omvänt tecken. Dessa kallas ibland Dual-Supply OP-förstärkare och får ett drivspänningsintervall vars storlek är lika stor som skillnaden mellan V_+ och V_-. Exempelvis kan en ideal OP med +/- 15 V matningsspänning teoretiskt ge en utspänning på 0 – 30 V. Men det finns även Single-Supply OP-förstärkare som kan drivas på en enda spänningskälla. En sådan ansluts mellan positiv spänning (vanligen VDD) och jord (GND) och får ett drivspänningsintervall som är lika stort som VDD. Som jämförelse med föregående exempel, kan en ideal Single-Supply OP där VDD är 15 V ge en teoretisk maximal utspänning på 0 – 15 V.

I verkligheten kan dock aldrig en operationsförstärkare ge ut en riktigt lika stor utspänning som dess drivspänningsintervall, på grund av interna spänningsfall. I datablad kan denna avvikelse mellan utspänning och matningsspänning avläsas som V_{OH} och V_{OL}, där V_{OH} (High level Output Voltage) visar den lägsta maximala utspänningen som kan fås medan V_{OL} (Low level Output Voltage) visar den högsta minimala. Alltså kan man i värsta fall kunna få en utspänning på mellan V_{OL} och V_{OH} från en operationsförstärkare.

Normalt sett är det relativt stor skillnad mellan maximal utspänning och drivspänningsintervall, men det finns OP-förstärkare som är anpassade för att minska ovanstående avvikelse [6].

En Rail-To-Rail OP är konstruerad så att den ska kunna avge en utspänning som är relativt nära matningsspänningens minimum- och maximumvärden, det vill säga V+ och V- för Dual-Supply respektive VDD och 0 för Single-Supply. En Single-Rail OP klarar bara att ge en utsignal som nära överensstämmer med en av matningsspänningarna, exempelvis V+ för Dual-Supply eller 0 för Single Supply, medan den övre avvikelsen är relativt stor.

En enkel differensförstärkare vars utsignal är en spänning som är proportionellt förstärkt mot differensen mellan dess insignaler enligt ekvation 3 nedan kan konstrueras med en OP-förstärkare enligt Figur 6, där resistanserna $R_1 = R_3$ och $R_2 = R_4$. [7, sid 171]



Figur 6. Kretsschema över differensförstärkare

$$V_{ut} = \frac{R_2}{R_1} * (V_2 - V_1) \quad (3)$$

Detta innebär att denna koppling kan användas som subtraktor, med vilken skillnaden mellan V_2 och V_1 kan fås som utsignal. Detta genom att sätta OP- kretsens förstärkning till 1, vilket sker då $R_1 = R_2 = R_3 = R_4$. Denna differens kan förstärkas på olika sätt genom att förändra relationen mellan dessa resistanser.

Genom små modifikationer av denna krets, kan man på liknande sätt använda OP-förstärkare för att även skapa bland annat adderare, multiplicerare, dividerare och deriverare.

2.3 Multiplexer

En multiplexer (MUX) är en integrerad krets bestående av flertalet logikgrindar. Dess funktion är att genom en eller flera selektorsignaler välja en signal från en av flera ingångar och skicka denna vidare till en utgång.

Vanligast är digitala multiplexrar, vilka endast släpper igenom signaler som är antingen logiskt höga eller logiskt låga. För att kunna välja -och vidarebefordra analoga signaler krävs multiplexrar som är anpassade för detta ändamål. Analoga multiplexrar använder dock fortfarande digitala signaler som selektorsignaler.

2.4 Arduino

Arduino är en plattform bestående av utvecklingskort, en utvecklingsmiljö samt ett programmeringsspråk. Det finns en rad olika utvecklingskort vilket gör det möjligt att välja kort utifrån behov. Några exempel på utvecklingskort är Arduino Uno, Arduino Leonardo och Arduino Nano. Dessa kort kan även fås i olika utföranden med olika egenskaper. Exempelvis är Arduino Nano 33 BLE Sense utrustad med en 6 DoF IMU, medan Arduino Nano 33 IoT har en IMU med 9 DoF. Den sistnämnda varianten har även en inbyggd DAC (Digital-till-analog konverterare), vilket den andre saknar.

Utvecklingskorten består av omprogrammeringsbara mikrokontroller. För att programmera korten används med fördel Arduino Software IDE. Genom IDE:n tillgängliggörs olika bibliotek och exempelkod vilka underlättar tillämpning, test och verifikation av relevanta moduler och funktioner.

Genom IDE:n möjliggörs användandet av Arduinos C-liknande programmeringsspråk, vilket innebär att användaren kan använda vissa förutbestämda funktioner och variabler. Exempel på förutbestämda funktioner är digitalRead och digitalWrite, vilka kan användas för att läsa av respektive skriva till en I/O -pinne [8].

2.4.1 Arduino Nano 33 IoT specifikationer

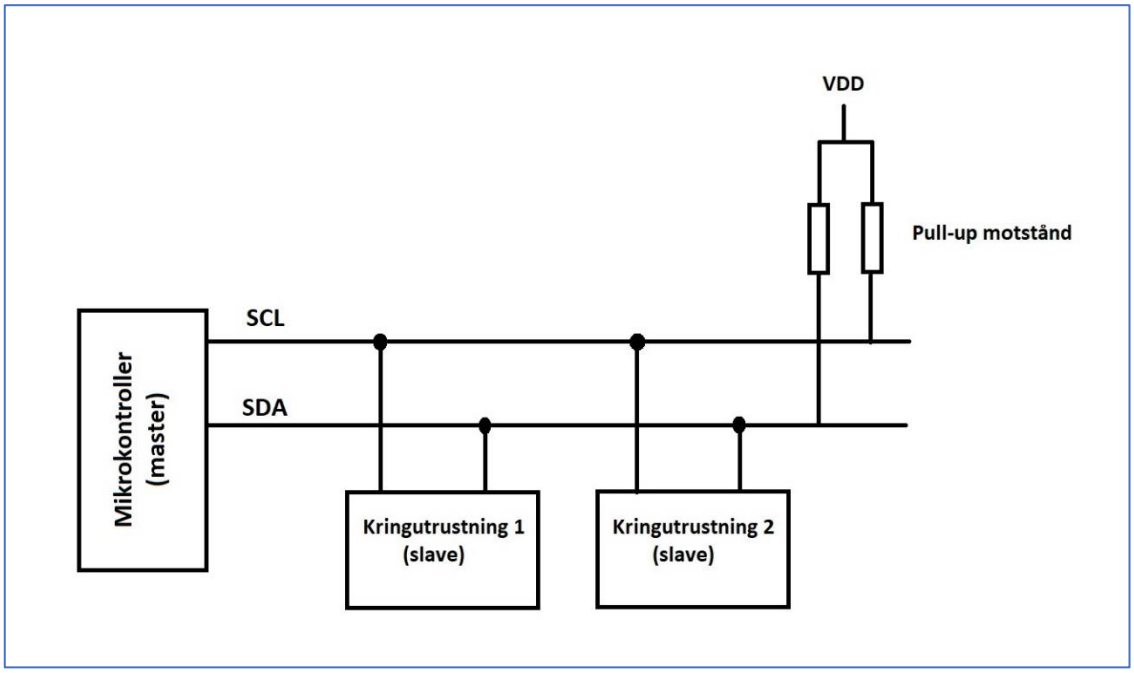
Arduino Nano 33 IoT baseras på en mikrokontroller (MCU) av modell SAMD21 Cortex®-M0+ och har en drivspänning på 3.3 V. Specifikationerna för Arduino Nano 33 IoT går att se i Tabell 2 [9].

Tabell 2. Specifikationerna för Arduino Nano 33 IoT.

Microcontroller	**SAMD21 Cortex®-M0+ 32bit low power ARM MCU**
Radio module	u-blox NINA-W102
Secure Element	ATECC608A
Operating Voltage	3.3V
Input Voltage (limit)	21V
DC Current per I/O Pin	7 mA
Clock Speed	48MHz
CPU Flash Memory	256KB
SRAM	32KB
EEPROM	none
Digital Input / Output Pins (nr)	14
PWM Pins	11 (2, 3, 5, 6, 9, 10, 11, 12, 16 / A2, 17 / A3, 19 / A5)
UART	1
SPI	1
I^2C	1
Analog Input Pins	8 (ADC 8/10/12 bit)
Analog Output Pins	1 (DAC 10 bit)
External Interrupts	All digital pins (all analog pins can also be used as interrupt pins but will have duplicated interrupt numbers).
LED_BUILTIN	Pin 13
USB	Native in the SAMD21 Processor.
IMU	LSM6DS3
Length	45 mm
Width	18 mm
Weight	5 gr (with headers)

2.5 I²C

Inter-Integrated Circuit (I²C) är ett protokoll för överföring av data via en buss. Protokollet använder sig 2 trådar för att sammanlänka olika enheter och flera enheter kan dela samma buss. De 2 trådar som används kallas serial data (SDA) och serial clock (SCL) och som synes i Figur 7 är alla enheter i en buss sammankopplade via dessa trådar. I en I²C buss finns det alltid en master-enhet och en eller flera slav-enheter och det är master-enheten som styr bussen och som genererar och skickar ut SCL-signalen.



Figur 7. Principen för ett system som använder I2C.

I²C använder open-drain vilket innebär att protokollet är aktivt lågt, dvs då någon enhet vill skicka bitar via någon av trådarna drar enheten ner spänningen till en logisk nolla via jord. För att sedan dra upp spänningen till en logisk etta igen används den externa spänningen (märkt VDD i Figur 7). Protokollet använder sig också av acknowledgment (ACK) och not-acknowledgment (NACK). Detta är en 1-bit data som måste skickas efter varje mottagen byte. I Tabell 3 visas hur det går till när en master-enhet läser data från en slav-enhet.

Tabell 3. Händelseförlopp för 1 bytes läsning.

Steg	1	2	**3**	4	**5**	6	7	**8**	**9**	10	11
Master-enhet	**S**	**AD+W**		**RA**		**S**	**AD+R**			**NACK**	**P**
Slav-enhet			ACK		ACK			ACK	DATA		

Förklaring till händelseförloppet i Tabell 3, i vilken master-enhet kallas ME och slav-enhet kallas SE:

1. ME skickar Start-bit (S)
2. ME skickar ut adressen (AD) till den SE som ska användas. Den skickar också ut att den ska skriva (W) till SE.
3. SE skickar en ACK-bit för att bekräfta att den mottagit en byte.

4. ME skickar adressen till det interna register (RA) hos SE som denna önskar få data ifrån.
5. SE skickar en ACK-bit.
6. ME skickar en andra start-bit (S).
7. ME skickar därefter SE:s adress (AD) igen, men denna gång med en read-bit (R).
8. SE skickar en ACK-bit.
9. SE skickar den data (DATA) som finns på det register som ME önskar läsa från.
10. ME skickar en NACK-bit när data är mottagen.
11. ME skickar en stop-bit (P) för att indikera att SE ska sluta sända data.

Om master-enheten önskar läsa från flera register, en s.k. burst read sequence, skickar den en ACK-bit i steg 10 istället för en NACK-bit. Slav-enheten skickar då data från nästkommande interna register i stigande ordning.

I Tabell 4 nedan visas hur det går till när en master-enhet skriver en byte data till en slav-enhet. Steg 1–5 i detta händelseförlopp är samma som vid läsning av data. Sedan följer stegen:

6. ME skickar önskade data (DATA).
7. SE skickar en ACK-bit.
8. ME skickar en stop-bit (P) för att indikera att den inte kommer skicka mer data.

Tabell 4. Händelseförlopp för 1 bytes skrivning.

Steg	1	2	**3**	4	**5**	6	**7**	8
Master-enhet	**S**	**AD+W**		**RA**		**DATA**		**P**
Slav-enhet			ACK		ACK		ACK	

Om master-enheten önskar skicka mer än 1 byte data, en s.k. burst write sequence, gör denna det genom att skicka nästa bit data istället för stop-biten i steg 8 [3].

2.6 Hysteres

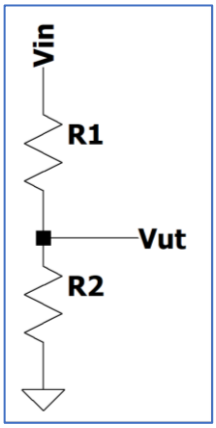
Fenomenet hysteres innebär att ett systems gränsvärden för interaktion är olika beroende på i vilken riktning dess värdeförändring sker. Ett exempel på detta kan vara vid temperaturreglering, där värmen ska slås på då temperaturen sjunker under en viss nivå, men slås av då temperaturen överskrider en annan nivå. Hysteres används för att undvika en okontrollerad systemförändring just kring ett specifikt gränsvärde.

Ett exempel på hur hysteres kan uppnås inom elektroniken är med schmittrigger- kretsar [10].

2.7 Spänningsdelning

Genom att koppla två resistorer i serie och sedan mäta spänningen över den ena av dem enligt

Figur 8, fås en utspänning som är proportionell mot insignalen och förhållandet mellan resistorerna.



Figur 8. Spänningsdelare

Med hjälp av ohms lag, kan spänningen V_{ut} beräknas enligt ekvation 4:

$$V_{ut} = \frac{R_2}{R_1+R_2} * V_{in} \quad (4)$$

3 METOD

Projektet delades in i följande huvudmoment:

- Planering av arbete.
- Val av sensorer och komponenter.
- Konstruktion av elektronisk styrkrets samt implementering av böjningssensorer.
- Programmering av mikrokontroller samt implementering av rotationssensor.
- Framtagning av PCB.
- Implementering av framtagen prototyp mot befintligt styrsystem samt testning.

Arbetet inleddes med att fastställa vilka handgester som skulle användas för att styra RC- bilen. Därefter undersöktes vilka sensorer som kunde tänkas vara lämpliga för att känna av dessa gester genom att besöka komponenthandlare på internet och leta igenom deras utbud samt genom informationssökning om IMU:s, gyroskop och accelerometrar (se avsnitt 2.1).

Signaler från de tre sistnämnda sensorerna behöver bearbetas, tolkas och omvandlas till lämpliga storheter. Detta görs programmatiskt, varför även en lämplig, programmerbar mikroprocessor valdes ut. Denna valdes genom att se till de kommunikations- upplösnings- och I/O-portkrav som ställdes av systemet och jämföra dessa med information ur datablad. Bilen provkördes även i originalutförande för att undersöka dess beteende och känslighet vid exempelvis förändring av styrutslag och hastighet.

Därefter undersöktes den befintliga fjärrkontrollen för att avgöra hur styrsignalerna behöver se ut. Detta gjordes genom att göra spänningsmätningar mellan styrreglagen och övriga kretsar inuti fjärrkontrollen medan reglagen sattes i viloläge samt dess extremlägen. Därmed kunde det fastställas hur prototypkretsen behövde designas för att avge en matchande signal.

Övriga komponenter valdes därefter för att kunna avge den eftersökta signalen och samtidigt kunna drivas av tillgängliga spänningskällor i fjärrkontrollen.

Beställning av ovanstående komponenter gjordes inledningsvis via Mouser, men för att minimera fraktkostnader och ledtider gjordes senare kompletteringsbeställningar via mindre återförsäljare inom Sverige och på eBay. Lista över använda komponenter kan ses i Bilaga A.

Därefter designades en krets i LTSpice, vilken även simulerades i samma program för att kontrollera att de förväntade signalerna kunde genereras. Kretsen ritades även upp i KiCad (se avsnitt 4.3.1 samt Bilaga B) för senare framställning av PCB (vilken kan ses i Bilaga C). Den uppritade kretsen monterades därefter på ett brödkort och dess utsignaler kontrollmättes med hjälp av ett oscilloskop.

Samtidigt skapades ett program för mikrokontrollern och dess kommunikation med IMU:n, vilka testades genom att skriva ut avlästa sensorvärden via Arduino IDE:s Serial Monitor. Störningar och signalbrus från IMU:n till Arduinon undersöktes genom verktyget Serial Plotter i Arduino IDE. Val av känslighetsfaktor och bandpassfilter för accelerometer valdes genom att undersöka utfallet för de respektive inställningarna, med hjälp av Arduino IDE:s Serial Monitor.

Slutligen monterades det framtagna systemet på en arbetshandske, där kretskort och lutningssensorer sattes fast på handens ovansida medan flexsensorerna monterades på pek-lång- och ringfinger.

Löpande under projektets gång, gjordes riskanalys av uppkomna och potentiella problem och risker för att avgöra om och hur dessa brister skulle åtgärdas eller motverkas.

4 GENOMFÖRANDE

Här följer en redovisning av hela projektets genomförande från förundersökning till framtagning av prototyp till implementering och testning. I detta kapitel refererar uttrycket ”rådata” till den definierad i avsnitt 2.2.1, det vill säga en axels acceleration i förhållande till jordens gravitation.

4.1 Förberedande arbete

Innan konstruktionsarbetet kunde påbörjas, behövde en konstruktionsplan göras. För detta behövde förberedande undersökning av konstruktionsalternativ samt befintlig hårdvara göras.

4.1.1 Hårdvaruanalys

För att kunna lägga upp en konstruktionsplan och veta vilka signaler som behövde genereras av prototypen undersöktes fjärrkontrollen och beteendet hos den radiostyrda bilen:

Bilens batterier laddades fullt och provkördes sedan med originalfjärrkontrollen för att undersöka dess köregenskaper, genom att använda bilen på sådant sätt som den är ämnad. Tester genomfördes i alla tänkbara riktningar och reglagenivåer och omfattade olika kombinationer av acceleration, inbromsning och neutralt fartläge i samband med olika styrutslag.

Dessa tester visade att styrningen generellt var relativt okänslig på just denna bil. Vid stillastående svängde bilen ofta inte alls förrän bilen började köra, oavsett läge på styrreglaget. Detta tros bero på bilens servomotorer, som inte ger samma utslag som fjärrkontrollens

känslighet, men då produkten skulle anpassas till denna bil, gavs lägre krav på känslighet av styrningens sensorer.

Vidare konstaterades att detta system har en inbyggd begränsning vid riktningsändring framåt/bakåt: Körning framåt eller bakåt, efterföljt av direkt omslag till körning åt motstående riktning gjorde att bilen bromsade kraftigt tills stillastående. Bilen fortsatte inte att köra åt sistnämnd riktning utan stod stilla tills fartreglaget ställts i neutralt läge och sedan åter till den önskade riktningen. Detta gällde dock inte om önskad riktning var samma som utgångsriktningen. För förtydligande exempel, se Tabell 5.

