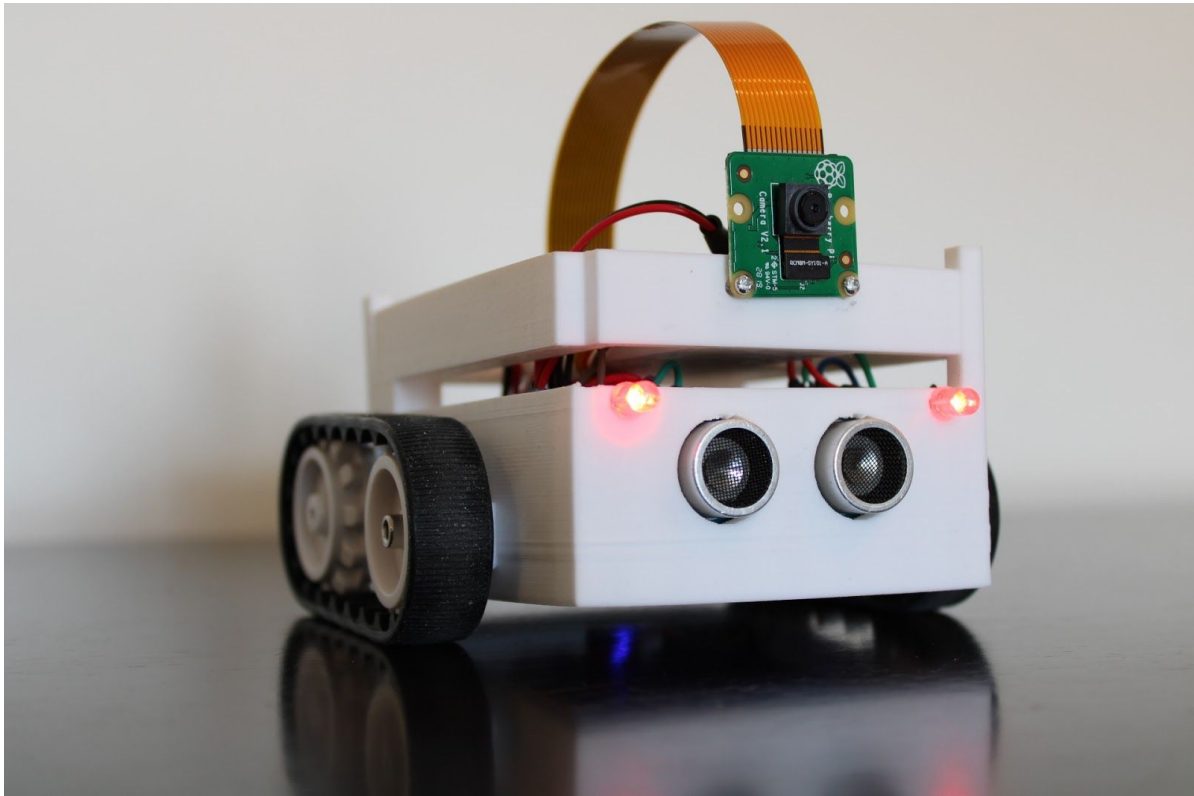


Radiostyrt fordon med mikroprocessor

En projektrapport i digitala system

EITA15



Handlerare: Bertil Lindvall

Grupp 1

David Vilppu

Victor Krook

Filip Sjövall

Oscar Blixt

Abstract

This paper describes the process of building a radio controlled vehicle. Some functions of the vehicle, in addition to being able to drive, was to be able to live-stream video to a computer. Other properties added to the vehicle was the ability to measure the distance in front of the vehicle, honk and have LEDs indicate if the vehicle is connected.

The primary goal was to be able to steer the vehicle wirelessly through a computer. This resulted in a graphical user interface, developed in Python, which communicates with the vehicle. The protocol used for communication is Bluetooth.

To create the vehicle in which all the components would fit, a chassis was required. The chassis was modeled in a 3D computer graphic software called Blender. When the model was finished, it was then printed out via a 3D-printer.

The project resulted in a vehicle that meets the goals that was initially set. The vehicle can be controlled and steered from a computer via Bluetooth and show a live-stream video at the same time. The video-livestream is managed with a Raspberry Pi and the data is sent over Wifi. The honk and LEDs are working as intended, showing a blue light when connected, and red when it's not.

Except for the car, a simple website was also developed. The purpose of the website is to show pictures of the vehicle and describe the different parts of it, including hardware and software. The website was created in html, css and javascript.

Sammanfattning

Denna rapport behandlar arbetsprocessen i ett projekt med målet att skapa ett radiostyrt fordon. Målet med fordonet var, utöver att kunna köra, även att ha en kamera som kan strömma video till dator. Tillval till fordonet längs vägen blev även en avståndsmätare som opererar via ultraljud, en tuta samt led-lampor indikerar när fordonet är uppkopplad eller inte.

