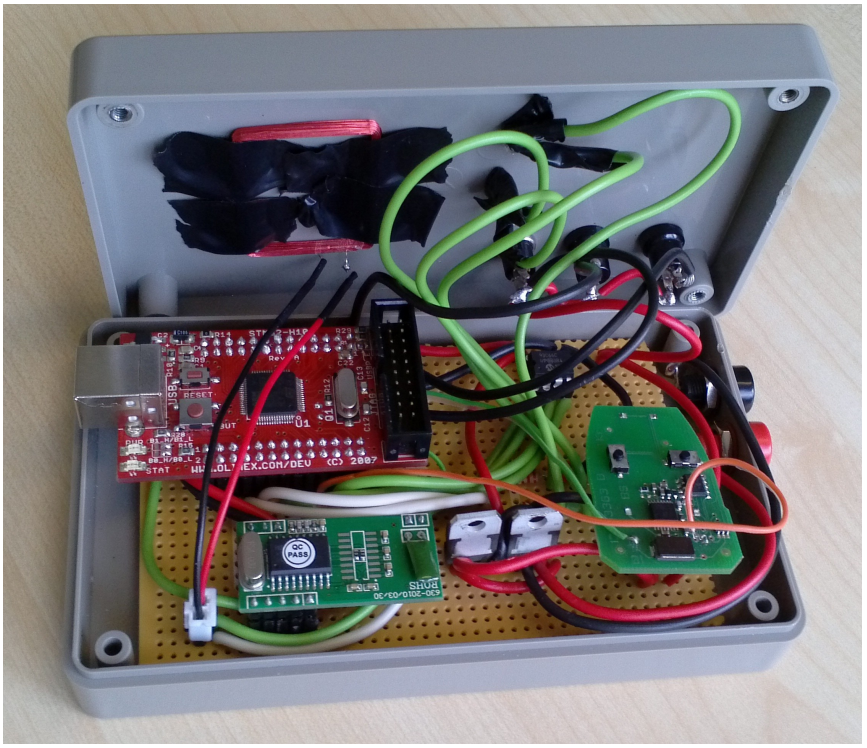
CHALMERS



Automatisk Låsstyrning med RFID

Passive Keyless Entry with RFID

Examensarbete för högskoleingenjörsexamen inom Elektroingenjörsprogrammet och Dataingenjörsprogrammet

Daniel Josefsson

Dina Zuko

Institution för signaler och system

CHALMERS UNIVERSITY OF TECHNOLOGY

Gothenburg, Sweden, 2013

Examinator: Bertil Thomas

Automatisk Låsstyrning med RFID

Passive Keyless Entry with RFID

Daniel Josefsson
Dina Zuko

Institution för signaler och system
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden, 2013
Examinator: Bertil Thomas

Förord

Examensarbetet för automatisk styrning av låssystem är genomfört på konsultbolaget Broccoli Engineering AB. Broccoli är ett företag verksamt inom inbyggda system. Arbetet är utfört tio veckor under våren 2013 motsvarande 15 högskolepoäng. Vi som har utfört arbetet studerar till högskoleingenjörer inom Datateknik respektive Elektroteknik. Handledare från Chalmers har varit Göran Hult och examinator Bertil Thomas.

Vi vill rikta speciellt tacka till följande personer: Björn Bergholm för att ha givit oss möjligheten till detta exjobb och all hjälp längs vägen, Thomas Hellström, Henning Colliander och Tobias Ekman på kontoret genom att alltid ta sig tid att hjälpa, tjejerna på kontoret som har fått en att känna sig välkommen, företagets handledare Henrik Brenander som hjälpte till att få nytt perspektiv på saker, skolans handledare Göran Hult som kommit med tips och idéer samt examinator Bertil Thomas för examination.