Tabell 5. Exempel på fartförändring vid växling mellan fram och back på styrreglage. Exemplen gäller då bilen redan har en ingångshastighet enligt fartreglagets första läge.

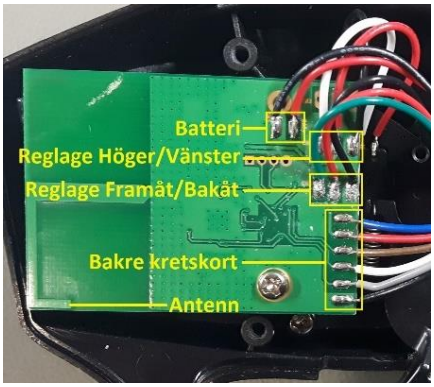
Lägeskombination, fartreglage	**Resulterande aktivitet, bil**
Framåt, sedan bakåt	Bromsar till stillastående
Bakåt, sedan framåt	Samma som ovanstående
Framåt sedan snabbt bakåt sedan snabbt framåt	Bromsar tills fartreglaget återgår till framåt, då bilen åter accelererar
Bakåt, sedan snabbt framåt sedan snabbt bakåt.	Samma som ovanstående, fast motstående riktning
Framåt, sedan bakåt, sedan neutralt, sedan bakåt (innan bilen hinner stanna)	Bilen bromsar till stillastående
Framåt, sedan bakåt, sedan neutralt när bilen har stannat, sedan bakåt	Bilen bromsar till stillastående och kör bakåt efter neutralläge.

Den befintliga fjärrkontrollen skruvades sedan isär för att undersöka hur befintliga kretsar och styrenheter såg ut. Det visade sig att den inbyggda elektroniken bestod av två mindre kretskort, hopkopplade med ett antal kablar (se Figur 9 – 11) samt ett batteripaket anslutet till det ena av korten.

Figur 9. Insida av fjärrkontroll. Styrkrets till vänster, inställningskrets till höger, batteripaket i botten.

Till det bakre kretskortet (höger i Figur 9), fanns en trimpotentiometer, en strömbrytare och en lysdiod anslutna. Potentiometerns funktion var att finjustera styrningen, strömbrytaren användes för att ändra inställningar och lysdioden visade fjärrkontrollens status.

Till det andra kortet (nedan kallat ”styrkretsen”) fanns förutom anslutningar till ovanstående krets och batteripaket, styrreglagen för fram/bak, höger/vänster inkopplade via 6 kablar enligt Figur 10.



Figur 10. Detaljbild styrkretsens baksida.

Figur 11. Styrkortets framsida, anslutningspunkter till vänster.

På detta kort fanns även ett antal extra anslutningspunkter och diverse andra komponenter, så som RF- sändare och riktad antenn för dataöverföring mellan fjärrkontroll och fordon. Styrsignalerna från reglagen undersöktes genom att mäta spänningen mellan ovanstående kablar och jord med hjälp av en voltmeter. Mätningen visade att anslutningspunkten ST gick till reglaget för bilens styrutslag höger/vänster, medan anslutningen TH gick till reglaget för bilens hastighet framåt/bakåt. Dessa anslutningspunkter och dess tillhörande anslutningar för jord och 3,3V spänningsmatning kan ses i Figur 11 ovan.

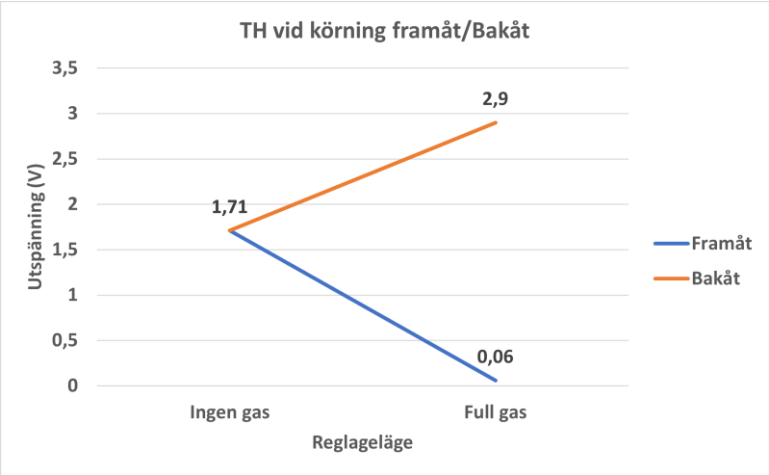
Efter detta uppmättes spänningen hos anslutna kablar vid TH och ST, vid olika lägen på tillhörande styrreglage för att fastställa önskade styrsignaler. Mätresultatet noterades i Tabell 6. Spänningen över batteriet mättes även vid anslutningspunkt BAT+, vilket visade en matningsspänning på 5,80V.

Tabell 6. Uppmätt spänning vid anslutningspunkterna TH och ST på fjärrkontrollens styrkrets.

Mätpunkt på styrkretsen	**Reglageläge**	**Mätvärde (V)**
TH (Grön kabel)	Neutral	1,71
TH (Grön kabel)	Fullt intryckt (full gas)	0,06
TH (Grön kabel)	Fullt framdragen (full back)	2,90
ST (Vit kabel)	Neutral	1,66
ST (Vit kabel)	Fullt utslag medsols (höger)	3,29
ST (Vit kabel)	Fullt utslag motsols (vänster)	0,09

Enligt dessa mätningar var maximal utspänning vid TH 2,90 V. Men då reglagets inspänning, utspänning ST samt driftspänning vid övriga delar av styrkretsen är 3,3 V, har verklig maximal utspänning vid både TH och ST antagits vara 3,3 V.

Genom att analysera uppmätta signaler i Tabell 6, noterades att utsignalen vid körning bakåt var omvänt proportionell mot utsignalen vid körning framåt (se Figur 12).



Figur 12. Diagram som visar spänningen över TH vid olika lägen på hastighetsreglaget.

Då maximal utspänning har antagits vara 3,3 V, enligt ovan, är dock detta diagram något missvisande: Om utspänning kan variera mellan 0 och 3,3 V, blir dess medelvärde 1,65 V. Uppmätt viloläge var 1,71 V, vilket är nära 1,65 V och således fortfarande rimligt.

För att undersöka spänningsgränserna för vilka RC- bilen stod stilla, startades RC- bilen varpå spänningen vid anslutningspunkt TH uppmättes medan gasreglaget justerades långsamt. Under testet hölls bilen i luften för att inte friktionen mot marken skulle bidra till hjulens inbromsning. Testet visade att bilen börjar driva framåt då spänningen över TH överskred 1,83 V. Bakåt stod däcken stilla till dess att spänningen över TH blev mindre än 1,58 V. Alltså befinner sig bilens drivmotorer i vilande tillstånd mellan 1,58 och 1,83 V, vilket också stämmer överens med ovanstående uppmätning samt beräkning av viloläge vid 1,71 respektive 1,65 V.

Då utsignalen från styrreglagen motsvarade en modifierad version av insignalen, där utsignalen var förändrad relativt till reglagens läge, konstaterades att motsvarande utsignal borde kunna återskapas med samma insignal genom användning av spänningsdelningar och signalförstärkare.

Den befintliga fjärrkontrollens styrkrets tillhandahöll två olika spänningskällor vid anslutning av ytterligare kretsar: Det bakre kretskortet samt båda styrreglagen var anslutna till varsin anslutningspunkt med 3,3 V reglerad utspänning. Fjärrkontrollens batteripaket var anslutna till kortet vid anslutningspunkt BAT+, vid vilken en oreglerad spänning på ca 4,0 – 6,0 V, beroende på batteriernas laddning, kunde levereras. Samtliga av dessa levererade endast likström.

4.1.2 Val av sensorer

Flexsensorns egenskaper och format gör denna ytterst lämplig för användning i detta projekt, varför det beslutades att spänningen och därmed bilens hastighet skulle justeras med hjälp av sådana, monterade på en handske. Kretsens utspänning skulle därmed förändras proportionellt mot flexsensorernas resistansvärden och därmed fingrarnas böjning.

Flexsensorn FS-L-0112-103-ST från Spectra Symbol har en längd på 112 mm vilket i normalfallet bör kunna täcka två knogar på ett finger utan att sticka ut för långt fram. Enligt databladet ska denna sensor ha en resistans mellan 10k Ω (utsträckt sensor) och 110k Ω (böjd sensor), men uppmätningar med multimeter visade att resistansen varierade mellan 10k Ω utsträckt och ca 30k Ω vid böjning som kunde motsvara fullt böjda fingrar [11].

För detektering av handens rotation krävdes en sensor som kunde känna av handens läge i relation till omgivningen och inte i relation till en annan kroppsdel (så som flexsensorn ovan). Därav fattades beslutet att ett 3 DoF gyroskop kunde monteras på handsken och således användas för detta ändamål. Då projektmålen innefattade framtagning av PCB, var det önskvärt att undvika användning av färdiga moduler, men då samtliga lämpliga gyroskop som kunde hittas hade en formfaktor som krävde professionella monteringsmetoder valdes modulen Adafruit 1032, vilket samtidigt erbjöd flexiblare monteringsmöjligheter för själva sensorn. Men strax efter att implementering av modulen påbörjats, upptäcktes ett problem med drivning av returnerat mätvärde. För att motverka detta problem, behövde kretsen kompletteras med en accelerometer för att användas som fast referenspunkt i förhållande till gyroskopets rotatiosvinkel. Därav byttes hela modulen ut mot en MPU-6050 som är en 6 DoF IMU vars specifikationer motsvarar den tidigare, med skillnaden att den har en adderad accelerometer.

Dock avhjälpte inte detta tillägg problemet med drivning av gyroskopvärde, men accelerometern visade sig vara tillräckligt funktionsduglig i sig själv, varför gyroskopet helt kunde slopas.

För val av riktning framåt eller bakåt övervägdes till en början att använda en tiltswitch för att känna av handens lutning framåt eller bakåt. Men då vald IMU även kan detektera rörelser i den riktningen valdes att implementera denna funktionalitet med hjälp av IMU:n istället. Detta gav dessutom bättre flexibilitet, då gränsläget för vid vilka vinklar på handen som ska aktivera riktningsändring hos fordonet kan fastställas i mjukvaran. IMU:n är dessutom bättre lämpad till ändamålet, dels för att programmatiskt bestämda gränsvärden tillåter hysteres, vilket här var önskvärt och dels då tiltswitchens uppbyggnad (se avsnitt 2.1.4) kan ge upphov till falska utslag vid exempelvis hastiga handrörelser.

4.1.3 Val av mikrokontroller

För att kunna välja korrekt mikrokontroller fastställdes först följande kravspecifikation:

- Inbyggd DAC med:
  - minst 1 analog utgång,
  - analog utgång på 0 – 3,3 V eller mer,
  - 8-bitars upplösning.
- Kommunikationsmöjligheter via I^2C.
- Drivspänning 3,3 – 5,0V.
- I/O-spänning 3,3 V.
- Minst 1 digital ingång (för kalibreringsknapp).
- Minst 4 digital utgång (för signalselektor och statusindikatorer).
- Minst 3 extra utgångar för utbyggnad.

MCU:n PIC18F16Q41 visade sig täcka dessa krav, varför denna primärt valdes ut. Dock visade det sig när programmering skulle påbörjas att denna inte var kompatibel med tillgänglig PIC-programmerare (PICKIT3). Därför beställdes en motsvarande, fast äldre modell av MCU, nämligen PIC16F1829.

På grund av tidsbrist togs dock snart beslutet att ersätta ovanstående MCU med ett utvecklingskort av typen Arduino Nano 33 IOT, som även den uppfyllde ovanstående krav. Detta för att den programmatiska tillämpningen av IMU:n var svår och drog ut på tiden och

Arduino kunde erbjuda både bättre felsökningsalternativ genom dess seriemonitor och ett stort utbud av kodexempel och förenklande bibliotek att titta på vid behov.

4.1.4 Val av operationsförstärkare

För enklare tillämpning och spänningssättning direkt via befintlig styrkrets, var operationsförstärkare som inte krävde negativ drivspänning önskvärt. Vidare krävdes att valda operationsförstärkare kunde leverera utspänningar nära 0 V till 3,3 V DC.

Dessa krav; att önskad utspänning skulle kunna vara nära tillgänglig drivspänning och den sistnämnda dessutom inte kunde vara negativ blev de mest avgörande faktorerna vid val av OP-modell. Då systemet inte ställde utmärkande höga krav på precision, blev kraven på operationsförstärkarnas noggrannhet och slew rate mindre avgörande vid komponentval.

Enligt databladet till OP-förstärkaren TLC271BCP kan denna drivas av 3V till 16V under optimala förhållanden (Under temperaturer mellan -40°C och 85ºC) [12].

Denna OP-förstärkare uppfattades dessutom vara av typ Single-Supply Rail-To-Rail, vilket tolkades som att den kan drivas av en enda spänningskälla ändå leverera utspänningar nära 0 V och 3,3 V även vid 3,3 V matningsspänning [13].

4.1.5 Val av övriga komponenter

Övriga komponenter så som resistorer, trimpotentiometrar och tryckknappar valdes ut löpande under projektets gång allteftersom behovet för dem tydliggjordes. I de flesta fall ställdes inga hårda krav på dessa komponenter, varför tillgänglighet eller leveranstid ofta blev avgörande vid val.

Resistorer valdes ut efter närmast möjliga värde utifrån gjord simulering i LTSpice, samtidigt som tillgänglig noggrannhet i närliggande utbud avvägdes. Resistorer valdes ur E96- serien, då dessa fanns lättillgängliga, erbjöd värden nära de önskade och hade en hög noggrannhet på 1%.

För att få kretsen att välja mellan att driva bilen framåt eller bakåt, behövde en 2-till-1 multiplexer implementeras. Då signalen som ska multiplexas var den som ska gå till styrkretsen, behövde denna vara analog, varför en analog multiplexer eftersöktes. De enda sådana som kunde hittas i närliggande utbud var 4-till-1 multiplexrar, vilka kunde användas genom att bara inte använda de kanaler som inte behövdes. Därav beställdes en 74HC4052 Analog multiplexer för detta ändamål.

4.2 Mjukvara

I detta avsnitt beskrivs den mjukvara som har tagits fram för reglering av bilens körriktning. Komplett programkod finns presenterad i Bilaga D.

4.2.1 I^2C

Arduinon kommunicerar med MPU6050-sensorn med hjälp av I^2C-protokoll. För att upprätthålla kommunikationen inkluderades det förinstallerade biblioteket wire.h. Med hjälp av detta bibliotek sattes Arduinon till att vara master. Inkluderandet av wire.h innebar också att vissa specifika funktioner kunde användas för att skicka data till- och läsa data från sensorn. Dessa funktioner kunde sedan i sin tur användas till att skapa funktionerna *Write_to_Slave* och *Read_from_Slave*. Efter detta sattes, med hjälp av databladet för MPU6050, de register som

krävs för att aktivera de funktioner som var önskvärda. Se Tabell 7 för vilka register som används. Detta görs i funktionen *setupMPU* och denna funktion kallas på efter att Arduinon har initierats till master.

Tabell 7. De register och tillhörande adresser som sattes för att kunna använda accelerometer.

Adress (Dec.)	Adress (Hex.)	Registernamn
107	0x6B	PWR_MGMT_1
26	0x1A	CONFIG
28	0x1C	ACCEL_CONFIG

Genom att skriva 0x09 till PWR_MGMT_1-registret väcks MPU6050, samt att temperatursensorn inaktiveras. Temperatursensorn är satt att inaktiveras då den inte används i detta projekt. Genom registret CONFIG sätts bandbredden för inom vilken sensormodulen ska jobba. Bandbredden sattes först till 260 Hz, men under störningstest (se avsnitt 4.2.3) kom detta att justeras till 21 Hz. Med hjälp av ACCEL_CONFIG-registret sattes accelerometerns känslighet till ±4096 LSB/g.

4.2.2 Kalibrering

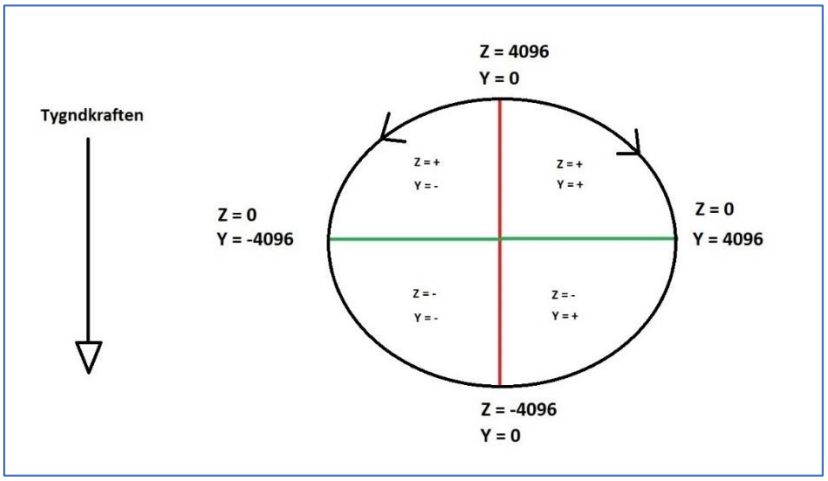
Efter att Arduinon har initierats, stannar programmet i en tom while-loop i väntan på att startknappen ska tryckas ner. Detta implementerades för att användaren ska ha kontroll över när denne vill att programmet ska starta. I samband med att användaren trycker på startknappen körs kalibreringsfunktionen *Calculate_Offset*. Funktionen gör 1000 läsningar av rådata från sensorn, adderar all rådata och delar sedan detta värde med 1000. Detta är ett krav för att starta programmet men användaren kan sedan kalla på funktionen *Calculate_Offset* om denna vill kalibrera sensorn igen vid ett senare tillfälle. Offset-resultatet subtraheras sedan från sensors rådata, genom följande kod:

```
acc_X = (acc_X_raw - acc_X_offset_ptr);
acc_Y = (acc_Y_raw - acc_Y_offset_ptr);
```

Denna funktion implementerades för att ge användaren möjlighet att hålla handen i en annan vinkel än plant vid start och därmed göra den nya vinkeln till utgångpunkt (0°) när bilen manövreras.