Visionen med fordonet var att den skulle kunna styras från en dator. Detta resulterade i ett grafiskt användargränssnitt, utvecklat i Python, som kunde kommunicera med fordonet. Kommunikationen mellan dator och fordon skulle ske via bluetooth. Bluetooth skulle användas för att styra fordonet samt kontrollera tutan och avståndsmätaren

För att skapa fordonet behövdes ett chassi som passade de komponenter som valdes. Chassit för fordonet modellerades i 3D-modelleringsprogrammet Blender. När modellen sedan var klar kunde den skrivas ut den via en 3D-skrivare.

Projektet resulterade i ett fordon som satisfierar de mål som var uppsatta. fordonet kan styras via dator genom Bluetooth samt strömma bild via Raspberry Pi över Wifi. fordonet kan tuta och led-lamporna fungerar enligt de villkor som sattes upp.

Det bestämdes även att det skulle finnas en hemsida till projektet, där fordonet visades upp med hjälp av bilder och texter. Denna hemsida konstruerades med hjälp av en grundlig skiss, och skrevs därefter i html, css och javascript.

Innehållsförteckning

Abstract	1
Sammanfattning	2
Innehållsförteckning	3
1. Inledning	5
1.1 Bakgrund	5
1.2 Syfte	5
1.3 Målformulering	5
1.4 Problemformulering	6
1.5 Motivering av projektarbetet	6
2. Teknisk bakgrund	7
2.1 Hårdvara/komponenter	8
Mikroprocessor, ATmega1284	8
H-brygga, DRV883	8
Raspberry Pi W Zero	8
Raspberry Pi Camera V2	8
Avståndssensor, HC-SR04	9
Tuta, Piezobuzzer	9
Bluetooth-modul, HM-10 Bluetooth BLE	9
Step-up Boost Converter	9
Batteri	9
Laddnings-modul, TP4506	10
Leds, RGB-Led common cathode	10
Resistorer	10
Kristalloscillator, 20 MHz	10
DC motor	10
Larvband	10
2.2 Mjukvara	11
Programvara till bandvagnen	11
Programvara till PC	11
Programvara till Raspberry Pi	11
Android-applikation	11
Blender	11

3. Metod	12
Hemsida	12
Firmware	12
Mjukvara på dator	13
Modellering av fordonet	13
Utveckling av mobilapplikation	13
Komponenter	14
3.1 Källkritik	14
4. Analys	15
5. Resultat	16
Firmware	16
Mjukvara på dator	17
Utveckling av mobilapplikation	
Ingen fungerande mobilapplikation skapades.	18
Modellering av fordonet	19
6. Slutsats	20
7. Källförteckning	21

1. Inledning

1.1 Bakgrund

Denna rapport sammanfattar det projekt som utgör en fjärdedel av kursen Digitala System, EITA15. Syftet med projektet var att använda de kunskaper som lärts ut tidigare under kursen för att skapa en valfri prototyp som är baserad på mikroprocessorn ATmega1284. Valet av prototyp gjordes i början av projektet och skedde i samråd med handledare. Utifrån deltagarnas förkunskaper i gruppen beslöts det att prototypen som rapporten kommer handla om är ett radiostyrt fordon som kan styras från en dator via Bluetooth samt kunna strömma video från en kameramodul till datorn.

1.2 Syfte

Syftet med detta projekt var att utöka den generella förståelsen för digitala kretsar och samspelet mellan mikroprocessorn ATmega1284 och resterande komponenter. Detta innefattar även fördjupning av kunskaperna inom programmering med C. Syftet var också att ge en träning i att arbeta i grupp med ett större projekt.

1.3 Målformulering

- Skapa ett fordon som kan styras trådlöst från en dator och samtidigt strömma video i realtid.
- Skapa en applikation till Android som kan styra fordonet
- Skapa en hemsida där projektet visas och förklaras