Daniel Josefsson och Dina Zuko

Sammanfattning

Rapporten beskriver arbetet med att utveckla ett passivt automatiskt låsstyrningssytem för fordon med hjälp av RFID-teknik. Det huvudsakliga målet med projektet var att utveckla en prototyp där dörrarna till ett fordon ska låsas upp när rätt nyckelkort, så kallad tagg, befinner sig cirka 2-3 meter från fordonet och låsas när den inte längre gör det. Att integrera systemet med fordonets elsystem är inte en del av projektet. Som en del av projektet ska även utvecklingsmiljö för utveckling av mjukvara väljas, systemet ska kravställas och lämplig RFID väljas med avseende på dessa krav. Lämplig microcontroller ska väljas, schema för ett mönsterkort ritas och mjukvara ska testas. Hela projektet har utförts på Broccoli kontor i Göteborg. Broccoli är ett konsultföretag med fokus på hårdvaru- och mjukvaruutveckling. De erbjuder tjänster inom design, konstruktion och testning av inbyggda system med inriktning mot olika branscher såsom fordon, automation, marin, medicin med flera.

Vad gäller valet utvecklingsmiljö så har en Open Source variant där Eclipse kombineras med OpenOCD använts. Ett schema för mönsterkort har tagits fram och med detta kan ett kretskort beställas. Microcontrollern som valdes var en ARM processor, en microcontroller som används inom industrin. Att hitta en billig RFID-modul som kan kommunicera med taggen på cirka 2-3 meters avstånd visade sig vara problematiskt. Därför byggdes inledningsvis en prototyp som fungerar på kort avstånd. Nästa steg var att få bygga en prototyp som fungerar på långt avstånd. Då det saknas billiga RFID-moduler på efterstävat avstånd fattades beslutet att bygga en egen RFID-modul bestående av RF-modul och microcontroller. Arbetet med RF-modulen misslyckades. Orsaken var att RF-modulen som köptes in endast kan kommunicera via ett speciellt applikationsprotokoll. Taggar saknar intelligens och har därmed inte förmåga att kommunicera via ett applikationsprotokoll.

Arbetet har resulterat i en fungerande funktionsprototyp bestående av microcontroller, RFID-modul och externt minne. Då arbetet med att få igång kommunikationen på långt avstånd inte har implementeras på grund av tidsbrist har en prototyp som fungerar på kort avstånd byggts. Den demonstrerar funktionen och visar att det är möjligt att konstruera ett sådant system. Fortsatt arbete bör fokusera på att antigen söka vidare efter en befintlig RFID-modul, satsa på ett system med en aktiv tagg och RF-transceiver eller att utveckla en egen RFID-modul från grunden.

Abstract

The purpose of this project was to develop a passive keyless entry system for vehicles using RFID-technology. The main goal was to develop a prototype that will control the door in a vehicle, the doors will unlock when the right tag is detected within 2-3 meters from the vehicle and lock when the tag is no longer detected. To integrate this system with the system of the vehicle was not a part of this project. Also, as a part of the project a software development environment is to be chosen, system requirements are to be specified and a suitable RFID based on these requirements is to be chosen. A suitable microcontroller is to be chosen, a PCB-layout drawn and the software is to be tested.

The whole project has been done at Broccoli's office in Gothenburg, Sweden. Broccoli is a consulting-firm that works primarily with hardware and software development. They offer services in designing, constructing and testing embedded systems within primary automotive and marine industries.

Looking at the results, concerning the development environment an Open Source framework was chosen where Eclipse is combined with OpenOCD. A PCB-layout was drawn and from this layout a PCB can be manufactured. The microcontroller that was chosen was an ARM processor, STM32F103, a microcontroller frequently used in the industry. To find an inexpensive RFID-module that can communicate with tags at a distance at minimum 2 meters proved to be the most difficult task. Due to this, a prototype that works over short distances was initially developed. The next step was to develop a prototype that operates at far distance. Because of the lack of suitable RFID-modules a decision was made to construct a RFID-module using a RF-module and a microcontroller. The attempt to do that proved unsuccessful. The reason being that the purchased RF-module only can communicate through an application protocol called EnOcean. The tags lack intelligence and therefore the ability to communicate through an application protocol.

The result from this project is a working functionality-prototype consisting of a microcontroller, RFID-module and external memory. Due the fact that the attempt to develop a far-distance prototype failed, a prototype that functions at short distances was build. It demonstrates the functionality and proves it possible to build such a system. The future work should focus on either continue search for a suitable RFID-module, use a system with active tags and RF-transceivers or construct a RFID-module from scratch.