4.2.3 Förskjutning

För att mäta rotationer runt respektive axel användes accelerometerns rådata. Y-axeln är den axel som används för att svänga RC-bilen och X-axeln används för att bestämma om bilen ska köra framåt eller bakåt. Då variabeln *acc_X* och *acc_Y* undersöktes med hjälp av Seriell Monitor i Arduino IDE, upptäcktes det att då axlarna vinklas mer än ±90° i förhållande till tyngdkraften, börjar sensorn att räkna uppåt eller neråt (beroende på vilket håll sensorns vinklas i). I Figur 13 illustreras detta fenomen, medan Figur 14 visar resultatet av en sådan mätning. Det rödmarkerade värdet i bilden är vid -90° för Y-axeln och som synes så ökar sedan värdena igen och rör sig således mot noll igen.



Figur 13. Illustrationsbild över rotationen kring Y- och Z-axel.

```
acc_Y = -3635
acc_Y = -3655
acc_Y = -3867
acc_Y = -3764
acc_Y = -4053
acc_Y = -3978
acc_Y = -3894
acc_Y = -3859
```

Figur 14. Hämtade värden för Y-axeln.

Detta skulle i sin tur leda till att vid tex en vänstersväng med bilen, skulle denna sedan börja svänga tillbaka mot noll då sensorn vrids över ±90°. Detta var ingen önskad effekt. För att få bukt med problemet delades cirkeln i Figur 13 in i kvadranter med hjälp av den hämtade rådata från Y- och Z-axeln. Detta gav fyra unika kombinationer enligt Tabell 8, som sedan kunde användas för att räkna ut *acc_Y*

Tabell 8. Framtagna ekvationer för att kalibreringen ska fungera önskvärt.

Kvadrant	**Kombination**		**Ekvationer**
	Z	Y	
1	Positiv	Positiv	acc_Y = acc_Y_raw - acc_Y_offset_ptr;
2	Positiv	Negativ	acc_Y = acc_Y_raw - acc_Y_offset_ptr;
3	Negativ	Negativ	Acc_Y= 2*(-4096-(4096 - abs(acc_Y_raw)) - (acc_Y_offset_ptr))
4	Negativ	Positiv	acc_Y = 2*(4096+(4096 - abs(acc_Y_raw)) - (acc_Y_offset_ptr));

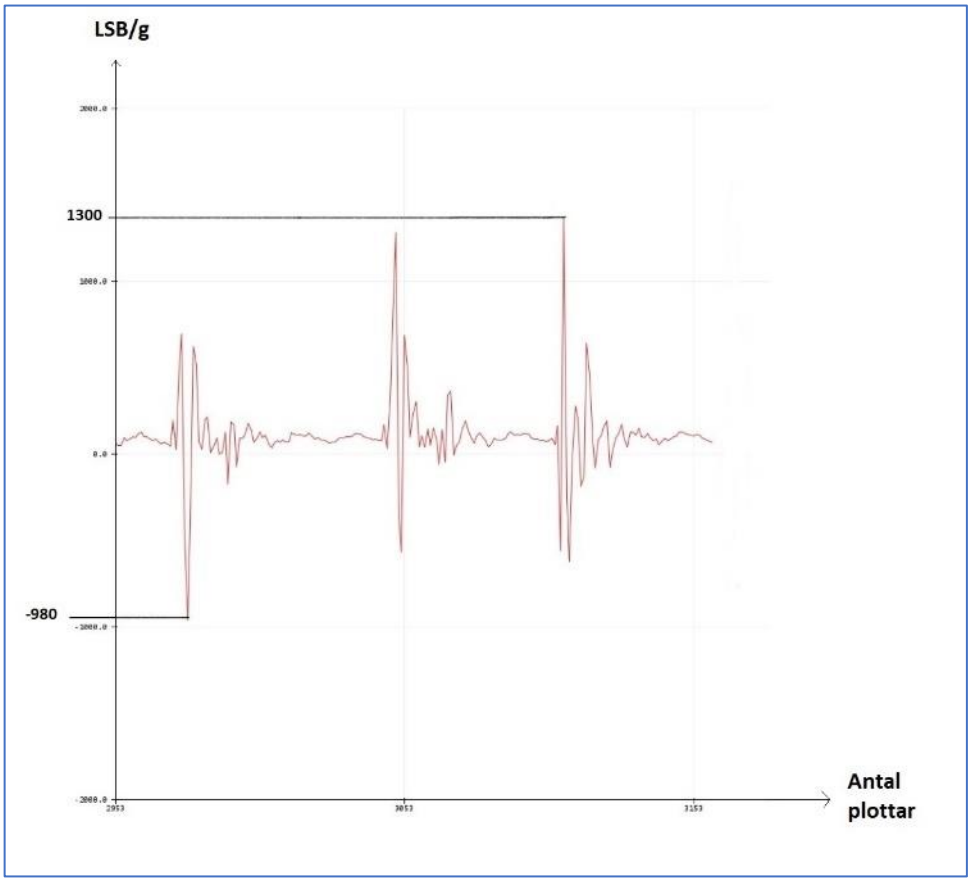
Detta resulterade i att det kalkylerade värdet från accelerometern fortsatte att minska eller öka (beroende på i vilken riktning sensorn vreds), även om värdet från den hämtade rådata minskade eller ökade. Resultatet av en sådan undersökning går att se i Figur 15. Det markerade värdet i bilden är vändpunkten för värdet av rådata, dvs. sensorn har vridits mer än 90° i förhållande till tyngdkraften, och värdet av denna rådata har således börjat öka igen. Detta samtidigt som det kalkylerade värdet, *acc_Y*, försätter att minska.

```
acc_Y = -1279
acc_Y_raw = -3955
acc_Y = -1314
acc_Y_raw = -3990
acc_Y = -1484
acc_Y_raw = -4032
acc_Y = -1424
acc_Y_raw = -4092
acc_Y = -1389
acc_Y_raw = -4127
acc_Y = -1490
acc_Y_raw = -4026
acc_Y = -1456
acc_Y_raw = -4060
acc_Y = -1516
acc_Y_raw = -4000
acc_Y = -1671
acc_Y_raw = -3845
acc_Y = -1718
acc_Y_raw = -3798
```

Figur 15. Kontroll av det kalkylerade Y-värdet.

4.2.4 Filter

Då MPU6050-chipet är känsligt för väldigt små rörelser och ger utslag på vibrationer, undersöktes det om detta eventuellt skulle kunna påverka projektet. Undersökningen skedde med hjälp av Arduino IDE:s Serial Plotter och accelerometerns känslighet var inställd på ±8g, vilket ger ett intervall på ±4096 LSB/g. Undersökningen gick till på ett sådant sätt att sensorn placerades plant på ett bord. För att vibrationer skulle uppstå knackades det på bordet. Resultatet går att se i Figur 16.

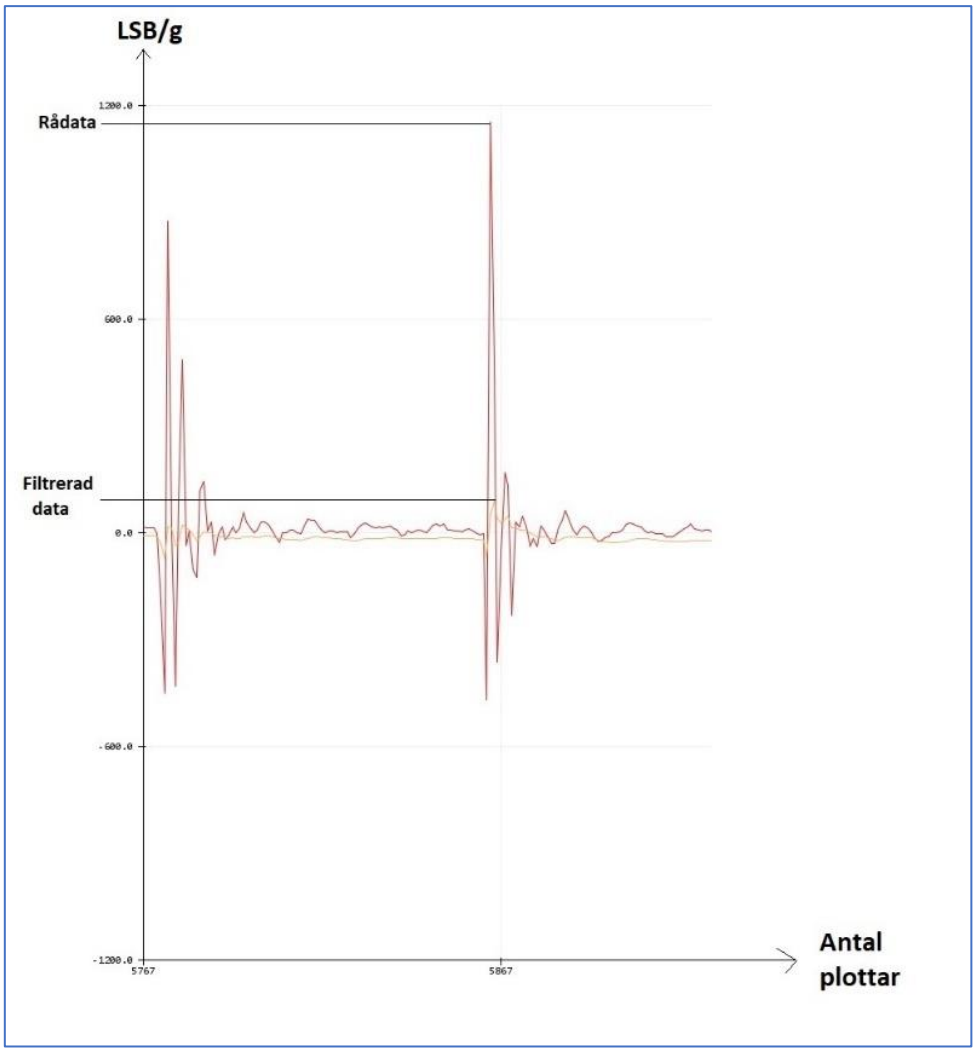


Figur 16. Vibrationstest av accelerometer (y-axeln)

Som synes i bilden så varierade värdet av accelerometerns rådata kraftigt vid vibrationer. Detta ansågs kunna störa systemet. För att komma till rätta med detta problem togs ett filter fram, se Bilaga D. Filtret är ett första ordningens Infinite Impulse Response-filter och baseras på följande ekvation:

$$s_t = \alpha x_t + (1 - \alpha) s_{t-1} \quad (5)$$

Filtret går ut på att vid en tidpunkt, *t*, spara undan värdet från accelerometern, vänta en tid, för att sedan få ett nytt värde och vikta dessa värden mot varandra med hjälp av $\boldsymbol{\alpha}$, där $0 < \boldsymbol{\alpha} < 1$ [14]. Genom att göra detta jämnas kurvan ut. Variabeln *w* fungerar som viktningen mellan värdena och tiden *t* justeras med hjälp av en loop. Vid framtagandet av filtret laborerades det med tiden och viktningen. Resultatet av appliceringen av filtret går att se i Figur 17, där samma test som tidigare utfördes.



Figur 17. Vibrationstest av accelerometer. Röd graf är rådata och gul graf är filtrerad data.

Som synes av resultatet i bilden var filtrerad data betydligt närmare noll LSB/g än rådata vid vibrationstestet. Den filtrerade avvikelsen ansågs vara inom rimliga felmarginaler för detta system.

4.2.5 Hysteres

X-axelns vinkelvärde sattes till att användas som en tiltswitch. Detta innebär att när sensorns läge är tiltat framåt drivs bilen framåt och när sensorn sedan vrids till ett visst läge bakåt, stoppas först bilen för att sedan kunna backa. Detta görs genom att Arduinon skickar en hög eller låg signal ut till MUXen. För att undvika att sensorn pendlar mellan gränsvärdet, infördes hysteres runt det satta gränsvärdet. Detta gjordes genom användning av två if-satser, se Figur 18.

```
296             //Backward
297             if ((forward == 0) && (filter_X > 2900)) {
298                 digitalWrite(MUX_out, 1);
299                 forward = 1;
300
301             }
302             //Forward
303             else if ((forward == 1) && (filter_X <= 2100)) {
304                 digitalWrite(MUX_out, 0);
305                 forward = 0;
306             }
307
```

Figur 18. De två if-satserna som används för att skapa hysteresen.

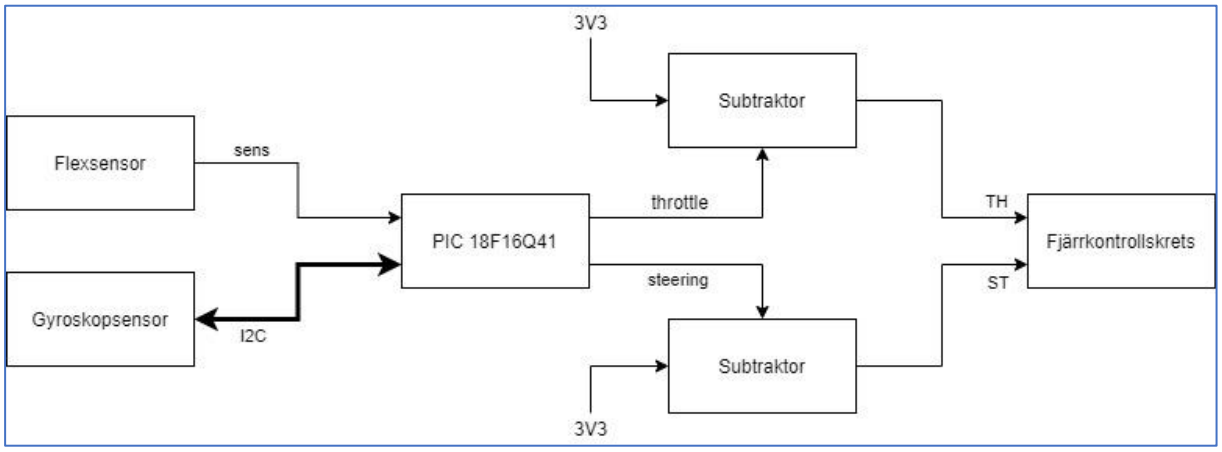
4.3 Hårdvarukonstruktion

Här beskrivs processen där enhetens elektriska krets designades, konstruerades och testades. För tydliggörande, har hela denna konstruktion (häri kallad ”sensorenheten”) delats upp i en ”flexkrets” och en ”rotationskrets”, där flexkretsen är den del av kretsen som inkluderar flexsensorn och hanteringen av dess signal fram till mikrokontrollern medan rotationskretsen är resterande del av sensorenheten, det vill säga mikrokontrollern, gyroskopet, accelerometern och dess tillhörande kringkomponenter.

4.3.1 Flexkretsen

Flexkretsen är den del av enheten som kontrollerar bilens hastighet. Detta sker genom att en utspänning till fjärrkontrollens styrkrets varieras. Storleken på denna spänning påverkar bilens hastighet och flexkretsen behövde därmed konstrueras för att ge en utspänning motsvarande uppmätta värden i Tabell 6 (avsnitt 4.1.1). Eftersom kretsens utspänning skulle vara olika beroende på om bilen skulle köras framåt eller bakåt, behövde kretsen kunna producera två olika, valbara utspänningar. Som exempel skulle kretsen vid fullt böjda sensorer ge 0 V vid körning framåt och 3,3 V vid körning bakåt.

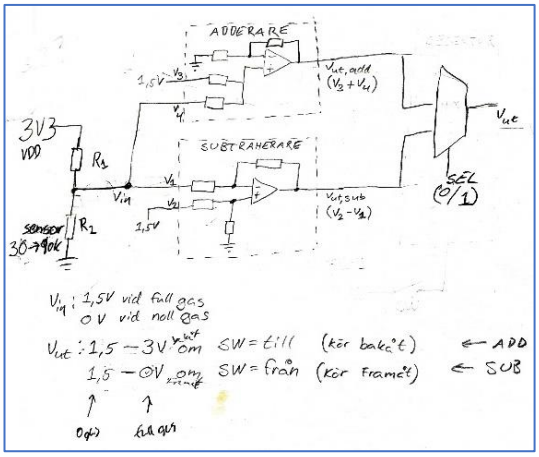
Kretsdesignprocessen påbörjades genom att först skriva ned en plan för hur kretsen kunde konstrueras och ett grovt blockdiagram ritades upp enligt Figur 19. Detta diagram blev utgångspunkten för projektet, men blev snart kraftigt förändrat, då systemet snabbt vidareutvecklades under arbetets gång. Dessutom var det önskvärt att kontrollera hastigheten till stor del via hårdvara, varför mikrokontrollern i möjlig mån skulle undvikas vid framtagning av flexkretsen.



Figur 19. Ett första utkast av ett blockdiagram över systemets funktionalitet.

Flera modeller togs fram på detta sätt och den då mest lovande idén var att koppla flexsensorn i serie med en spänningsdelare. Spänningen ut från denna spänningsdelare skulle därefter användas som referensspänning till två operationsförstärkare, vilka skulle addera respektive subtrahera ett fast värde till/från denna referensspänning. Därigenom skulle en utspänning fås som motsvarade den signal tidigare hastighetsreglage gav vid körning framåt respektive bakåt. Genom användning av transistorer skulle sedan en av dessa förstärkare aktiveras, beroende på om fordonet skulle röra sig framåt eller bakåt. Utspänningen från kretsen skulle sedan kopplas till den befintliga styrkretsens anslutningspunkt för hastighetsreglage (TH).

Efter att ha skissat upp denna design och undersökt brister och konstruktionssvårigheter med kretsen, justerades kretsen flera gånger och olika varianter av den togs fram. Den första förändringen som gjordes var att ersätta ovanstående transistorer med en analog multiplexer, vilket underlättade konstruktionen markant utan förlust av funktion eller byggyta. En skiss av denna versions kretsschema kan ses i Figur 20.