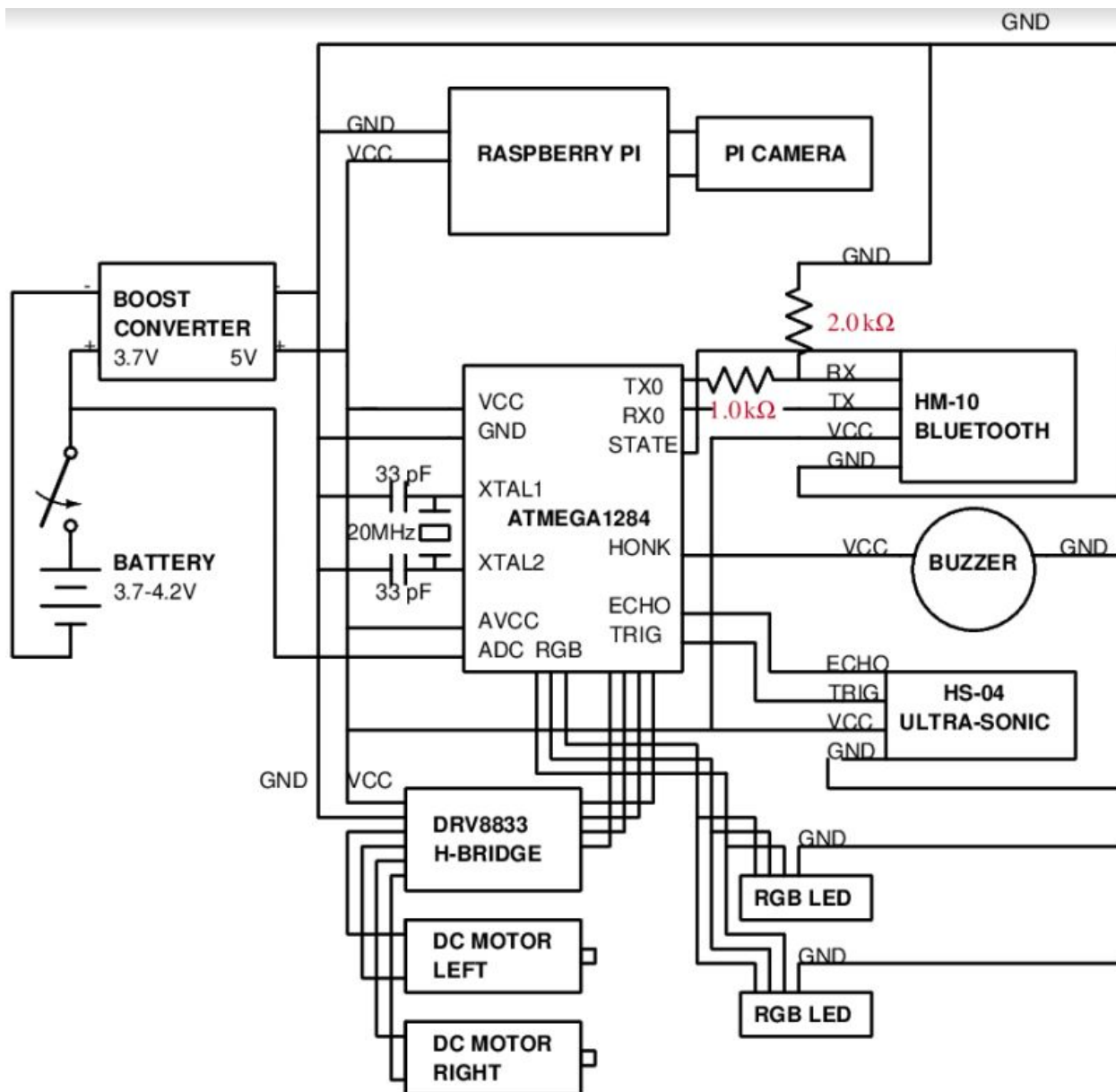
1.4 Problemformulering

- Hur ska kommunikation mellan mikroprocessorn och resterande komponenter ske?
- Hur ska komponenterna kopplas ihop?
- Hur ska fordonet styras från dator?
- Hur ska Android-applikationen utvecklas?
- Kan videoströmning ske via mikroprocessorn eller måste det ske via annan hårdvara?

1.5 Motivering av projektarbetet

Anledningen till att ett radiostyrt fordon valdes var att det upplevdes kunna ge bra förståelse om hur en mikroprocessor kommunicerar med komponenter, hanterar IO:s, avbrott och undersöka om det går att strömma video i realtid.

2. Teknisk bakgrund



Figur 1 - Kopplingsschema med samtliga komponenter till fordonet

2.1 Hårdvara/komponenter

Mikroprocessor, ATmega1284

ATmega1284 är en 8-bitars mikroprocessor med en RISC-arkitektur samt en Harvard-arkitektur. Den har 128 kB programmerbart flash-minne och 16 kB RAM. Klockfrekvensen kan bestämmas genom speciella fuse-bits¹ och kan då nå upp till 20 MHz med hjälp av en extern oscillator.

H-brygga, DRV883

Med hjälp av h-bryggan kan mikroprocessorn styra motorerna. Genom att kombinera insignalerna till h-bryggan kan strömflödets riktning styras. På så vis kan motorerna styras framåt respektive bakåt.

Raspberry Pi W Zero

Raspberry Pi W Zero är en mikroprocessor med en klockfrekvens på 1 GHz, 512MB RAM, Wifi, IO pins och mycket annat. Den kan köra hela operativsystem via microsd-kort, vilket betyder att det exempelvis går att köra Linux på den.