Innehållsförteckning

Beteckningar

ARM - Acorn RISC Machine, processorer baserade på ARM arkitekturen utvecklade av ARM Holdings

ASCII - American Standard Code for Information Interchange, en teckenkodning som används för att representera bokstäver och andra tecken i datorer

ASK - Amplitude Shift Keying, digital amplitudmodulering

CAN - Controller Area Network, en busstandard främst använd inom fordonsindustrin

CS - Chip Select

CrossWorks - Kommersiell utvecklingsmiljö som används som att kompilera och debugga C/C++-kod på microprocessorer, till exempel ARM

DB-9 - en elektrisk kontakt som används särskilt ofta inom datorsammanhang, har 9 pinnar

DB-25 - en elektrisk kontakt som används särskilt ofta inom datorsammanhang, har 25 pinnar

DCE - Data Communication Equipment, har som funktion att omvandla signaler och klocka dem. Till exempel ett modem

DTE - Data Terminal Equipment, tar emot signaler från DCE och gör om dem till information eller tvärtom. Till exempel en dator

Eclipse - en utvecklingsmiljö för att exempelvis utveckla programvara i programmeringsspråken Java och C/C++

EEPROM - Electrically Erasable Programmable Read-Only Memory, ett icke-flyktigt minne där innehållet bibehålls även när enheten inte är spänningssatt

EIA - Electronic Industies Alliance, en organisation i USA som utvecklade en standard för att produkter av olika tillverkare ska kunna kommunicera

EPC - Electronic Product Code, identifikationssystem som ger ett unikt ID till varje fysisk produkt globalt, används inom RFID-tekniken

FSK - Frequency-Shift Keying, digital frekvensmodulering

Full duplex - kommunikation i två riktningar där båda riktningarna kan verka samtidigt

GPIO - General Purpose Input/Output, en programerbar pinne på en microcontroller, kan användas för input eller output

Halv duplex - kommunikation i två riktningar där endast en riktning kan användas samtidigt

I2C - Inter-Intergrated-Circuit, synkron seriell multimasterbuss från Philips, används för att koppa ihop låghastighetsenheter

IC - Integrated Circuit, en elektronisk krets där komponenterna tillverkas tillsammans på samma chip

IEC - International Electronic Commision, kommission som tar fram och faställer internationella standarder inom elektronik

ISO - International Organization for standardization, tar fram standarder i samband med andra organization som till exempel IEC

JTAG - Joint Test Action Group, kontakt som används för att följa och felsöka processen under exekvering av microprocessorer

KiCad - Open Source-programpaket för design av kopplingsschema och mönsterkort (PCB)

MISO - Master In, Slave Out

MOSI - Master Out, Slave in

Open Source - avser datorprogram där källkoden är tillgänglig för att använda, läsa, modifiera och vidaredistribuera

OpenOCD - Open On-Chip Debugger, ett open source-program som används för att programmera, debugga och scanna inbyggda system

OSI-modellen - Open Systems Interconnection-modellen, delar in datakommunikationen i sju lager från det fysiska lagret till applikationslagret

PCB - Printed Circuit Board, mönsterkort med monterade elektriska komponenter

RDM630 - en RFID-läsare med frekvensen 125 Khz

RFID - Radio Frequency Identification, kontaktlös kommunikation och identifikation med hjälp av radiovågor

RS232 - Recommended Standard 232, asynkron kommunikationsstandard med seriell databuss

SCK - Serial clock

SPI - Serial Peripherial Interface, en synkron buss som används för seriekommunikation

UART - Universal Asynchronous Receiver/Transmitter, datorhårdvara som omvandlar parallell data till seriell data och överför sekventiellt en byte i taget

UHF - Ultra High Frequency, radiovågor med frekvensen mellan 300 MHz och 3GHz. Har våglängden mellan 1 m och 10 cm

USB - Universal Serial Bus, seriellt kommunikationsprotokoll som är standard i dagens datorer

TTL - Transistor - Transistor Logic, en standard som använder sig av 5-volts spänningskälla. Nolla/låg befinner sig mellan 0 och 0,8 V och etta/hög befinner sig mellan 2 och 5 V över jordterminalen

1. Inledning

1.1 Bakgrund

Det är vanligt att fordon som utvecklas idag har låssystem som kan fjärrstyras, så kallad keyless entry. Det kan handla om att dörren till fordonet ska låsas upp när en knapp trycks in inom ett visst avstånd från fordonet. Ett annat alternativ är att föraren inte ska behöva trycka på någon knapp överhuvudtaget, föraren ska vara passiv. Det finns många namn för detta men det vanligaste är Passive Keyless Entry. Ett sådant system kan fungera på två sätt. Antingen att fordonet låses upp när föraren med en tagg befinner sig på rätt avstånd. Eller att föraren även behöver ta i handtaget för att dörren skall låsas upp med taggen på sig. De ovan nämnda systemen är ganska vanliga i nyproducerade fordon. Dock är utbudet väldigt begränsat på eftermarknaden.