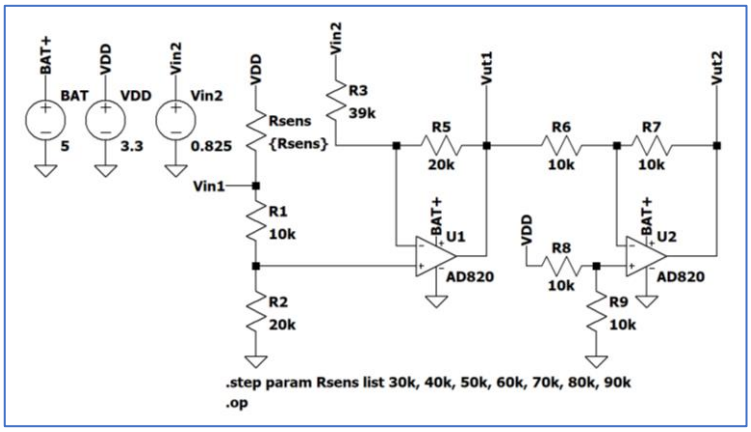


Figur 20. En grov skiss av en idéversion av flexkretsen.

Det under kretsanalysen (avsnitt 4.1.1) observerade förhållandet mellan THs utspänning vid körning framåt och densamma vid körning bakåt kunde användas i kretsdesignen, vilken reviderades igen så att förstärkarkretsarna istället seriekopplades och designades som differensförstärkare. Genom dessa kunde en utsignal för körning framåt beräknas av den första operationsförstärkaren medan en, mot denna omvänt proportionell, utsignal genererades av den andre operationsförstärkaren vilken användes vid körning bakåt. Signal för färd framåt respektive bakåt kunde sedan hämtas vid respektive operationsförstärkares utgång.

För enkelhets skull var det önskvärt att hela flexkretsen skulle drivas av en gemensam spänningskälla på 3,3 V. Men vid test av operationsförstärkaren visade det sig att denna då maximalt kunde leverera en utspänning på 2,8 V (se avsnitt 6.3.1 för mer information), vilket inte ansågs tillräckligt nära önskad maxspänning på 3,3 V. Därav valdes att använda batterikällan (BAT+) utan reglering som matningsspänning till OP-förstärkarna, vilket åtgärdade detta problem och OP-förstärkarna kunde leverera en utspänning på över 3,3 V.

Innan denna krets realiserades på kopplingsplatta, skapades en modell av den i LTSpice enligt Figur 21 för att simulera dess funktionalitet och underlätta val av komponentvärden. Samma figur används för vidare beskrivning av kretsen nedan.



Figur 21. Kretsschema för flexkretsen ur LTspice.

Till vänster i denna krets (Figur 21 ovan) syns R_{SENS}, vilken består av 3 seriekopplade flexsensorer med en gemensam total resistans om mellan 30k Ω och 90k Ω. De två resistorerna

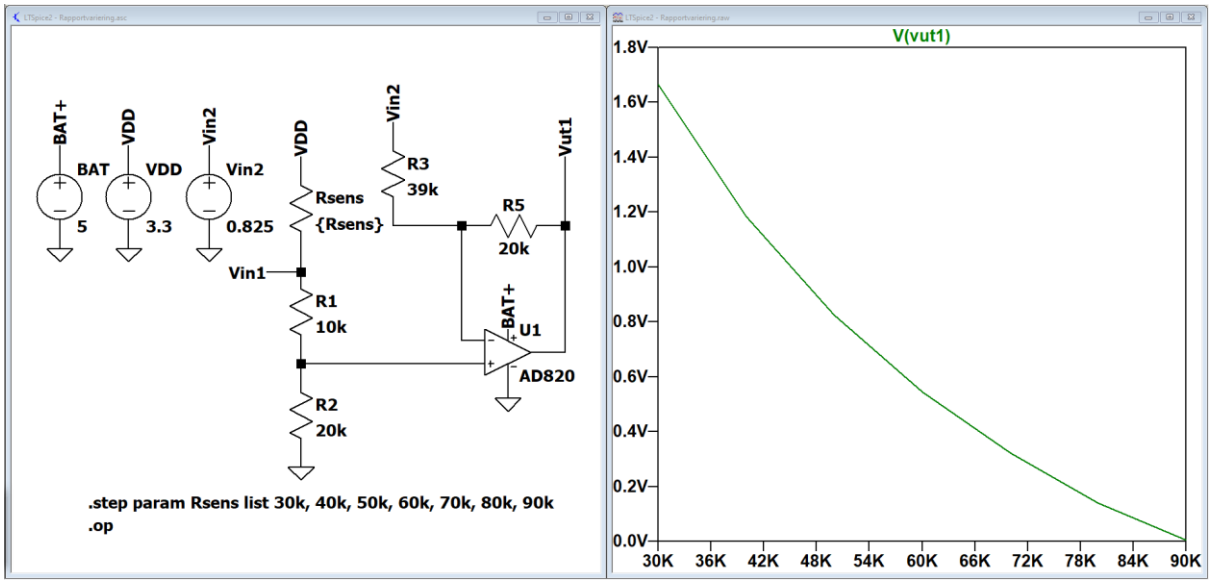
R_1 och R_2 agerar inresistans för U_1 och motsvarar resistorerna R_1 och R_3 hos en differensförstärkare enligt Figur 6 (avsnitt 2.2). Detta ger att V_{in} i Figur 21, kan motsvaras av V_2 i Figur 6, vilken kan betraktas som en spänningsdelning över R_1 och R_2 i förhållande till VDD och kan, enligt Ekvation 4 således variera mellan:

$$\frac{R_1+R_2}{R_1+R_2+R_{sens_max}} * VDD \;=\; \frac{30k\,\Omega}{120k\,\Omega} * 3{,}3\,V \;=\; 0{,}825\,V \text{ och}$$

$$\frac{R_1+R_2}{R_1+R_2+R_{sens_min}} * VDD \;=\; \frac{30k\,\Omega}{60k\,\Omega} * 3{,}3\,V \;=\; 1{,}65\,V.$$

Genom att använda U_1 som en subtraktor kan 0,825 V subtraheras från V_{in}, vilket ger ett lägsta värde V_{ut1} på $0{,}825 - 0{,}825 = 0$ V. Då V_{in} har sitt högsta möjliga värde, nämligen 1,65 V, subtraheras 0,825 även från denna spänning. Genom att multiplicera den resulterande differensen med 2, återfås maximala utsignalen V_{ut1} på 1,65 V. Tillsammans ges en utspänning V_{ut1} som går från 1,65 V mot 0 V då R_{sens} ökar från 30k Ω till 90k Ω. Allt detta uppnåddes genom att sätta R_2 = 2*R_1, R_3 = R_1, R_5 = R_2 och V_{in2} = 0,825.

Resulterande utspänning från U_1 som funktion av resistans hos flexsensorn simulerades genom funktionen ”Operating Point Analysis” (.op) i LTspice. Genom att göra R_{sens} till parameter och variera dess värde med direktivet .step kunde utspänningen analyseras vid olika resistansvärden på flexsensorn. Resultatet kan ses i Figur 22.



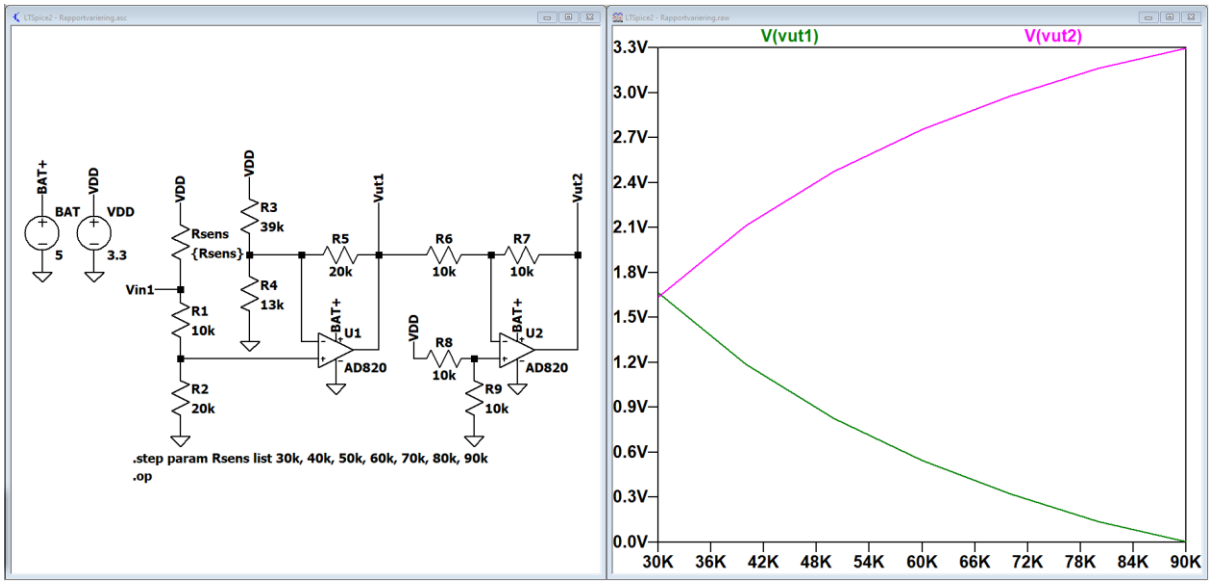
Figur 22. Simulering av flexkretsen i LTspice. Kurvan visar utspänningen Vut1 (V) i förhållande till resistans Rsens (Ω).

Efter detta lades en andra differensförstärkare, U_2, till i kretsen. Denna gång sattes samtliga kringliggande resistorer R_6, R_7, R_8 och R_9 till samma värde, 10k Ω för att få en subtraktor utan förstärkning. Som insignal på plussidan (motsvarande V_2 i Figur 6) sattes VDD = 3,3V och på minussidan (motsvarande V_1 i Figur 6) kopplades föregående förstärkares utgång. Därmed subtraheras V_{ut1} från VDD, vilket ger en utsignal V_{ut2} vars kurva är omvänt proportionell mot V_{ut1} och således går från 1,65 mot 3,3 V då resistansen R_{sens} ökar.

För att få en inspänning på 0,825 V vid V_{in2}, men fortfarande använda VDD som spänningskälla skapades en spänningsdelning vid V_{in2}. För att inte förändra förstärkningsfaktorn hos U_1, ersattes R_3 av en spänningsdelare vars totala resistans var nära R_3 = 10k Ω. Eftersom U_1 ser resistorerna hos

den nya spänningsdelaren som parallella enligt Thévenins Teorem fås en ekvivalent totalresistans på 9,75k Ω, sett från U_1, då R_3 = 39k Ω och R_4 = 13k Ω (se Figur 23 nedan), medan V_{in2} bibehålls oförändrad till 0,825 V [15].

Kretsen simulerades på nytt på samma sätt som tidigare och resultatet av detta kan ses tillsammans med det nya kopplingsschemat till vänster i Figur 23. Genom att antingen välja V_{ut1} eller V_{ut2}, kan nu flexsensorn användas för att ge en utspänning som varierar mellan antingen 1,65 och 0 V eller 1,65 och 3,3 V, vilket tydliggörs av grön respektive röd kurva till höger i Figur 23.



Figur 23. Kretsschema samt simuleringsresultat för Vut1 (V) (grön kurva) och Vut2 (V) (rosa kurva) som funktion av resistans Rsens (Ω).

För att möjliggöra finjustering av hastighetsregleringens max- och minimilägen ersattes R_4 i ovanstående krets med en trimpotentiometer i serie med en resistor. Resistorns motstånd valdes till 6,8k Ω och trimpotentiometerns maximala resistans sattes till 10k Ω. Detta gjorde att de två tillsammans har samma resistans som den tidigare R_4 då trimpotentiometern är satt nära sitt mittersta läge. Motsvarande resistansvärde R4 kan därmed justeras både uppåt och nedåt kring dess standardvärde. Effekten av denna justering är en förflyttning längs med x-axeln av graferna för V_{ut1} och V_{ut2} i Figur 23 ovan.

Om extremvärdena på flexsensorn avviker från de vid konstruktionen beräknade värdena kan styrningen påverkas på ett icke önskvärt sätt. Exempelvis kan en för hög minimal resistans göra att bilen inte stannar helt, medan en för låg maximal resistans skulle innebära att bilen inte når maximal acceleration. En sådan avvikelse kan nu kompenseras i viss mån (se diskussion i avsnitt 6.4.2 för avgränsningar), genom att vrida på trimpotentiometern RV1 (se kretsschema i Bilaga B) tills önskat beteende hos bilen är återställt.

Slutligen anslöts förstärkarnas utgångar V_{ut1} och V_{ut2} till varsin ingång på en multiplexer, medan dennes utgång senare anslöts till styrkretsens anslutningspunkt för hastighetsreglering. MUXens selektorsignal som avgör vilken av signalerna V_{ut1} och V_{ut2} som ska läggas mot utporten, kommer från mikrokontrollern, vars funktionalitet kan ses i nästa avsnitt (avsnitt 4.3.2).

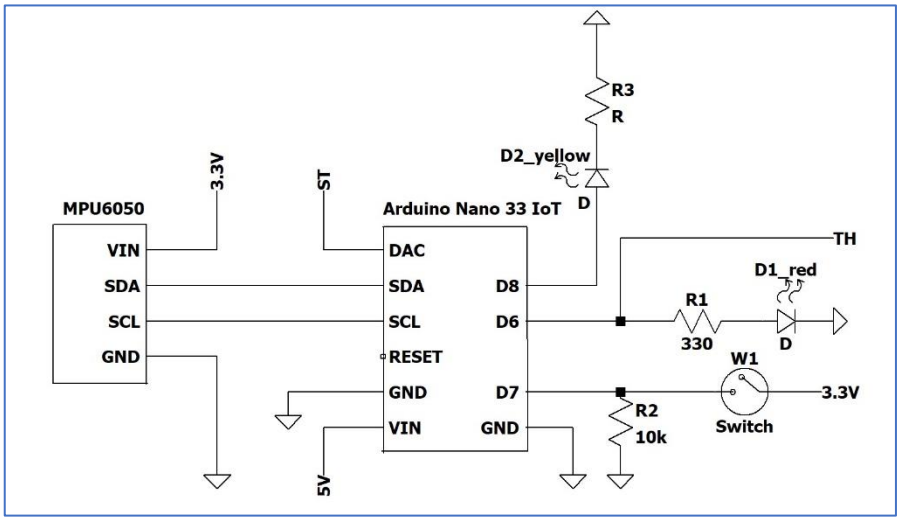
När ovanstående krets designats och simulerats, ritades ett kretsschema av densamma upp i KiCad för tydliggörande vid konstruktion samt som del av PCB- konstruktionen. Därefter monterades

kretsen enligt detta schema (vars slutliga version kan ses i Bilaga B) på ett kopplingsdäck för vidare tester och kontrollmätningar.

4.3.2 Rotationskretsen

Rotationskretsen, vars kretsschema kan ses i Figur 24 nedan, är den del av systemet som används för att svänga RC-bilen, samt bestämma huruvida bilen ska färdas framåt eller bakåt. Kretsen konstruerades med en Arduino Nano 33 IoT och en MPU6050 IMU. Arduino Nano 33 IoT har en inbyggd IMU som valdes bort då önskan fanns att kunna placera sensorn mer fritt från mikrokontrollen för att ge en mer flexibel prototyp. Rotationskretsen skickar styrsignaler till flexkretsens MUX för val av riktning framåt/bakåt beroende på handens vridning runt Y-axeln (pitch). En LED inkluderades för att ge indikation på om bilens riktning är fram- eller bakåt. Kretsen programmerades till att använda Arduinons inbyggda DAC. Detta gav den analoga signalen till styrkretsens anslutningspunkt, ST, för reglering av riktning åt höger/vänster. Signalen baseras på IMU:ns vridning runt X-axeln (roll). Rotationskretsen utrustades med en strömbrytare som fungerar som kalibreringsknapp.

MPU6050 har integrerade pull-up resistorer för I^2C kommunikationen.



Figur 24. Kretsschema över rotationskretsen.

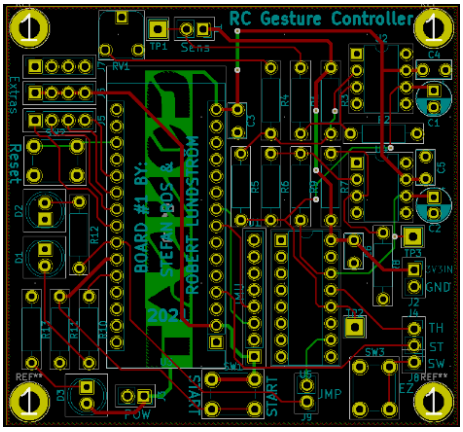
När rotationskretsen testades och undersöktes upptäcktes att Arduinons DAC varierade inom intervallet 0 – 3.1V och inte det önskade intervallet 0 – 3,3 V. Avvikelsen noterades vid mätningar men ansågs försumbar.

4.3.3 Sensorenheten

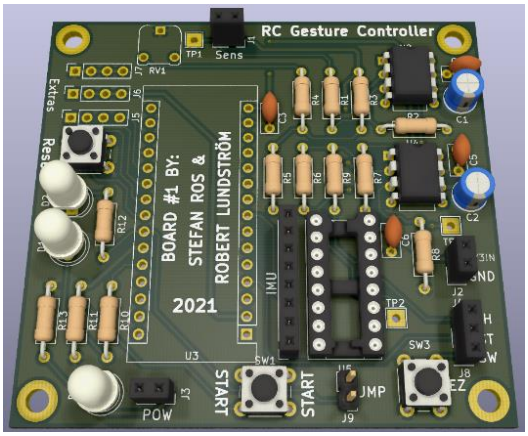
När rotationskretsen och flexkretsen var konstruerade och testade var för sig kopplades dessa ihop tillsammans med styrkretsen till ett gemensamt system. Komplett kretsschema för flex- och rotationskrets kan ses i Bilaga B. Styrkretsen anslöts sedan som extern modul via kontakterna J2, J4 och J8 i detta schema och flexsensorerna anslöts till J1. Anslutningspunkterna J5, J6 och J7 lades till för framtida vidareutveckling av kretsen och jumperpinnarna J9 används för att ansluta eller bryta kontakten mellan multiplexern och Arduinon i felsökningssyfte.

Utifrån ovanstående kretsschema skapades dessutom en komplett PCB- layout enligt Figur 25 nedan. Detta gjordes med hjälp av Pcbnew i KiCad. Komplett uppsättning av PCB:ns olika

lager visas i Bilaga C och en simulerad version av den färdiga kretsen, bortsett monterad Arduino, trimpotentiometer och multiplexer kan ses i Figur 26.