Raspberry Pi Camera V2

Raspberry Pi Kamera-modul version 2 har en 8-megapixels sensor som med hjälp av Raspberry Pi:s skräddarsydda grafikkort Videocore IV, kan filma i 30 fps med HD-upplösning.

¹ Speciella registerbitar som kan sättas i mikroprocessorn för att kontrollera generella beteenden, exempelvis vilken klockfrekvens den ska använda

Avståndssensor, HC-SR04

Modulen är en ultraljudssensor som vid kommando skickar ut ljudpulser. Därefter skickar den en signal till MCU:n då ljudvågorna har studsats tillbaka och genom att beräkna tiden det tar mellan att pulserna skickas och att pulserna har studsats tillbaka, kan avståndet från sensorn beräknas.

Tuta, Piezobuzzer

Har två ben, plus och minus. Ger ifrån sig ett pipande ljud då pluspolen spänningssätts.

Bluetooth-modul, HM-10 Bluetooth BLE

En BLE-modul som kan kommunicera via USART. Inställningar såsom baudrate och MAC-address går att programmera med hjälp av AT-kommandon². Stödjer baudrate från 4800-115200 bitar/sek.

Step-up Boost Converter

En spänningsomvandlare omvandlar spänningen från insignalen till en högre spänning. Då majoriteten av komponenterna kräver 5 V, men batteriet har en spänning på 3.7-4.2 V, behöver spänningen höjas.

Batteri

Batteriet som används är ett laddningsbart litiumjon-batteri, vars spänning är mellan 3.7-4.2 V.

² Speciella kommandon för att kommunicera med moduler

Laddnings-modul, TP4506

Laddningsmodulen möjliggör laddning av batteriet, genom USB. Modulen har inbyggd överladdningsskydd och bryter strömmen då batteriet är fulladdat.

Leds, RGB-Led common cathode

RGB-led är en LED med fyra ben. Ett ben är jord och det resterande ger vid spänning en färg på röd, blå eller grön.

Resistorer

Vissa komponenter, såsom LEDS och RX på Bluetooth-modulen behöver en lägre spänning än övriga komponenter. Då används resistorer för att sänka spänningen, exempelvis genom spänningsdelning.

Kristalloscillator, 20 MHz

Med hjälp av en extern kristalloscillator kan mikroprocessorn uppnå en klockfrekvens på 20 MHz (jämfört med dessa maximala 8 MHz utan). Detta möjliggör en snabbare processor, vilket betyder bättre responstid.

DC motor

Till motorer användes två stycken 6V 100:1 Micro Metal Gearmotor.

Larvband

Larvbanden är ett par 22-tandade silikonband men en diameter på 35mm.

2.2 Mjukvara

Programvara till bandvagnen

C-koden har skrivits i utvecklingsmiljön Eclipse. För att bränna in koden i mikroprocessorn har en Arduino används, som med hjälp av SPI³ kan programmera ATMegar. Då JTAG inte används har det ej gått att använda någon debug-funktion.

Programvara till PC

Koden till PC skrivs i Python och har skrivits i utvecklingsmiljön VS Code.

Programvara till Raspberry Pi

Koden har skrivits i Python och Bash i utvecklingsmiljön VS Code och förts över till Raspberry Pi via ssh och git.

Android-applikation

Applikationen utvecklades i utvecklingsverktygen Android Studio, med hjälp av Java.

Blender

Chassit utvecklades i programmet Blender. Blender är ett modelleringsverktyg som kan exportera filer till .stl vilket är den filtyp som används av 3D-printers för utskrift.

³ Serial Peripheral Interface, ett kommunikationsprotokoll som är populärt inom mikrocontrollers

3. Metod

Projektarbetet började med planering. I denna planeringsfas togs beslut om vilka komponenter som skulle användas, hur chassit skulle skapas samt hur logiken i mjukvaruutvecklingen skulle gå till. När detta var beslutat skapades ett kopplingsschema som ses i figur 1. Därefter fortsatte projektet med individuellt arbete med varje gruppmedlems tilldelade arbetsområde.

Under projektet har varje gruppmedlem arbetat på egen hand för att sedan stämma av under veckovisa möten. Under mötena har eventuella frågor, problem och beslut som ska fattas diskuterats.

Hemsida

Det fanns en grundtanke om hur hemsidan skulle se ut, där det skulle vara någon typ av kubanimation på startsidan som visar olika bilder på fordonet. Denna slider konstruerades med hjälp av en JQuery-plugin⁴ som heter Swiper. Designen för resterande delar av hemsidan besutades under möterna och arbetades fram succesivt.

Firmware

Mjukvaran till fordonet utvecklades successivt allteftersom fler beslut togs och komponenter införskaffades. Varje del av koden skrevs för sig, innan huvudprogrammet kunde skrivas ihop. Detta för att kunna felsöka varje del av kod individuellt. Efter experiment med diverse olika kameramoduler fick hoppet om att skicka video i realtid från en 8-bitars mikroprocessor med en klockfrekvens på 20 MHz ges upp. För att kunna uppfylla projektets mål fick då en ny plan skapas där en Raspberry Pi istället skulle stå för videoöverföringen med hjälp av en Raspberry Pi Camera Module V2. Då

⁴ JQuery är Javascript-bibliotek med massor av färdigskrivna metoder

Bluetooth LE inte kan uppnå tillräckligt höga överföringshastigheter ansågs Wifi vara det mest lämpliga protokollet att använda för videoöverföring. Raspberry Pi:en kan programmeras genom att skapa en ssh-tunnel⁵ för att sedan skriva kod direkt på mikroprocessorn eller skicka över koden från datorn.

Mjukvara på dator

Koden utvecklades allt eftersom. En GUI⁶ utvecklades samt två program som kunde kommunicera med ATmega1284 genom Bluetooth, samt ta emot videoöverföringen från Raspberry Pi:en. Dessa funktionaliteter togs fram med hjälp av färdigskrivna bibliotek.

Modellering av fordonet

Modellering av chassit utfördes i programmet Blender. Eftersom ingen ritning hade tagits fram för utseendet var det fria händer och ett agilt arbetssätt användes för modelleringen. Inspiration för designen togs av ett liknande chassi som kom tillsammans med motorerna. Chassit var tvunget att kunna innesluta komponenterna, bestå av hål för fyra hjul, en avståndssensor, led-lampor samt hål för montering av kamera. Många utskrifter gjordes för att nå den slutgiltiga designen som bestod av två våningar.

Utveckling av mobilapplikation

För utveckling av mobilapplikationen användes Googles utvecklingsmiljö Android Studio och programmeringsspråket Java. Eftersom kunskap om applikationsutveckling saknades krävdes en fas av inläring om utvecklingsmiljön, hur BLE fungerar samt applikationsutveckling som helhet. Därefter påbörjades skrivandet av koden.

⁵ En krypterad anslutning mellan två klienter

⁶ Graphical User Interface, ett grafisk användargränssnitt

Komponenter

Komponenterna sattes ihop med hjälp av kretsschemat som skapades under planeringsfasen. Det mesta av komponenterna är fastlödda på ett kretskort som sitter fast i chassit med hjälp skruvar. Sladdar och krympslang har använts för att göra kopplingarna kortslutningssäkra och hållbara. På kretskortet finns pins som sticker upp, för att enkelt kunna programmera om bandvagnen vid behov. På undersidan av bandvagnen sitter batteriet, spänningsomvandlaren och strömbrytaren som alla är fastlimmade.

3.1 Källkritik

Datablad från komponenttillverkare användes för detaljer och information. För mjukvaruutvecklingen användes diverse olika källor. Dessa källor var allt ifrån Youtube, Stack Overflow och Github till andra programmeringsforum. Trots att det kan vara svårt att säkerställa legitimiteten av dessa källor, spelade det i detta sammanhanget inte så stor roll, då koden testades och om det då funkade, var källorna användbara och får således anses som pålitliga.

4. Analys

Det största problemet var en direkt effekt av den rådande pandemin vilket ledde till att varje gruppmedlem arbetade individuellt snarare än gemensamt, detta försvårade projektet avsevärt eftersom arbete skedde i separata arbetsmiljöer. Under programmeringen saknades även tillgång till en debugger vilket försvårade arbetet med mikroprocessorn.

Vid konstruktionen av basen till fordonet uppstod en komplikation. Måtten vid utskrivning skiljde sig från de mått som användes i Blender. Detta ledde till omgångar av utskrivningar med baser som inte passade övriga komponenter. För att lösa problemet skapades nästa version med större mått för att kompensera för felet.