Figur 25. Framtagen PCB- design. (Consat- loggan använd med tillstånd)



Figur 26. Simulerad modell av färdigt kretskort.

När flexkretsens MUX tar emot en signal från rotationskretsen, sätts huruvida bilen ska färdas framåt eller bakåt. Flexkretsen skickar data om önskad acceleration (i framåt- eller bakåtriktning beroende på signal från rotationskretsen) till styrkretsens anslutningspunkt TH.

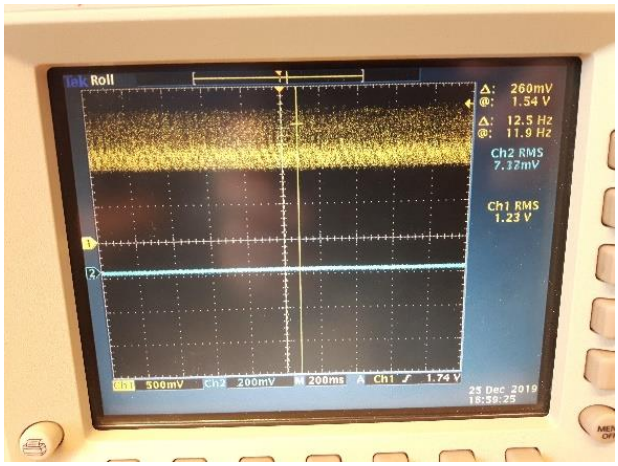
Styrkretsen och dess funktionalitet har inte förändrats från dess originalutförande och använder de mottagna signalerna från flex- och rotationskretsen för att skicka styrkommandon till mottagaren (RC- bilen).

Som spänningskälla för hela systemet används batteripaketet, som tagits från originalfjärrkontrollen. Denna driver direkt OP-förstärkarna i flexkretsen, Arduinon och IMU:n i rotationskretsen samt styrkretsen. Där matning om 3,3 V krävs i flexkretsen, hämtas detta från reglerad utspänning på styrkretsen. Likvärdig matning vid rotationskretsen hämtas från reglerad 3,3 V- utgång på Arduinon. På batteripaketet satt även en huvudströmbrytare, vars funktion behölls till det nya systemet.

Det bakre kretskortet (till höger i Figur 9, avsnitt 4.1.1) som var installerat i originalfjärrkontrollen togs bort då alla dess funktioner kunde integreras i den nya prototypkretsen: En tryckströmbrytare monterades på sensorenheten och anslöts direkt mellan jord och anslutningspunkten SW, vid vilken originalknappen tidigare var kopplad. Lysdioden från det borttagna kretskortet ersattes med en ny sådan, vilken nu styrs direkt av inspänningen från batteriet, så att denna alltid lyser när spänningen är på. Ytterligare två lysdioder adderades även till systemet, vilka indikerar status för kalibrering samt färdriktning framåt/bakåt. Det borttagna kortets trimpotentiometer blev redundant i det nya systemet, då dess funktionalitet ersattes av rotationskretsens kalibreringsfunktion (se avsnitt 4.2.2) och kunde således tas bort.

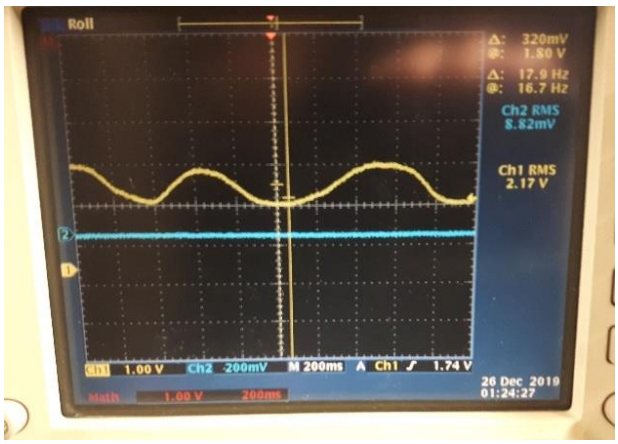
När samtliga delar av systemet hade anslutits, testades systemet mot den radiostyrda bilen. Omedelbart vid start började bilen långsamt driva okontrollerat framåt och bakåt, så att dess hjul upprepat och hastigt bytte rotationsriktning. Vid kontrollmätning av systemet med oscilloskop upptäcktes att detta orsakats av en kraftig störning i systemet då styrkretsen anslöts till sensorenheten. Störningen, som kan avläsas som det kraftiga bruset i Figur 27, varierade

cirka 0,7 V vid vilande läge, men dess storlek varierade kraftigt beroende på hastighetsreglering.



Figur 27. Uppmätt brus vid styrkretsens utgång

För att motverka detta brus kompletterades systemet med en 10 µF elektrolytkondensator (C1 samt C2 i Bilaga B) mellan vardera OP-förstärkares utgång och jord. Detta tog helt bort bruset (se resulterande mätning i Figur 28) utan att påverka systemets funktionalitet i övrigt.

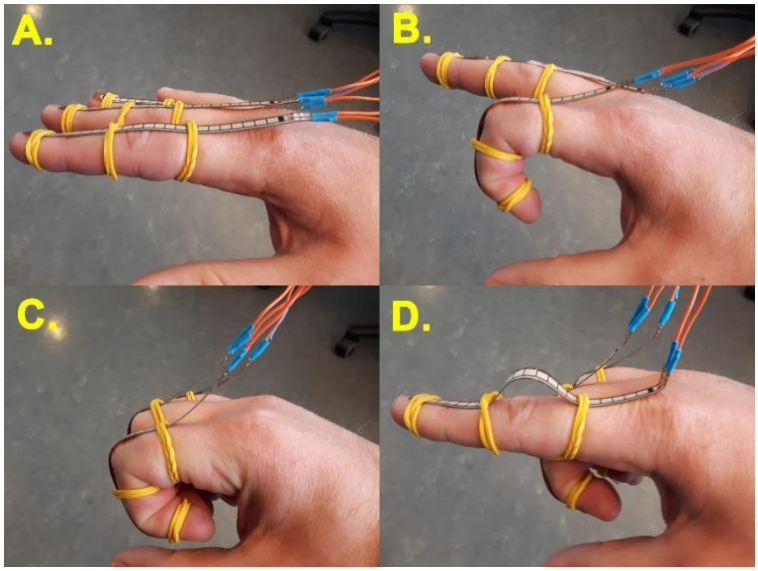


Figur 28. Uppmätt utsignal efter att störning filtrerats bort.

Utöver detta anslöts även keramiska avlastningskondensatorer (C3 – C6 i Bilaga B) på 100 nF vid matningsspänningens ingång på OP-förstärkarna, multiplexern och arduinon för ökat störningsskydd.

4.4 Montering

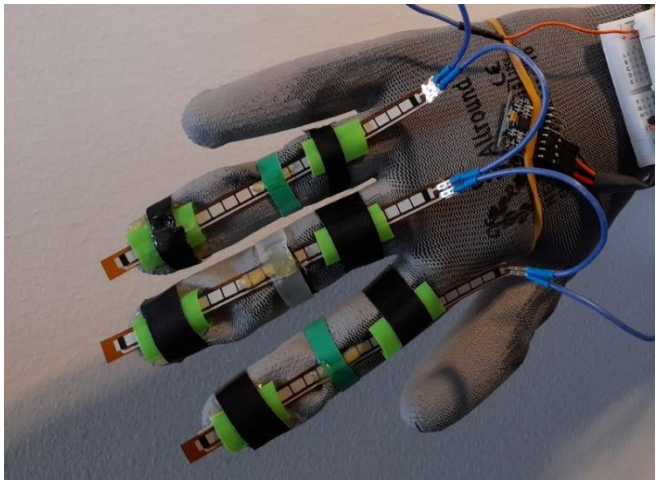
När hård- och mjukvaran konstruerats och testats enligt föregående avsnitt, monterades hela systemet på en handske. Flexsensorerna monterades på pek- lång- och ringfinger enligt Figur 30 nedan. Då flexsensorerna inte är töjbara på samma sätt som handsken eller skinnet på en hands knogar kunde de inte monteras fast längs med hela fingrarna då dess töjning behövde ersättas av rörelsefrihet längs med fingret. Problemet demonstreras i Figur 29, där sensorerna monterats på fingrarna med gummisnoddar, vilket begränsat sensorernas rörelse mot dess utgångsläge. Figuren visar även hur kabeln som använts vid sammankopplingen av sensorerna orsakar kraftpåverkan mellan dem vid fingerrörelser.



Figur 29. Flexsensorns beteende vid böjning och uträtning. Bild A: utsträckta sensorer. Bild B: Ett finger böjs, sensorn på detta finger reser sig i bakkant, Bild C: Alla fingrar böjda, sensorer resta i bakkant. Bild D: Finger åter uträtat, sensorn fortfarande böjd på mitten och vid anslutningspunkten.

För att undvika dessa problem, behövde sensorerna fästas på ett sätt så de kunde böjas och rätas ut tillsammans med fingrarna, utan att sitta fast längs med hela sensorn. Detta löstes genom att sätta fast sensorerna på fingrarnas mittpartier och sedan låta dem löpa fritt inuti ett plaströr på fingrarnas främre och bakre parti:

Först lades sensorn längs med fingrets ovansida och hölls fast med en bit krympslang runt sensorn och fingrets mittersta parti. Därefter limmades sensorn ihop med krympslangen. Sedan trädde sensorn in i en bit av ett sugrör, vilken limmades fast på fingrets främre och bakre parti. Slutligen säkrades plaströren fast mot handsken ytterligare med gaffatejp. Resultatet visas i Figur 30.

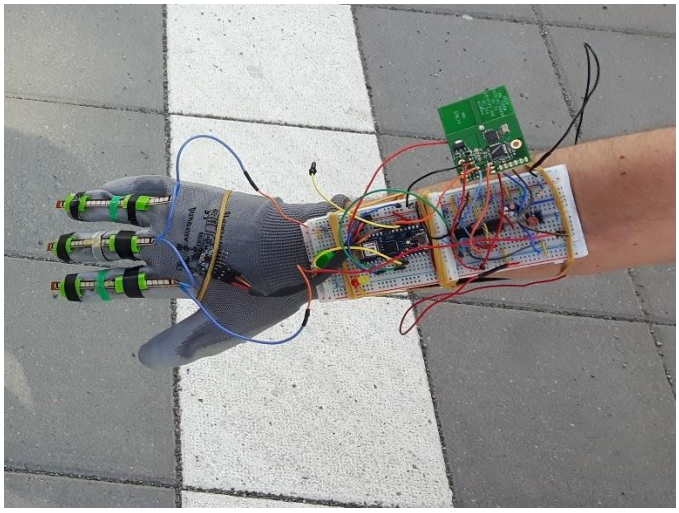


Figur 30. Flexsensorer monterade på en handske.

Denna konstruktion gjorde att sensorn kan sitta fast mot fingret och följa dess böjning, men fortfarande kan glida något över fingrets inre och yttre parti samt röra sig något från handens knogar vid behov.

För att undvika påfrestning på sensorernas lödpunkter och ytterligare möjliggöra rörelse av sensorerna individuellt från varandra, ersattes även den något styva orange kabeln i Figur 29 med en mjukare flertrådskabel av silikontyp. Detta medförde att effekten som kan ses i bild C och D i samma figur, minskade och sensorerna kunde röra sig mer fritt utan att böjas av varandras koppling via kabeln.

När flexsensorerna monterats på handsken, anslöts IMU:n och styrkretsen till det kopplingsdäck där övriga komponenter satt monterade och samtliga kretsar fästes med gummiband på ovansidan av användarens underarm enligt Figur 31. På undersidan av underarmen sattes batteripaketet fast med samma gummiband. IMU:n fästes med ytterligare ett gummiband på handens ovansida. Därmed var systemet klart för testkörning.



Figur 31. Hela systemet monterat på hand och arm.

5 SLUTSATS

När hela systemet kopplats ihop och monterats enligt föregående kapitel genomfördes ett funktionstest motsvarande det första testet som gjordes vid hårdvaruanalysen i avsnitt 4.1.1. Testet var lyckat och bilen navigerade så som önskat utefter handens rörelser och systemet fungerade således som förväntat, med undantaget att bilen ibland slutade svara helt på kommandon som följd av kontaktproblem vid det monterade kopplingsdäcket.

Resultatet av arbetet blev alltså en prototyp som uppfyller projektets syfte samt de flesta av de specificerade projektmålen och -kraven:

Den framtagna prototypkretsen, bestående av flexsensorer, accelerometer, mikrokontroller och kringkomponenter har ersatt styrreglagen i originalfjärrkontrollen samt ett av dess inbyggda kretskort utan märkbara funktions- eller precisionsförluster. Kretsen samverkar med originalfjärrkontrollens styrkort och spänningskälla enligt uppsatta projektkrav. Valda sensorer har uppfattats som korrekt valda och därmed ytterst lämpliga för att uppfylla projektets syfte.

Kalibrering av styrning höger/vänster sker genom en lättillgänglig strömbrytare (SW1 i Bilaga B, märkt ”Start” på PCB:n i Bilaga C) medan finjustering av hastighet kan ske genom trimpotentiometern RV1 (Bilaga B). Handrotationer registreras inte förrän kalibrerings-/startknappen tryckts ned och vid eventuella styrfel finns även en reset- knapp monterad (SW2

i Bilaga B, märkt ”Reset” på PCB:n i Bilaga C), vilken nollställer mikrokontrollern. Samtliga av dessa funktioner samt huvudströmbrytare finns lättillgängliga både innan och under drift, vilket således uppfyller ställda krav på finjustering, kalibrering och säker uppstart.

EZ- knappens funktion på originalfjärrkontrollen har behållits genom att denna flyttats till prototypkretsen (SW3 i kretsschemat, Bilaga B, märkt ”EZ” på PCB:n i Bilaga C)

En PCB- design har färdigställts enligt projektets mål, men har ej hunnit produceras och levereras i tid, varför prototypen vid projektets slut endast är konstruerat på ett kopplingsdäck. Där det platsmässigt har tillåtits, har även extra anslutningspunkter till mikrokontrollern implementerats i PCB- designen för möjliggörande av framtida vidareutveckling. De testpunkter (märkta TPx i kretsschemat och PCB- layouten) har dessutom designats med monteringshål, för att även dessa ska kunna användas som fasta anslutningspunkter vid behov. Samtliga monteringspunkter har designats för att vid montering tillåta val av direkt pålödning av kablar alternativt användning av kontaktskenor med ett hålavstånd om 2,54 mm.

För test och felsökning av kretsen har, förutom ovanstående testpunkter även en bygling implementerats mellan flex- och rotationskretsen. Denna bygling (kontakt J9, märkt JMP i kretsschemat) bryter selektorsignalen mellan MUX och Arduino och tillåter användaren således att sätta denna signal själv genom extern stimulering. Vid normal användning av systemet, kortsluts kontakterna på JMP, vilket sluter kontakten mellan delsystemen.

Enligt projektkraven skulle användningen av färdiga moduler och utvecklingskort undvikas i möjlig mån, varför användning av en PIC mikrokontroller samt enskilda sensorkomponenter undersöktes vid projektets början. Mikrokontrollern ersattes dock av en Arduino Nano på grund av tidsbrist (Se avsnitt 4.1.3). Hade inte detta val gjorts hade troligen inte projektet hunnit färdigställas i tid. Även vid val av rotationssensorer, beslutades att använda en modul så som beskrivet i avsnitt 4.1.2. Detta dels för att underlätta implementering, men framför allt då dessa sensorer inte fanns som hanterbara komponenter hos leverantör. Funktionsmässigt har detta val inte påverkat slutprodukten på något sätt, då modulen fungerar på samma sätt som motsvarande egen konstruktion skulle gjort.

Sammanfattningsvis fungerar systemet bra och uppfyller de förväntade funktionerna. Dessutom ges användaren en speciell kontrollkänsla med mycket inlevelse, då man måste röra sig så som man vill att bilen ska manövreras. Dock finns rum för förbättring och vidareutveckling inom vissa områden (se diskussion nedan). Den viktigaste av åtgärderna är att ordna en mer pålitlig koppling, då den nuvarande konstruktionen på kopplingsdäck är mycket bristfällig. Montering av designad PCB skulle troligtvis lösa de fel som uppstod under test, men har inte hunnit göras under projektets gång.

5.1 Användningsförfarande

Det slutliga systemet används så som kravspecifikationen föreskrev:

1. Användaren sätter på sig sensorhandsken. Batteripaketet fästs på lämpligt sätt på användarens underarm.
2. Flexsensorerna kontrolleras så att de böjs som väntat vid fingerrörelser.
3. Fingrarna sträcks ut och hålls raka.
4. Huvudströmbrytaren slås på och den gröna lysdioden tänds.
5. Handen hålls vågrät med marken (eller av användaren valt utgångsläge), samtidigt som Start-/Kalibreringsknappen trycks ner. Den gula lysdioden tänds.
6. Handen hålls stilla till dess att den gula indikatorn slocknat.
7. Mottagaren på RC- bilen startas.
8. Nu styrs bilen med följande handrörelser:
    - Fingrarna böjs, med handen i utgångsläge – Bilen accelererar framåt.
    - Handen lutas bakåt – Röd lysdiod tänds för att indikera bromsning/backning.
    - Fingrarna böjs, med handen lutandes bakåt – Bilen bromsar/accelererar bakåt.
    - Utsträckning av fingrarna – Bilen rullar fritt.
    - Lutning av handen åt höger – Bilen svänger åt höger.
    - Lutning av handen åt vänster – Bilen svänger åt vänster.

För att växla mellan drift framåt och drift bakåt måste bilen sakta ner till helt stillastående innan den kan börja driva åt motstående riktning som innan. Därav måste fingrarna helt sträckas ut medan handen lutar i önskad riktning. När bilen stannat, kan fingrarna åter böjas för acceleration i motstående riktning.

Om bilen inte stannar helt vid utsträckta fingrar, eller det inte går att växla mellan framåt- och bakåtdrift, justeras trimpotentiometern RV1 till dess korrekt beteende är uppnått.

Om bilen inte svänger enligt ovanstående handrörelser, trycks kalibreringsknappen ned på nytt, medan handen är i horisontellt läge. Detta återställer rotationssensorns referenspunkter.

Om enheten inte svarar alls eller påvisar ett oväntat beteende, kan Reset- knappen tryckas ned, vilket startar om mikrokontrollern.