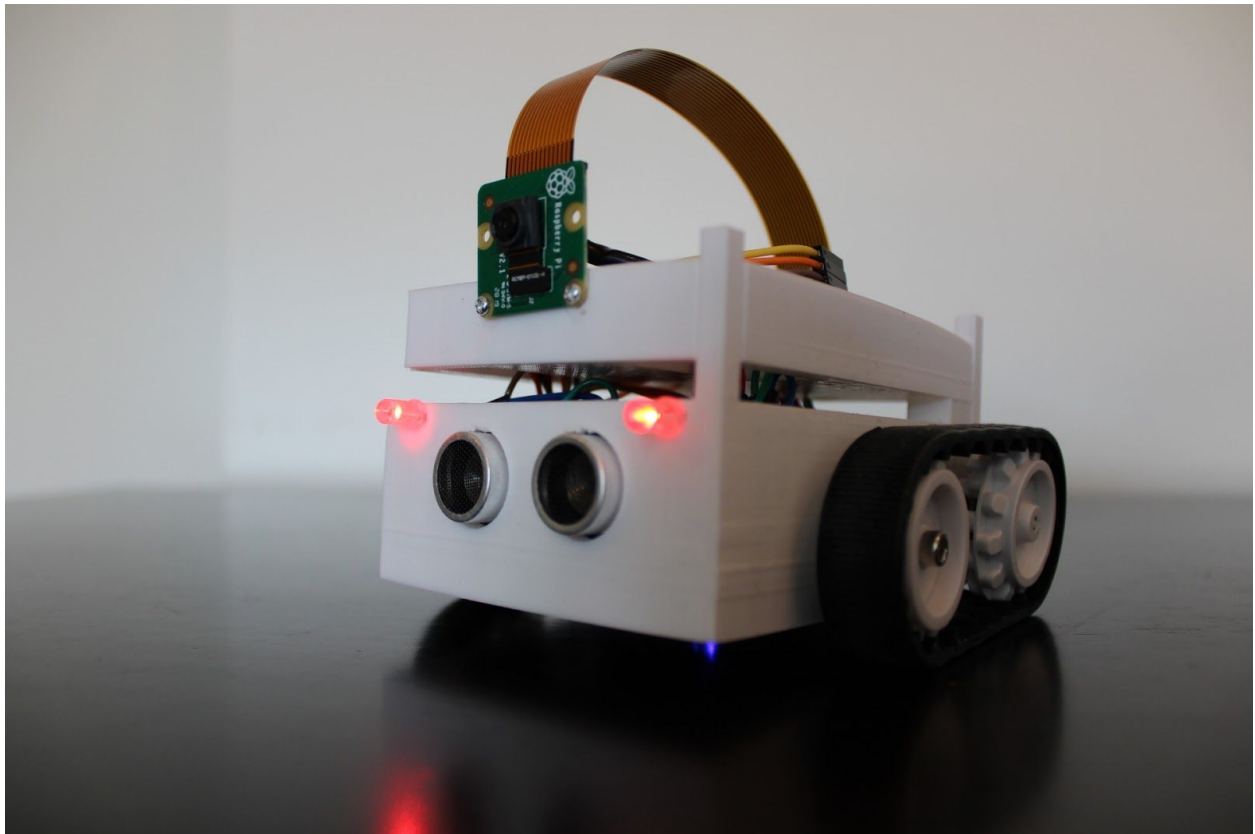
Under utveckling av mobilapplikationen bestod en stor del av arbetet av inläring om hur mobilapplikationer är kodade, hur BLE fungerar. Det visade sig vara mer avancerad än förväntat och mycket tid gick åt till inlärningsfasen.

Vid körning av fordonet sker ibland plötsliga stopp där fordonet inte längre går att köra. Det kan bero på att batteriet som används inte längre har sin fulla kapacitet. Anledningen är dock okänd och för att undersöka om teorin är korrekt bör ett annat batteri testas.

Anledningen till att kameran körs via Raspberry Pi var att Raspberry Pi:en sköter bild-komprimeringen oerhört effektivt och därmed kunde önskad videokvalité uppnås. Denna metod valdes på grund av att ATMegans inte kunde ge ett önskat resultat när det kom till upplösning eller bilder per sekund. Vid experiment med tidigare kameramodul som kommunicerade via USART, uppnåddes en videoöverföring med 1,5 fps, vilket är väldigt lågt.

5. Resultat

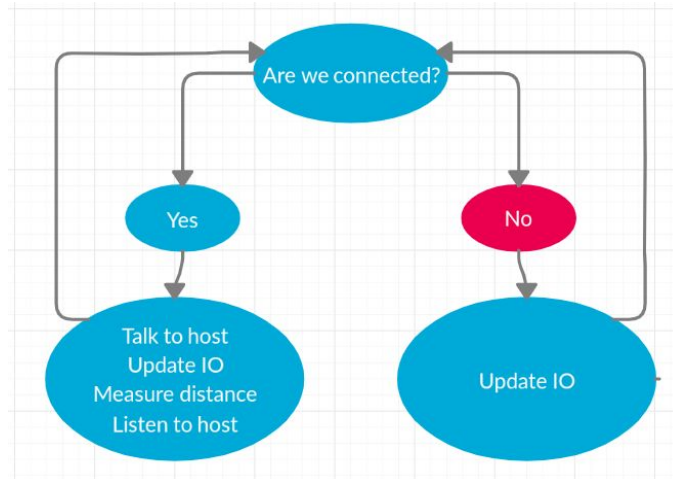
Projektet resulterade i ett fungerande fordon som kan köra framåt, bakåt samt svänga. Fordonet kan även tuta, mäta avstånd samt visa med led lampor om den är ansluten. Styrning av samtliga komponenter nämnda ovanför sker via bluetooth från en dator. Kameramodulen skickar en live-feed via Raspberry Pi över Wifi till samma program.



Figur 2 - Bild på färdigställda prototypen

Firmware

Logiken för bandvagnen är att kolla om den är ansluten till bluetooth och därefter ta beslut om vad den ska göra, kommunicera med PC och styra IO-enheter. Motorena styrs med hjälp av pwm.



Figur 3 - Synkron kommunikation

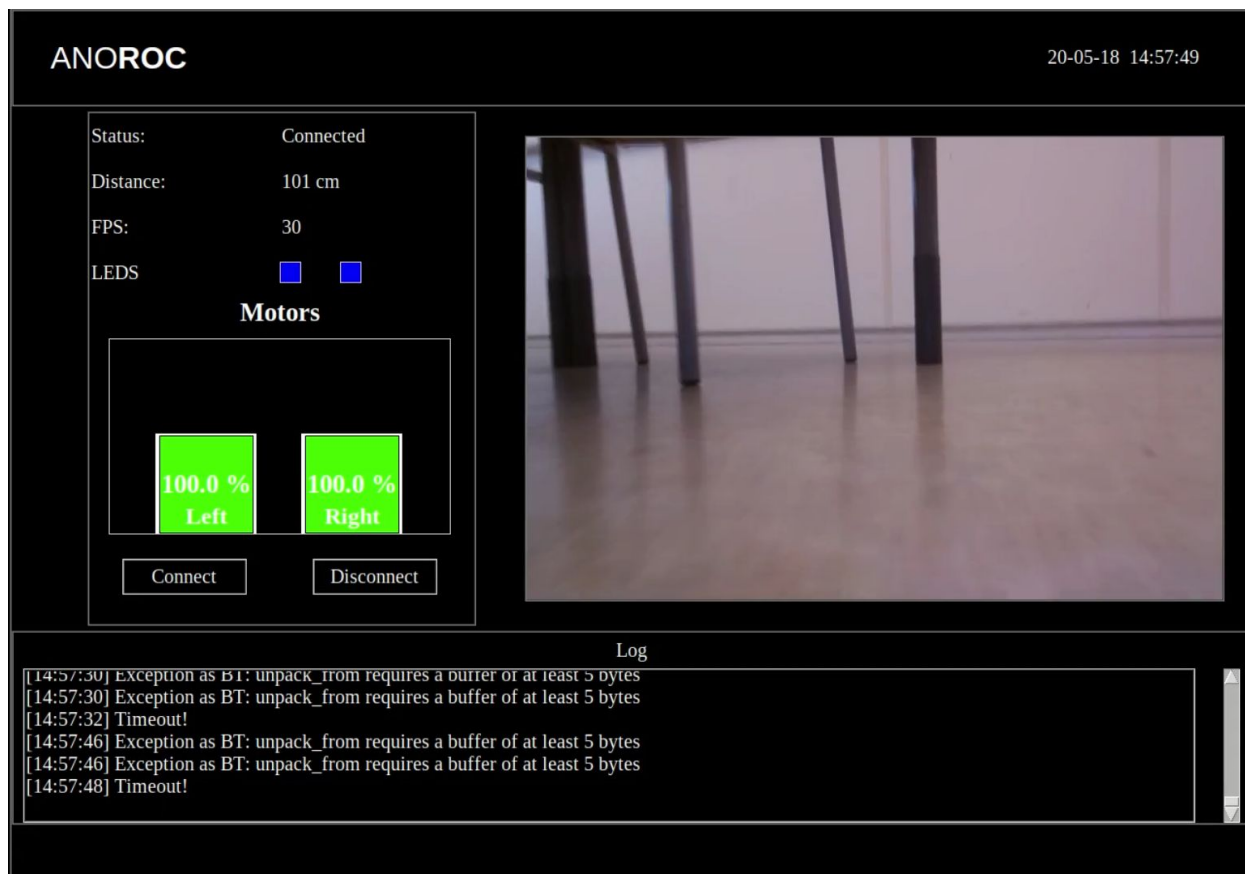
Som figur 3 visar, sker all kommunikation och styrning synkront, men eftersom mikroprocessorn går så pass snabbt upplevs det som asynkront. ATMegans får kommandon från Bluetooth-modulen via USART. För att förhindra att mikroprocessorn fastnar någonstans används en timer, som vid overflow ökar en global variabel.

Raspberry Pi:en kör en linuxdistribution som heter Raspbian. Då processorn startar letar den efter ett nätverk, försöker ansluta till detta nätverk, om detta lyckas, startas en TCP-förbindelse till PC, och därefter startas kameran och Pi:en börjar skicka videoöverföring via TCP-förbindelsen.