För användning av EZ- knappen (SW3 i Bilaga B), hänvisas till originalfjärrkontrollens användarmanual.

6 DISKUSSION

I detta avslutande kapitel delas några allmänna tankar om projektet och vissa uppkomna problem och framtida utvecklingsförslag diskuteras. Slutligen presenteras våra tankar om projektets eventuella påverkan på samhället ur etik- och miljösynpunkt.

6.1 Komponentbeställningar

Något vi tidigt fick erfara, och något vi vill tipsa andra som ska göra sitt första stora projekt om är att se till att beställa komponenter tidigt och ha med i både budgetberäkning och planering att kompletteringsbeställningar lär komma att behövas även om ett noggrant förberedelsearbete genomförs. I vårt fall fick vi köpa de flesta komponenter två gånger, både på grund av kompabilitetsproblem och avsaknad av funktionalitet men framför allt då komponentbehovet förändrades under projektets gång, även om köpen var väl genomtänkta från början.

Dessutom saknas (i enlighet med Murphys lag) ofta småkomponenter som en nödvändig komponent med ett visst värde, även om ett till syntes tillräckligt lagerutbud finns att tillgå. Därav vill vi också rekommendera att innan arbetet börjar, förbereda en lista med leverantörer som kan leverera grundkomponenter snabbt och med låga fraktavgifter.

6.2 Regleringskänslighet

Under testkörningar visade sig bilens styrning vara ganska oprecis och i vissa lägen rent opålitlig, då bilen ibland slutade svara på styrningskommandon helt. Vid långsam körning stannade ofta bilen helt och fick startas om. Dessa oönskade styrningsegenskaperna tycks dock vara ett resultat av bilens konstruktion: Dels säkerhetsfunktioner som inte fungerar optimalt, dels underdimensionerad hårdvara som styrservon och växellådor.

Dessa problem ligger utanför omfånget av detta projekt, men har orsakat svårigheter vid tester av den framtagna styrenheten. Dock verkar känsligheten hos styrreglaget i originalfjärrkontrollen vara högre än vad det verkar, sett till bilens köregenskaper och mätningar gjorda under hårdvaruanalysen.

Den framtagna prototypen är designad utifrån detta för att ge en ”tillräcklig” precision vid styrning av radiostyrda fordon generellt, även om just denna bil inte kräver speciellt hög precision alls. Med andra ord tros inte bilens sämre styrförmåga ha påverkat den framtagna konstruktionen. Dock har detta begränsat den verkliga uppskattningen av styrenhetens noggrannhet vid test och anpassning markant, vilket gör att systemets noggrannhet eventuellt kan behöva vidare arbete vid implementering mot andra enheter.

I övrigt är hastighetsregleringen inte helt linjär i detta system, vilket kan ses i Figur 23 i avsnitt 4.3.1. I en optimal situation hade utspänningen och därmed bilens hastighet varit linjärt proportionell mot flexsensorernas böjning. Men resultatet har här ansetts vara ”nära nog” linjär och avvikelsen från linjäritet har ansetts dels vara väldigt liten i sammanhanget och dels inte vara märkbar i verkligheten.

6.3 Problemlösning

I denna sektion påtalas och förtydligas ett axplock av de problem som dykt upp under projektets gång, tillsammans med lösningsbeskrivningar.

6.3.1 Operationsförstärkare

Innan flexkretsen anslöts till rotationskretsen, lämnades kretsens utgångar oanslutna, för att kontrollera delsystemets funktion. Det visade sig då att kretsen inte levererade korrekt maximal utspänning: Maximalt uppmättes 2,8 V vid Vut2, vilket inte ansågs vara en godtagbar avvikelse från de önskade 3,3 V. Noggrannare analys av OP-förstärkarnas datablad avslöjade att tidigare uppfattning om att dessa skulle vara rail-to-rail var felaktig och att Mousers rail-to-rail-angivelse gällde på ingångssidan och inte utgångssidan [12 sid 6]. Dock visade sig OP-förstärkarna vara av typen single-rail, då de kunde leverera utsignaler nära 0 V.

Lösningen blev att använda batteripaketet direkt som spänningskälla till OP-förstärkarnas positiva matningsingång. Maximal utspänning från förstärkarna blev då tillräcklig för att nå den önskade spänningen på 3,3 V.

Eftersom endast OP-förstärkarna använder denna spänningskälla i flexkretsen, påverkas inte någon annan del av kretsen direkt av denna förändring. Teoretiskt kunde detta dock lett till att förstärkarna i sig kunde leverera en för hög utspänning om strax under 6 V. I denna krets har detta motverkats genom kretsdesignens funktionalitet: Den första förstärkarens utsignal begränsas av kretsens inresistans medan den sista förstärkarens utsignal aldrig kan bli högre än dess maximala ingångsspänning VDD, vilken är konstant 3,3V.

Värt att notera är dock att eftersom denna inspänning är oreglerad, kan oönskade effekter uppstå i både flex- och sensorkretsen då batteripaketets laddning minskar. Dels kommer operationsförstärkarna i flexsensorn till slut inte ge full utspänning, vilket gör att maximal acceleration inte kommer kunna uppnås hos bilen och dels kan Arduinon börja bete sig oväntat eller sluta fungera helt, då denna kräver en matningsspänning om minst 5 V.

Vidare är operationsförstärkarnas funktionalitet endast testade vid batteriets momentära laddning om ca. 5,8 V samtidigt som dess datablad [12 sid 6] endast specificerat V_{OH} vid VDD = 5,0 V. Databladet anger dessutom härvid att maximal utspänning V_{OH} kan vara som lägst 3,2 V vid 25 ºC lufttemperatur, men kan vara så låg som 3,0 V vid 0 ºC. detta innebär alltså att förutom en sjunkande inspänning från batteriet, kan även omgivande temperatur orsaka funktionsproblem. Detta potentiella problem har inte undersökts eller testats närmare under arbetets genomförande, då det inte setts som ett akut problem för projektets grundläggande genomförande.

6.3.2 Mikrokontroller och programmeringsval

Tidigt i projektet användes en mikrokontroller av typen PIC16F1829. Valet föll på denna då vi hade erfarenhet av en liknande mikrokontroller sedan tidigare samt att den uppfyllde kraven, nämligen hantering av I^2C och inbyggd DAC. För att upprätta I^2C kommunikationen var det först nödvändigt att skapa ett eget bibliotek för detta. Detta resulterade i att någon form av kommunikation ut från PIC:en fungerade, men aldrig det motsatta. Efter detta testades MPLAB Code Configuration (MCC), vilket är ett tillägg till MPLAB och är till för att underlätta hanteringen av kod och generera bibliotek för tex UART och I^2C. Koden som genererades var svår att få överblick över och p.g.a. tidsbrist fattades därpå beslutet att använda Arduino istället.

För att få en djupare förståelse för sensorerna har koden för Arduinon skrivits helt utan bibliotek för MPU6050. I framtiden hade bibliotek för MPU6050 med fördel kunnat implementeras för att få en stabilare och mer beprövad kod, samtidigt som avsaknaden av använda bibliotek underlättar eventuell konvertering till C eller annat programmeringsspråk för användning av

exempelvis PIC mikrokontroller istället för Arduino. En stor fördel att projektets mjukvara flyttades över till Arduino var att det underlättade att undersöka sensormodulen, vilket hade varit svårare med mikrokontrollern PIC16F1829.

Den använda modellen av Arduino har även en inbyggd 6-DOF rörelsesensor, vilken teoretiskt skulle kunna ersätta den externa IMU:n som använts i detta projekt. Dock har vi valt att ha den externa ändå, för att lättare kunna placera den på handen, separerad från övrig krets. Dessutom innebär detta att samma krets kan användas med en PIC mikrokontroller, istället för den dyrare Arduinon om så önskas. (En adapter skulle dock krävas, då PCB:n ej har anslutningar anpassade för PIC.)

6.3.3 Driftning

Gyroskop MEMS har ett allmänt känt problem med driftning. Driftning innebär att om sensorn placeras helt stilla så kommer vinkelvärdet ändå att ändras över tid. Fenomenet uppstår då sensorn är väldigt känslig för störningar och temperaturförändringar. Detta problem undveks genom att använda IMU:ns accelerometer och således inte använda gyroskopet alls.

6.4 Förbättringsförslag

Under arbetets gång har en del brister noterats och alla av dem har inte kunnat lösas inom tidsramen för projektet. Här delas några förslag för vidareutveckling av produkten, vissa för att lösa befintliga brister, andra för att motverka potentiella framtida fel.

6.4.1 Montering och PCB

Eftersom den designade PBC:n inte hann beställas i tid för leverans innan denna rapport färdigställdes har systemet endast testats och monterats med kopplingsdäck. Som följd av detta uppstod en hel del signalfel vid testkörning, orsakade av dålig kontakt. Både i form av glapp och lossnade kablar, då armen och därmed kopplingsdäcken och dess anslutna komponenter rör sig ganska mycket vid användning. Dessa problem ledde till mycket onödig och förvirrande felsökning under projektets gång.

För ett mer pålitligt handhavande, skulle det färdiga systemet behöva monteras mer permanent, antingen på PCB eller experimentplatta. Likväl skulle den externa rotationssensorn behöva fästas mer säkert på handen, då den vid tester nu vrider och flyttar sig något under användning vilket påverkar styrningen.

Under arbetets gång hade dessutom kopplingsdäck av högre kvalitet varit att föredra då liknande signalproblem även uppstod vid arbetsbänken som följd av dåligt konstruerade anslutningskontakter i de kopplingsplattor som användes.

Dessutom saknas ännu hållare för samtliga kretsar, som hittills satts direkt mot armen med gummiband. För säkrare användning bör alltså systemet monteras i eller på någon form av inbyggnadslåda och/eller fäste.

Vidare är den PCB som designats relativt stor och otymplig. Detta för att få plats med de hålmonterade komponenterna. För framtida utveckling rekommenderas att designa om denna PCB och då använda ytmonterade komponenter för mindre byggyta och smidigare hantering.

6.4.2 Finjustering av hastighetsreglage

Den implementerade trimpotentiometern RV1, som används för justering av hastighetsreglaget ger i dagsläget möjlighet till viss finjustering. Dess funktion är dock ganska begränsad och mest bara användbar för justering av drivningens vilopunkt: Om resistansen hos flexsensorn är sådan att viloläge är balanserat (bilen står stilla vid utsträckta sensorer både i framåt- och bakåtriktning), men att maxhastighet inte kan nås (för låg resistans vid maxläge), eller nås för tidigt (för hög resistans vid maxläge), kan detta inte justeras utan att påverka vilopunkten. Denna funktionalitet är avsiktligt vald sådan, för att undvika fler komponenter på kretsen, men är något som troligen bör förbättras i framtida vidareutveckling av systemet, så att både max- och minimilägen kan justeras individuellt.

Dess nuvarande funktionalitet att specifikt justera drivningens vilopunkt fungerar dock mycket bra och kan med fördel användas om man exempelvis inte kan räta ut sina fingrar helt på grund av ledsjukdomar, eller om flexsensorernas resistans förändras över tid.

6.4.3 Överspänningar till styrkretsen

En annan förbättring som kan göras vid framtida revidering av systemet är en säkerhetsfunktion som förhindrar överspänningar i systemet vid kritiska fel: I dagsläget kan exempelvis inte flexsensorns värde understiga 30k Ω, men simuleringar har visat att om dess resistans sjunker under ca. 10k Ω (exempelvis som följd av yttre åverkan på sensorn), kan V_{UT1} teoretiskt överskrida 3,3 V, vilket i sin tur eventuellt kan skada styrkretsen. Denna risk skulle kunna begränsas genom att exempelvis användning av zenerdioder, vilka leder överspänningar från V_{UT1} till jord.

V_{UT2} är dock redan skyddad mot denna typ av fel: Om inresistansen minskar, går Vut2 mot noll och om flexsensorns resistans skulle öka (exempelvis genom att de dras loss från kretsen) går V_{UT2} mot VDD, som är reglerad till 3,3 V.

6.4.4 Systembugg vid handrotation

Om MPU6050-sensormodulen vrids över 90° i X-led (pitch) kommer bilen att svänga höger eller vänster (dvs Y-led, roll) även om användaren inte vill manövrera bilen i någon av dessa riktningar. Detta är ingen önskad effekt utan en bugg som upptäcktes sent in i projektet. P.g.a. tidsbrist är buggen kvar och skulle förmodligen kunna motverkas genom användning av gyroskopet.

Efter att hela prototypen var ihopsatt och testad tillsammans med RC-bilen kunde det ha varit intressant att testa ändra känslighetsfaktorn för accelerometern samt viktningen i filtret för att se för att se om en mer reaktiv styrupplevelse hade kunnat konstruerats. Detta blev inte gjort på grund av tidsbrist.

6.4.5 Tvärt riktningsomslag

En effekt som eventuellt kan slita på bilens mekaniska system vid frekvent användning är det faktum att man mycket snabbt kan gå från full gas framåt till full back. Detta innebär att motorn ger ett snävt spänningsomslag och bromsar in mycket hårt, vilket kan slita på både motordelar och växellåda. Någon djupdykning i frågan har inte gjorts då det är utanför projektets område, men diskussionen kom upp vid implementering av multiplexern:

Eftersom denna har två lediga insignalsportar och en oanvänd selektorsignal, kunde ett ”övergångsläge” kunna implementeras här. I nuläget väljs selektorsignalen ”00” eller ”01” vilket leder antingen V_{UT1} eller V_{UT2} till multiplexerns utsignal. Vinkling av handen växlar mellan dessa via Arduinon och gör att signalen till styrkretsen kan gå från +3,3 V till 0 V omedelbart. Härvid kunde exempelvis selektorn ”1X” lagt en konstant spänning om 1,65 V mot multiplexerns utgång under en förutbestämd tidsperiod vid samma lutning av handen. Därmed skulle utsignalen och således bilens motor kunna ”mellanlanda” på en mellanliggande nivå en kort stund innan extremläget nås. Om dessutom båda lediga signaler nyttjats (selektor ”10” och ”11”), kunde ytterligare ett mellansteg implementerats, vilket skulle skona hårdvaran något ytterligare.

6.5 Etiska och ekologiska aspekter

Bortsett val av komponenttillverkare och produktionsmetod, anses detta projekt inte i någon större utsträckning ha någon direkt negativ påverkan, varken etiskt eller ekologiskt. Det som under produktionsprocessen kan tänkas påverka vårt ekosystem är utsläpp vid komponentproduktion samt transport av komponenter.

Som färdig produkt kan viss ekologisk påverkan förekomma i form av något ökad konsumtion av elektronikprodukter för nöjes skull. Dock är produkten utformad för en specifik hobbygrupp och tros användas i väldigt liten utsträckning.

Däremot kan produkten i förlängningen tänkas ha en positiv samhällelig påverkan, då produkten kan vidareutvecklas till att underlätta för människan och bidra till ökad livskvalitet.

Exempel på sådana tillämpningar kan vara:

- Styrning av robotar för hantering av farligt gods.
- Styrning av RC- fordon och andra nöjesprodukter för människor som av medicinska skäl inte han hantera sedvanliga handkontroller.
- Styrning av TV- spel eller drönare för en mer inlevelserik spel- eller körupplevelse.
- Ersättning eller komplettering av industriella styrreglage för ökad ergonomi.

Vidare skulle tekniken även kunna användas i forskningssyfte, exempelvis för att övervaka och undersöka hur man rör sig när man utför specifika arbetsuppgifter, vilket i sin tur kan leda till förbättrad forskning inom exempelvis ergonomi.

6.5.1 Hobbyanpassat projekt

Projektet är genomfört hos Consat Engineering AB, men har på grund av covid-19-restriktioner i stor utsträckning utvecklats hemifrån och därmed utan avancerad, professionell utrustning. Oscilloskop var dock nödvändigt vid ett moment, för undersökning av uppkomna störningar i kretsen. I övrigt kunde denna ersatts av en multimeter om oscilloskop inte funnits att tillgå.

Detta faktum, tillsammans med den färdiga produktens användningsområde och relativt låga utvecklingskostnad, gör att projektet kan lämpa sig mycket bra i utbildningssyfte för hobbyister. Därmed kan projektet i sig förlängningen anses bidra till positiv samhällelig utveckling. En kopia av denna rapport kommer därför även att lämnas till hobbyföreningen Lindholmens Makerspace så fler teknikintresserade ungdomar och studenter lätt kan ta del av dess innehåll.

REFERENSER

[1] Fisher C J. ”Using An Accelerometer for Inclination Sensing” [Online]. Thief River Falls: Convergence Promotions LLC; 2011-05-06, Tillgänglig: https://www.digikey.com/en/articles/using-an-accelerometer-for-inclination-sensing (Hämtad 2021-05-25).

[2] Baker B. ”Apply Sensor Fusion to Accelerometers and Gyroscopes” [Online]. Digi-Keys nordamerikanska redaktörer; 2018-01-30, Tillgänglig: https://www.digikey.se/sv/articles/apply-sensor-fusion-to-accelerometers-and-gyroscopes (Hämtad 2021-05-25).

[3] TDK, ”MPU-6000 and MPU-6050 Product Specification Revision 3.4” [Online]. Tillgänglig: https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf (Hämtad 2021-05-25).

[4] Wikipedia, ”Mercury_switch.” [Online]. Tillgänglig: https://en.wikipedia.org/wiki/Mercury_switch (Hämtad 2021-05-25).

[5] Last Minute Engineers, “Interfacing Flex Sensor with Arduino” [Online]. Tillgänglig: https://lastminuteengineers.com/flex-sensor-arduino-tutorial/ (Hämtad 2021-05-25).

[6] Heath J. “Amplifiers: What do rail-to-rail and single supply mean?” [Online]. 2017-10-10, Tillgänglig: https://www.analogictips.com/amplifiers-rail-to-rail-single-supply-mean/ (Hämtad: 2021-05-25).

[7] Nordlund L, Wiklund I. Grundläggande elektronik. 3 upplaga. Stockholm: Liber, 2012.

[8] Kjell och Company, “Vad är Arduino och Genuino?” [Online]. Tillgänglig: https://www.kjell.com/se/kunskap/hur-funkar-det/arduino/introduktion/vad-ar-arduino-och-genuino (Hämtad 2021-05-25).

[9] Arduino, “Arduino Nano 33 IoT” [Online]. Tillgänglig: https://store.arduino.cc/arduino-nano-33-iot (Hämtad 2021-05-25).

[10] Wikipedia, ”Hysteres.” [Online]. Tillgänglig: https://sv.wikipedia.org/wiki/Hysteres (Hämtad 2021-05-25).

[11] Mouser, ”Flex Sensor” [Online.] Tillgänglig: https://www.mouser.se/datasheet/2/381/FLEX_SENSOR_DATASHEET_09-1203917.pdf (Hämtad 2021-05-25).

[12] Texas Instrument, ”TLC271, TLC271A, TLC271BLin CMOS PROGRAMMABLE LOW-POWER OPERATIONAL AMPLIFIERS” [Online]. Tillgänglig: https://www.ti.com/general/docs/suppproductinfo.tsp?distId=26&gotoUrl=http%253A%252F%252Fwww.ti.com%252Flit%252Fgpn%252Ftlc271b (Hämtad 2021-05-25).

[13] FAB Modules, ”Difference between ’Single-Supply’ and ’Rail-to-Rail’ Opamps” [Online]. Tillgänglig: https://fabmodules.com/difference-between-single-supply-and-rail-to-rail-opamps/ (Hämtad 2021-05-25).

[14] Wikipedia, ” Exponential smoothing” [Online]. Tillgänglig: https://en.wikipedia.org/wiki/Exponential_smoothing (Hämtad 2021-05-25).

[15] Wikipedia, ” Thévenins teorem” [Online]. Tillgänglig: https://sv.wikipedia.org/wiki/Th%C3%A9venins_teorem (Hämtad 2021-05-25).

FIGURFÖRTECKNING

Figur 1. Använd RC- bil, Traxxas E-Maxx. 2
Figur 2. Traxxas TQ-6516 fjärrkontroll 2
Figur 3: Illustrationsbild över en MEMS accelerometer. 4
Figur 4: Illustrationsbild över ett MEMS gyroskop. 5
Figur 5. En MPU6050. 6
Figur 6. Kretsschema över differensförstärkare 8
Figur 7. Principen för ett system som använder I2C. 10
Figur 8. Spänningsdelare 12
Figur 9. Insida av fjärrkontroll. Styrkrets till vänster, inställningskrets till höger, batteripaket i botten. 14
Figur 10. Detaljbild styrkretsens baksida. 15
Figur 11. Styrkortets framsida, anslutningspunkter till vänster. 15
Figur 12. Diagram som visar spänningen över TH vid olika lägen på hastighetsreglaget. 16
Figur 13. Illustrationsbild över rotationen kring Y- och Z-axel. 20
Figur 14. Hämtade värden för Y-axeln. 20
Figur 15. Kontroll av det kalkylerade Y-värdet. 20
Figur 16. Vibrationstest av accelerometer 21
Figur 17. Vibrationstest av accelerometer. Röd graf är rådata och gul graf är filtrerad data. 22
Figur 18. De två if-satserna som används för att skapa hysteresen. 22
Figur 19. Ett första utkast av ett blockdiagram över systemets funktionalitet. 23
Figur 20. En grov skiss av en idéversion av flexkretsen. 24
Figur 21. Kretsschema för flexsensorn ur LTspice. 24
Figur 22. Simulering av flexkretsen i LTspice. Kurvan visar utspänningen Vut1 (V) i förhållande till resistans Rsens (Ω). 25
Figur 23. Kretsschema samt simuleringsresultat för Vut1 (V) (grön kurva) och Vut2 (V) (rosa kurva) som funktion av resistans Rsens (Ω). 26
Figur 24. Kretsschema över rotationskretsen. 27
Figur 25. Framtagen PCB- design. (Consat- loggan använd med tillstånd) 28
Figur 26. Simulerad modell av färdigt kretskort. 28
Figur 27. Uppmätt brus vid styrkretsens utgång 29
Figur 28. Uppmätt utsignal efter att störning filtrerats bort. 29
Figur 29. Flexsensorns beteende vid böjning och uträtning. 30
Figur 30. Flexsensorer monterade på en handske. 30
Figur 31. Hela systemet monterat på hand och arm. 31
Figur B.1. Komplett kretsschema över framtaget styrsystem Bilagor, sid. 2
Figur C.1. Simulerad krets, framsida Bilagor, sid. 3
Figur C.2. Simulerad krets, baksida. Bilagor, sid. 3
Figur C.3. Alla PCB- lager utan utfyllda jordningslager Bilagor, sid. 3
Figur C.4. Kopparlager, framsida. Bilagor, sid. 4
Figur C.5. Kopparlager, baksida. Bilagor, sid. 4
Figur C.6. Komponentplacering Bilagor, sid. 4

Samtliga figurer i denna rapport är producerade av Stefan Ros och Robert Lundström 2021 och licenseras enligt CC BY-NC-ND.

TABELLFÖRTECKNING

Tabell 1. Pinout och kortbeskrivning av respektive pinne för modulen MPU6050. 6
Tabell 2. Specifikationerna för Arduino Nano 33 IoT. 9
Tabell 3. Händelseförlopp för 1 bytes läsning. 10
Tabell 4. Händelseförlopp för 1 bytes skrivning. 11
Tabell 5. Exempel på fartförändring vid växling mellan fram och back på styrreglage. Exemplen gäller då bilen redan har en ingångshastighet enligt fartreglagets första läge. 14
Tabell 6. Uppmätt spänning vid anslutningspunkterna TH och ST på fjärrkontrollens styrkrets. 15
Tabell 7. De register och tillhörande adresser som sattes för att kunna använda accelerometer. 19
Tabell 8. Framtagna ekvationer för att kalibreringen ska fungera önskvärt. 20
Tabell A.1. BOM för utvecklad systemkrets. Exporterad från KiCad. ..... Bilagor, sid. 1

BILAGA A. MATERIALLISTA

I tabell A.1 nedan visas en materiallista (BOM, Bill Of Materials) som exporterats ur KiCad. Denna visar de komponenter som använts i den resulterande kretsen. Utöver de där noterade komponenterna har tre stycken flexsensorer FS-L-0112-103-ST, kopplingsdäck, arbetshandske och kopplingstråd tillkommit under konstruktionen av systemet.

Tabell A.1. BOM för utvecklad systemkrets. Exporterad från KiCad.

part	description	ref	value5	footprint
C	Capacitor	C1	10u	Capacitor_THT:CP_Radial_D5.0mm_P2.50mm
C	Capacitor	C2	10u	Capacitor_THT:CP_Radial_D5.0mm_P2.50mm
C	Capacitor	C3	100n	_project_footprints:C_104_Disc_D5.0mm_W2.5mm_P3.50mm
C	Capacitor	C4	100n	_project_footprints:C_104_Disc_D5.0mm_W2.5mm_P3.50mm
C	Capacitor	C5	100n	_project_footprints:C_104_Disc_D5.0mm_W2.5mm_P3.50mm
C	Capacitor	C6	100n	_project_footprints:C_104_Disc_D5.0mm_W2.5mm_P3.50mm
LED	Light emitting diode	D1	LED_Y	LED_THT:LED_D5.0mm_Clear
LED	Light emitting diode	D2	LED_R	LED_THT:LED_D5.0mm_Clear
LED	Light emitting diode	D3	LED_G	LED_THT:LED_D5.0mm_Clear
Conn_02x01	Generic connector, double row, 02x01	J1	FS_conn	Connector_PinSocket_2.54mm:PinSocket_1x02_P2.54mm_Vertical
Conn_02x01	Generic connector, double row, 02x01	J2	3V3_Conn	Connector_PinSocket_2.54mm:PinSocket_1x02_P2.54mm_Vertical
Conn_02x01	Generic connector, double row, 02x01	J3	BATT_Conn	Connector_PinSocket_2.54mm:PinSocket_1x02_P2.54mm_Vertical
Conn_02x01	Generic connector, double row, 02x01	J4	IO_Conn	Connector_PinSocket_2.54mm:PinSocket_1x02_P2.54mm_Vertical
Conn_01x04	Generic connector, single row, 01x04	J5	Extra_Conn	Connector_PinSocket_2.54mm:PinSocket_1x04_P2.54mm_Vertical
Conn_01x04	Generic connector, single row, 01x04	J6	Extra3V3_Conn	Connector_PinSocket_2.54mm:PinSocket_1x04_P2.54mm_Vertical
Conn_01x04	Generic connector, single row, 01x04	J7	ExtraGND_Conn	Connector_PinSocket_2.54mm:PinSocket_1x04_P2.54mm_Vertical
Conn_01x01	Generic connector, single row, 01x01	J8	SW_Conn	Connector_PinSocket_2.54mm:PinSocket_1x01_P2.54mm_Vertical
Conn_01x02	Generic connector, single row, 01x02	J9	JMP	Connector_PinHeader_2.54mm:PinHeader_1x02_P2.54mm_Vertical
R	Resistor	R1	10k	Resistor_THT:R_Axial_DIN0207_L6.3mm_D2.5mm_P10.16mm_Horizontal
R	Resistor	R10	330	Resistor_THT:R_Axial_DIN0207_L6.3mm_D2.5mm_P10.16mm_Horizontal
R	Resistor	R11	10k	Resistor_THT:R_Axial_DIN0207_L6.3mm_D2.5mm_P10.16mm_Horizontal
R	Resistor	R12	330	Resistor_THT:R_Axial_DIN0207_L6.3mm_D2.5mm_P10.16mm_Horizontal
R	Resistor	R13	200	Resistor_THT:R_Axial_DIN0207_L6.3mm_D2.5mm_P10.16mm_Horizontal
R	Resistor	R2	20k	Resistor_THT:R_Axial_DIN0207_L6.3mm_D2.5mm_P10.16mm_Horizontal
R	Resistor	R3	39k	Resistor_THT:R_Axial_DIN0207_L6.3mm_D2.5mm_P10.16mm_Horizontal
R	Resistor	R4	6.8k	Resistor_THT:R_Axial_DIN0207_L6.3mm_D2.5mm_P10.16mm_Horizontal
R	Resistor	R5	20k	Resistor_THT:R_Axial_DIN0207_L6.3mm_D2.5mm_P10.16mm_Horizontal
R	Resistor	R6	10k	Resistor_THT:R_Axial_DIN0207_L6.3mm_D2.5mm_P10.16mm_Horizontal
R	Resistor	R7	10k	Resistor_THT:R_Axial_DIN0207_L6.3mm_D2.5mm_P10.16mm_Horizontal
R	Resistor	R8	10k	Resistor_THT:R_Axial_DIN0207_L6.3mm_D2.5mm_P10.16mm_Horizontal
R	Resistor	R9	10k	Resistor_THT:R_Axial_DIN0207_L6.3mm_D2.5mm_P10.16mm_Horizontal
R_POT	Potentiometer	RV1	10k	_project_footprints:Potentiometer_103_bad
SW_Push	Push button switch, generic, two pins	SW1	SW_Cal	Button_Switch_THT:SW_PUSH_6mm_H5mm
SW_Push	Push button switch, generic, two pins	SW2	SW_RE	Button_Switch_THT:SW_PUSH_6mm_H5mm
SW_Push	Push button switch, generic, two pins	SW3	SW_EZ	Button_Switch_THT:SW_PUSH_6mm_H5mm
TestPoint_Alt	test point (alternative shape)	TP1	TP_Vin	TestPoint:TestPoint_THTPad_2.5x2.5mm_Drill1.2mm
TestPoint_Alt	test point (alternative shape)	TP2	TP_Vut1	TestPoint:TestPoint_THTPad_2.5x2.5mm_Drill1.2mm
TestPoint_Alt	test point (alternative shape)	TP3	TP_Vut2	TestPoint:TestPoint_THTPad_2.5x2.5mm_Drill1.2mm
MPU6050_IMU	8- pin connected IMU	U1	MPU6050_IMU	Connector_PinSocket_2.54mm:PinSocket_1x08_P2.54mm_Vertical
TLC271BCP	OPAmp	U2	TLC271BCP	Package_DIP:DIP-8_W7.62mm
Arduino-NANO-33	Arduino NANO 33	U3	Arduino-NANO-33	Module:Arduino_Nano
TLC271BCP	OPAmp	U4	TLC271BCP	Package_DIP:DIP-8_W7.62mm
74HC4052	Dual Analog Multiplexer 4 to 1 line.	U5	74HC4052	Package_DIP:DIP-16_W7.62mm_Socket

Beräknad total produktionskostnad för ett färdigbyggt system inkluderande ovanstående komponenter och PCB-produktion blev cirka 1450 kronor inklusive moms, frakt-, och tullavgifter. (Fraktavgifter härvid gäller om endast en komponentbeställning hade behövt göras)

Vidare införskaffades även följande komponenter under arbetets inledande fas:

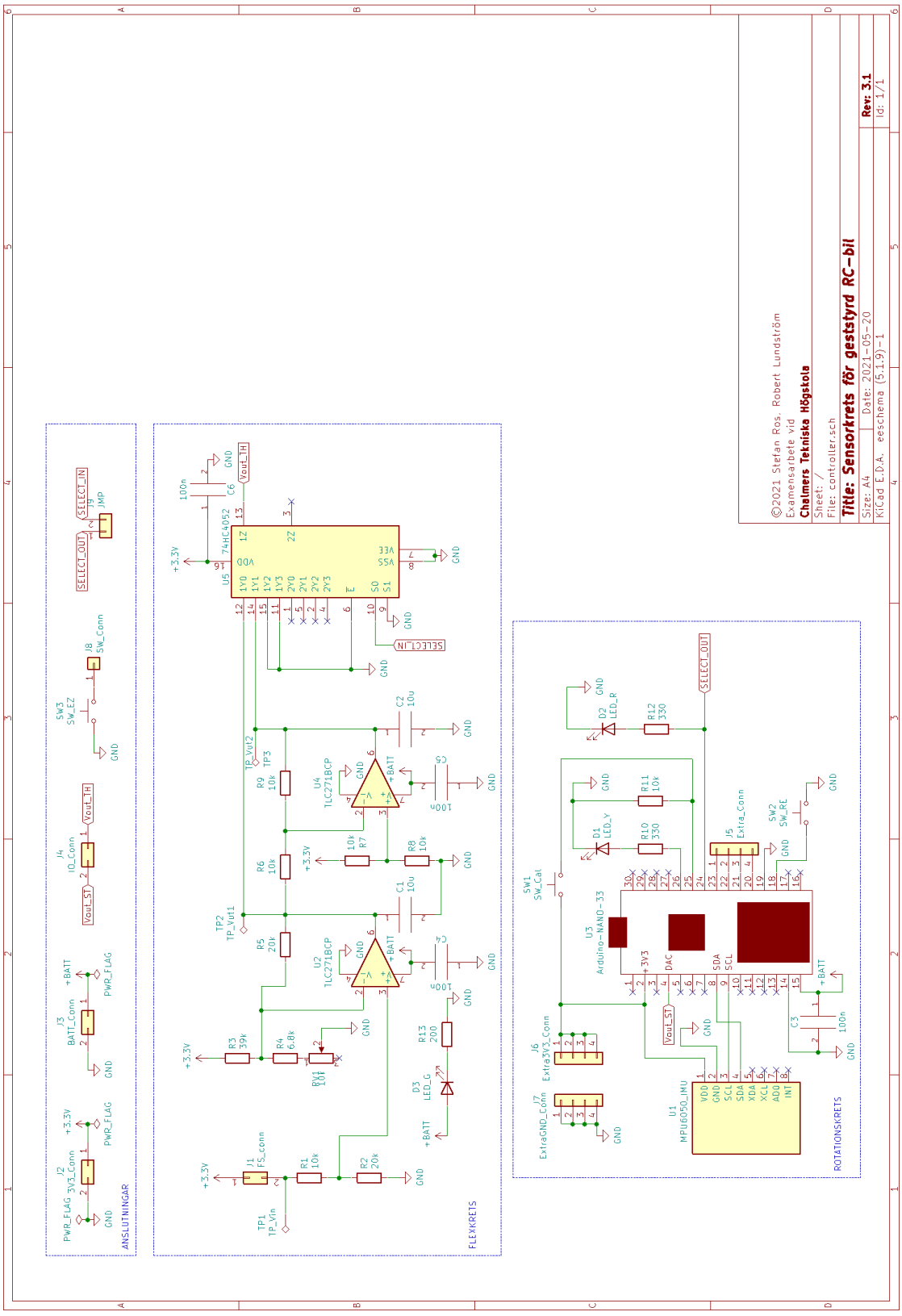
- Mikrokontroller PIC18F16Q41 (Användes ej pga inkompabilitet)
- Mikrokontroller PIC16F1829 (Ersattes med Arduino)
- Gyroskopmodul Adafruit 1032 (Ersattes med IMU)

Beräknad total utvecklingskostnad för projektet inklusive komponenter som ej användes i slutprodukten blev: 1951 kronor inklusive moms, (verklig) frakt- och tullavgifter.

De tre största utgiftsposterna bortsett fraktavgift var Arduino, 242:- (jmf. med PIC16F1829 26:-/st.), Flexsensorer, 213:-/st. samt PCB- produktion om 70:- (för min. antal om 5 st. kort).

BILAGA B. KRETSSCHEMA

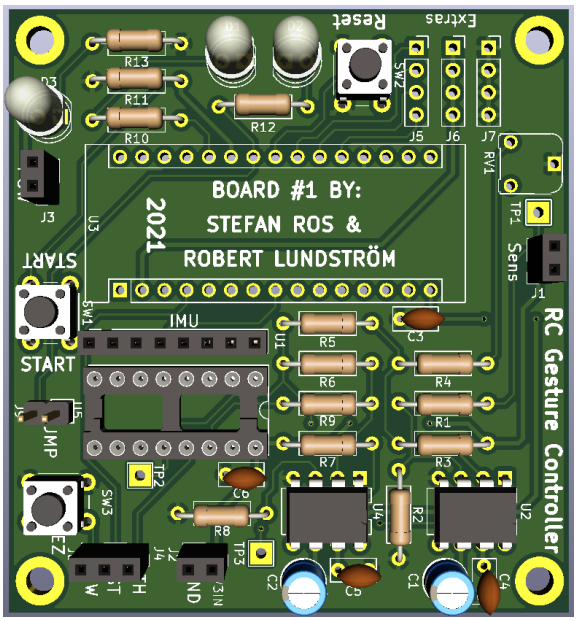
Komplett kretsschema, exporterad från KiCad, över flexkrets och rotationskrets samt dess anslutningspunkter.