Mjukvara på dator

På PC-sidan startas en Wifi Hotspot, samt en TCP-server som lyssnar efter anslutningar till det nya nätverket, för att ta emot videoöverföring. Samtidigt så skannar PC:n efter Bluetooth-enheter och ansluter till bandvagnens Bluetooth-modul om den hittas. Om anslutningen via Bluetooth lyckas, börjar data skickas fram och tillbaka mellan bandvagnen och PC.

Genom en interaktiv GUI kan bandvagnen styras från PC där live-feeden visas.



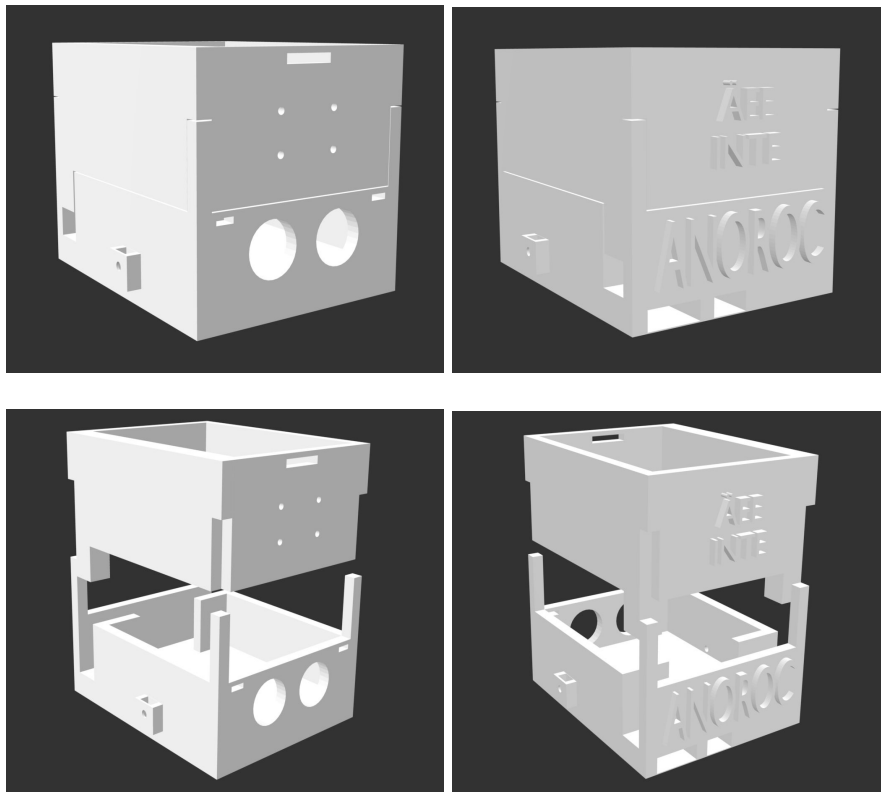
Figur 4 - Bilder på den interaktiva GUI:n

Utveckling av mobilapplikation

Ingen fungerande mobilapplikation skapades.

Modellering av fordonet

Modellen som skapades i Blender och skrevs ut via en 3D-skrivare fick den slutgiltiga designen som ni hittar nedanför i figur 5.



Figur 5 - Bilder på slutgiltiga modellen till chassit

6. Slutsats

Under detta projekt har ett fordon skapats som kan styras genom Bluetooth, från en dator. Fordonet kan även skicka en videoöverföring i realtid, lysa med leds, tuta och mäta avståndet till objekt framför det. En hemsida har även utvecklats där projektet visas upp men en Android-applikation kunde ej skapas.

Projektet kunde avslutas i tid och uppfyller majoriteten av målen som sattes upp och anses därför som ett lyckat projekt.

7. Källförteckning

Harvey, Ian. ianharvey.github.io. 2020. bluepy - a Bluetooth LE interface for Python (uppdaterad 2020) <https://ianharvey.github.io/bluepy-doc/> (Hämtad 2020-03-15)

Wiki.archlinux, 2020. NetworkManager - ArchWiki (uppdaterad 2020) <https://wiki.archlinux.org/index.php/NetworkManager> (Hämtad 2020-03-23)

Jones, Dave. 2016. picamera (uppdaterad 2020) <https://picamera.readthedocs.io/en/release-1.13/> (Hämtad 2020-03-27)

avr-libc. 1999. Standard C library for AVR-GCC (uppdaterad 2016) <https://nongnu.org/avr-libc/user-manual/> (Hämtad 2020-03-19)

JS js.foundation. JQuery API Documentation (uppdaterad 2020) <https://api.jquery.com/> (hämtad 2020-04-06)

Google, Bluetooth Low Energy (uppdaterad 2020) <https://developer.android.com/guide/topics/connectivity/bluetooth-le> (Hämtad 2020-03-20)