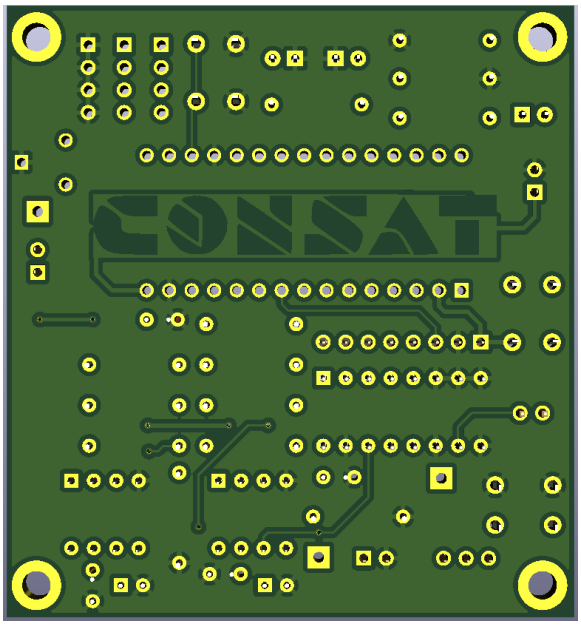
Figur B.1. Komplett kretsschema över framtaget styrsystem

BILAGA C. PCB- RITNING

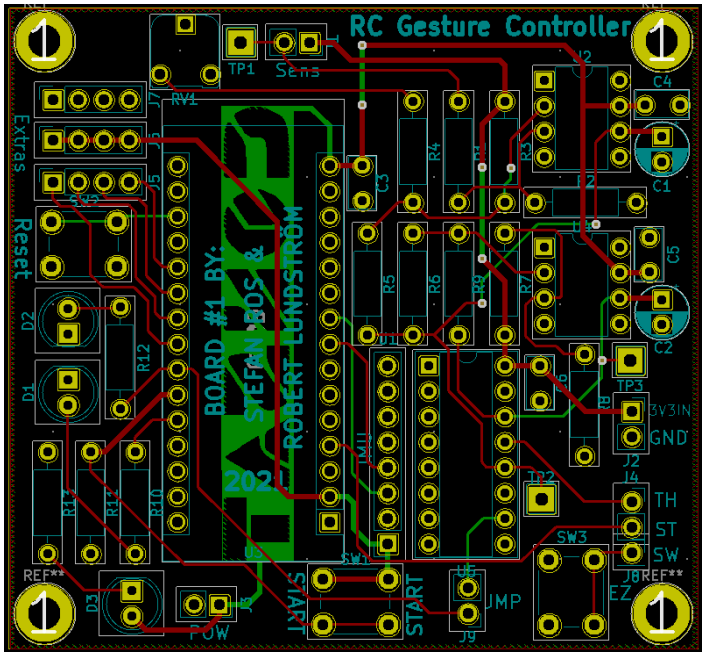
Här visas genererade bilder och ritningar över framtagen PCB- design.



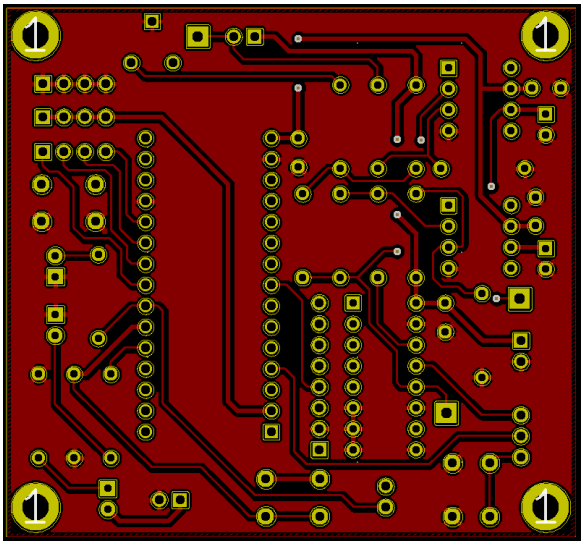
Figur C.1. Simulerad krets, framsida.



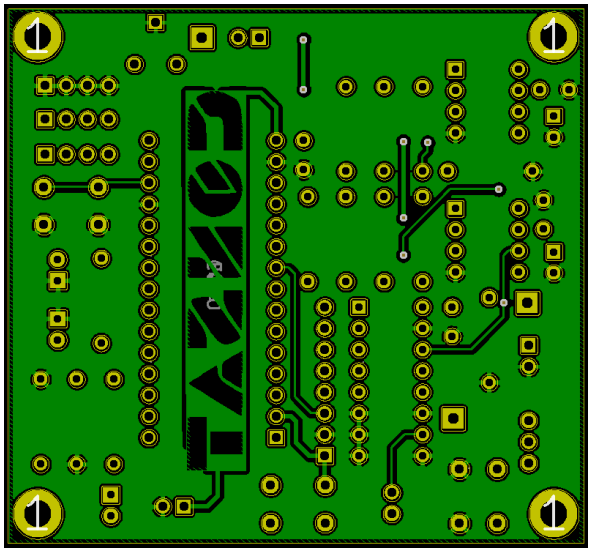
Figur C.2. Simulerad krets, baksida.



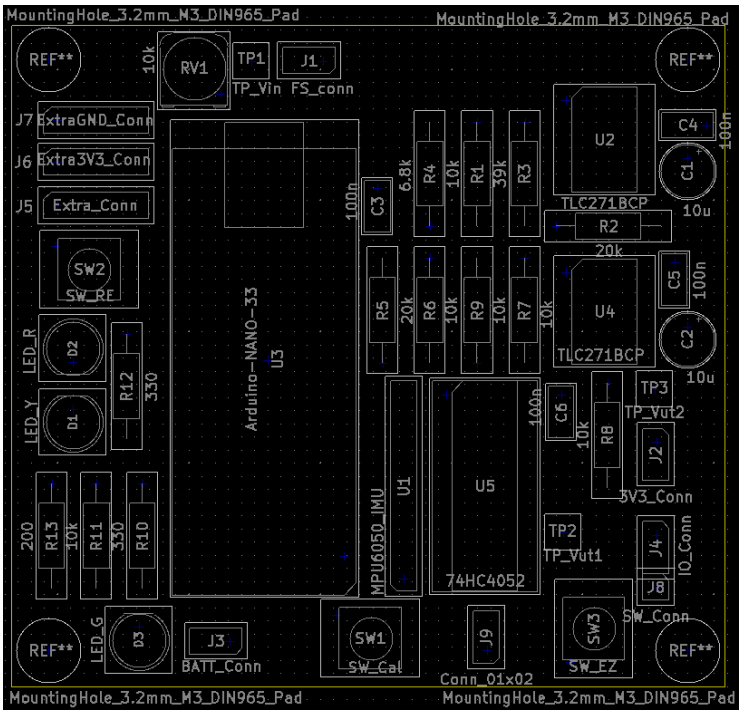
Figur C.3. Alla PCB- lager utan utfyllda jordningslager



Figur C.4. Kopparlager, framsida.



Figur C.5. Kopparlager, baksida.
Consat- loggan använd med tillstånd.



Figur C.6. Komponentplacering

BILAGA D. PROGRAMKOD

```
/*
 * Author:        Robert Lundström and Stefan Ros
 * Created On:      2021-04-14
 * Last Modified on:  2021-05-21
 *
 * This code is used for a DIY hand gesture controller.
 * The code is written and tested with an Arduino Nano 33 IoT
 * and a MPU6050 sensor. OBS! When using the wire.h library
 * the pin A5 and A4 are automatically used as SCL and SDA.
 * Also, when using wire.h the clock frequency for I2C communication,
 * is set to 100 KHz by default.
 *
 * In this project the DMP where not in use,
 * and the MPU address in use where the low (0x68). This means
 * that only four wires were needed for the MPU6050, namely
 * SCL, SDA, Vcc and GND.
 *
 * The exponential smoothing filter is adapted for the
 * following MPU6050 settings:
 *    * Bandwidth = 21 Hz.
 *    * FSR = +- 8g.
 *
*/


#include <Wire.h>

//--------------[ MPU6050 Register Map ]----------------------------------------
#define MPU_Address 0x68          //MPU 7-bit address. AD0 set to low
#define PWR_MGMT_1  0x6B          //Register 107 -Power Management 1
#define CONFIG      0x1A          //Register 26 -Configuration
#define ACCEL_CONFIG 0x1C         //Register 28 -Accelerometer Configuration

//--------------[ Variables used for getting Raw data from accelerometer ]------
static int16_t  acc_X_raw;
static int16_t  acc_Y_raw;
static int16_t  acc_Z_raw;
static long   acc_X;
static long   acc_Y;
static long   acc_Z;

//--------------[ Variables used for getteting the offset ]----------------------

static int16_t acc_X_offset_ptr = 0;
static int16_t acc_Y_offset_ptr = 0;
static int16_t acc_Z_offset_ptr = 0;
static int16_t forward = 1;


//--------------[ Variables used for filtering ]-----------------------------

static float  last_x_value;
static float  last_y_value;
static float  last_z_value;
static float  w = 0.1;    //w is the weighting factor in the range in percent
static int16_t  filter_X;
static int16_t  filter_Y;
static int16_t  filter_Z;
static int16_t  DAC_out_Y;


//--------------[ Variables for I/O ]-------------------------------------

static int Offset_Button = 7;
static int Offset_LED = 8;
static int MUX_out = 6;
static int buttonState = 0;
static int StartButton = 0;


//--------------[ Setups ]-----------------------------------------------

void setup() {

  Serial.begin(115200);
  Wire.begin();                 //Join the bus as master
  setupMPU();
  setup_I_O();
  set_last_read_value_data(0, 0, 0);

}
```

```
83
84  void setup_I_O(){
85    pinMode(MUX_out, OUTPUT);   // Set digital pin 6 as output
86    pinMode(Offset_Button, INPUT);  // Set digital pin 7 as input
87    pinMode(Offset_LED, OUTPUT);  // Set digital pin 8 as output
88
89    }
90
91
92  /*
93   * Wake Up + Setting The Sample (Data) Rate
94   * Wake up the MPU (Sleep = '0')
95   * and setting the Clock Source PLL
96   * with X axis gyroscope reference.
97   *
98   */
99  void setupMPU(void){
100
101   delay(100);
102   Write_to_Slave(MPU_Address, PWR_MGMT_1, 0x09);    //0x01 //Wake up the MPU (Sleep = '0')
103                                 //and setting the Clock Source PLL with
104                                 //X axis gyroscope reference, 0x09 = Temp disable
105   Write_to_Slave(MPU_Address, CONFIG, 0x04);      //0x04 => Bandwidth = 21 Hz.
106   Write_to_Slave(MPU_Address, ACCEL_CONFIG, 0x10);  //(0x10 => FSR = +- 8g) (+- 8g gives = 4096 LSB/g)
107 }
108
109
110 //--------------[ Functions ]-------------------------------------------------
111
112 void Write_to_Slave(char Address, char Register_Address, char Setting){
113
114   Wire.beginTransmission(Address);        //Start- and Write-bit included in the function (S, AD + W)
115   Wire.write(Register_Address);         //Puts the register adress on the bus (RA).
116                               //The functions check ACK/NACK
117   Wire.write(Setting);              //Puts the data on the bus.
118   Wire.endTransmission(true);
119 }
120
121
122 /*
123  * This function reads from Start_from_Register
124  * + next 4 register
125  */
126 void Read_from_Slave(char Address){
127
128   Wire.beginTransmission(Address);
129   Wire.write(0x3B);                           // Writing: starting with register 0x3B (ACCEL_XOUT_H)
130   Wire.endTransmission(false);                // The parameter indicates that the
131                               //Arduino will send a restart.
132                               //As a result, the connection is kept active.
133   Wire.requestFrom(MPU_Address, 6, true);       // Request a total of 6 registers: (A
134                               //CCEL_XOUT_H) to (ACCEL_ZOUT_L).
135   acc_X_raw = Wire.read()<<8 | Wire.read();
136   acc_Y_raw = Wire.read()<<8 | Wire.read();
137   acc_Z_raw = Wire.read()<<8 | Wire.read();
138 }
139
140
141 /*
142  * The Calculate_Offset function is
143  * taking 1000 samples of raw data,
144  * adding the values and then dived it
145  * by 1000 in order to calculate the offset.
146  */
147 void Calculate_Offset(int16_t *acc_X_offset_ptr, int16_t *acc_Y_offset_ptr, int16_t *acc_Z_offset_ptr){
148
149   long acc_X_raw_offset = 0;
150   long acc_Y_raw_offset = 0;
151   long acc_Z_raw_offset = 0;
152
153   for(int calc = 0; calc <1000; calc ++){
154
155     Read_from_Slave(MPU_Address);
156     acc_X_raw_offset += acc_X_raw;
157     acc_Y_raw_offset += acc_Y_raw;
158     acc_Z_raw_offset += acc_Z_raw;
159     delay(3);             //To avoid repeated measures
160   }
161
162   *acc_X_offset_ptr = (acc_X_raw_offset/1000);
163   *acc_Y_offset_ptr = (acc_Y_raw_offset/1000);
164   *acc_Z_offset_ptr = (acc_Z_raw_offset/1000);
```

```
  forward = 1;

}


/*
 * The set_last_read_value_data function
 * is used to store vules from the accelerometer.
 * The is used in the exponential smooth filter.
 */
void set_last_read_value_data( float x, float y, float z) {

  last_x_value = x;
  last_y_value = y;
  last_z_value = z;

}


/*
 * The get_last_value functions are used to
 * get the stored values. Those functions are used
 * in the exponential smooth filter.
 */
inline float get_last_x_value() {
  return last_x_value;}
inline float get_last_y_value() {
  return last_y_value;}
inline float get_last_z_value() {
  return last_z_value;}


//--------------[ Main program ]----------------------------------------------------

void loop() {

  StartButton = digitalRead(Offset_Button);     //Used to start the main loop. Pushed once.

  if (StartButton == HIGH){

    while(1){

      buttonState = digitalRead(Offset_Button);

      if (buttonState == HIGH){
        digitalWrite(Offset_LED, 1);  //This led is used to indicate that offset calulation is active

        Calculate_Offset(&acc_X_offset_ptr, &acc_Y_offset_ptr, &acc_Z_offset_ptr);

      }

      else {

      digitalWrite(Offset_LED, 0);

      //----------[ Getting the X acc ]-----------

        Read_from_Slave(MPU_Address);

        acc_X = (acc_X_raw - acc_X_offset_ptr);


      //----------------[ +- 90degree check for Y_acc ]---------------------------------------
        /*
         * If the sensor is turned more the +-90 degree in Y-axis
         * this part will keep the sensor from start working backwards,
         * using the Z-axis as refrence.
         */

        //1 quadrant: Y_raw positive , Z_raw positive
        if (acc_Y_raw > 0 and acc_Z_raw > 0 ){

          acc_Y = (acc_Y_raw - acc_Y_offset_ptr);

        }

        //2 quadrant: Y_raw negative, Z_raw positive
        else if ( acc_Y_raw < 0 and acc_Z_raw > 0)   {

          acc_Y = acc_Y_raw - acc_Y_offset_ptr;

        }
```

```
247
248         //3 quadrant: Y_raw negative, Z_raw negative
249         else if (acc_Y_raw < 0 and acc_Y_raw < 0){
250
251           acc_Y = 2*(-4096-(4096 - abs(acc_Y_raw)) - (acc_Y_offset_ptr));
252
253         }
254
255         //4 quadrant: Y_raw positive, Z_raw negative
256         else if (acc_Y_raw > 0 and acc_Z_raw < 0){
257
258           acc_Y = 2*(4096+(4096 - abs(acc_Y_raw)) - (acc_Y_offset_ptr));
259         }
260
261
262         if ((acc_Y <= -4096) || ((acc_Y <= -4096)  & (acc_Y_raw > 0))){
263         acc_Y = -4096;
264         }
265
266         if ((acc_Y >= 4096) || ((acc_Y >= 4096)  & (acc_Y_raw < 0))) {
267         acc_Y = 4096;
268         }
269
270
271       //---------------[Filtering the result]------
272         /*
273          * An exponetial smoothering filter. The filter
274          * is getting the last saved value and uses it to
275          * calculate the next value. The variable w is used as
276          * smoothing factor. OBS! w need to be: (0 > w > 1)
277          */
278         filter_X = (w*acc_X) + ((1-w)* get_last_x_value());
279         filter_Y = (w*acc_Y) + ((1-w)* get_last_y_value());
280         filter_Z = (w*acc_Z) + ((1-w)* get_last_z_value());
281
282         for(int32_t x = 0; x < 10000; x++){
283
284           set_last_read_value_data(filter_X, filter_Y, filter_Z);
285
286         }
287
288       //----------------[ Setting the MUX backward/forward with hysteresis ]-------------------------
289         /*
290          * This part is setting the signal (TH) to the MUX.
291          * The forward variable is used for the hysteresis.
292          * In order to wider or narrow the range of the hysteresis,
293          * one can change the values in the if-cases.
294          */
295
296         //Backward
297         if ((forward == 0) && (filter_X > 2900)) {
298           digitalWrite(MUX_out, 1);
299           forward = 1;
300
301         }
302         //Forward
303         else if ((forward == 1) && (filter_X <= 2100)) {
304           digitalWrite(MUX_out, 0);
305           forward = 0;
306         }
307
308       //-----------------[ DAC out ]-----------------------------------------------
309         /*
310          * 0x10 => FSR = +- 8g. This give: 4096 LSB/g.
311          * The DAC resolution is 256 bits (by default)
312          * This give an interval of 4096/256 = 16.
313          * DAC out is using the DAC0-pin.
314          * OBS! +- 4096. DAC can only handle postivie values.
315          * So, first add 4096 to avoid negative values.
316          * Then dived it by 2*16=32.
317          */
318
319         DAC_out_Y = ((filter_Y + 4096)/32);
320         analogWrite(DAC0, DAC_out_Y);
321       //---------------------------------------------------------------------------
322
323         delay(10);
324       }
325     } //End of "while(1)"
326   } //End of "if (StartButton == HIGH)"
327 }
328
```

INSTITUTIONEN FÖR ELEKTROTEKNIK
CHALMERS TEKNISKA HÖGSKOLA
Göteborg, Sverige 2021
www.chalmers.se

CHALMERS