Projektet går ut på att utveckla ett passivt automatiskt låssystem. Arbetet kommer att utföras på Broccoli Engineering AB, ett konsultföretag med fokus på hårdvaru- och mjukvaruutveckling. De erbjuder tjänster inom design, konstruktion och testning av inbyggda system med inriktning mot olika branscher såsom fordon, automation, marin, medicin med flera.

Broccolis kund är i behov av ett system där föraren inte ska behöva interagera med fordonet. Det skall alltså automatiskt låsas när man går ifrån och låsas upp när man åter närmar sig. De har tidigare försökt köpa ett färdigt system på marknaden men systemen har inte fungerat tillfredsställande. De har även varit i kontakt med en fordonstillverkare för att få hjälp med att utveckla ett system men på grund av att det endast skulle gå att implementera lösningen i nyproducerade fordon var det inte aktuellt.

1.2 Syfte

Syftet med projektet är att utveckla ett passivt automatiskt låsstyrningssystem för fordon med hjälp av RFID. Det skall tas fram en prototyp som visar funktionen i systemet. Dörrarna ska låsas när chauffören befinner sig utanför en visst avstånd från bilen och de ska låsas upp när chauffören befinner sig innanför detta avstånd. För att slippa fladdrande funktion då chauffören befinner sig på gränsen mellan zonen för låsning och upplåsning behöver hysteres tas i beaktande. Systemet ska vara säkert i den bemärkelsen att endast rätt RFID-tagg kan låsa upp bilen.

Med den färdiga prototypen skall även dokumentation angående framtida utveckling av systemet presenteras. Detta skall innehålla slutsatser som visar lämplig fortsättning med avseende på till exempel pris och stabilitet för att nå en färdig produkt.

1.3 Avgränsningar

Endast en prototyp ska utvecklas där funktionen kan demonstreras. Att integrera systemet med fordonets elsystem är inte en del av projektet. Även helt egenutvecklad RFID står utanför tidsramen för det här projektet då RF-utveckling är för tidskrävande. Istället kommer färdiga moduler användas för konstruktion av systemet.

1.4 Precisering av arbetsuppgiften

Val av utvecklingsmiljö för utveckling av mjukvara.

Kravställande av hela systemet med avseende på kostnad, funktion, säkerhet och tillförlitlighet.

Välja lämplig RFID variant med avseende på avstånd, funktion säkerhet och tillförlitlighet.

Välja lämplig microcontroller.

Konstruera hårdvara (schema och PCB) och mjukvara.

Testa mjukvara.

Bygga en prototyp.

2. Metod och material

Detta kapitel har för avseende att ge en bild av arbetsmetoden, beskriva de olika faserna i projektet och hur arbetet har planerats och utförts. Kapitlet innehåller också en del där använt material finns angivet.

2.1 Metod

Detta projekt har genomförts enligt iterativ utvecklingsmetod. Beslutsfattandet kring val av metod har skett i samband med utvecklingen av prototypen.

I början av projektet togs en planeringsrapport fram för de kommande tio veckorna, den skulle vara vägledande under arbetets gång. För varje vecka sattes det upp mål för att ge en struktur i det kommande arbetet. Dessa mål kom senare att följas upp vecka för vecka. Planeringsrapporten togs fram i samråd med Henrik Brenander och Björn Bergholm på Broccoli. Projektet inleddes även med en förberedelsefas där en litteraturstudie genomfördes för att inhämta information om RFID-teknik, microcontrollern STM32F103-RBT6 samt de olika gränssnitten som skulle användas i projektet. Därefter följde en genomförandefas där utvecklingen av själva prototypen utfördes. Det var den mest omfattande delen i detta projekt. Genomförandefasen bestod av praktiskt arbete med själva prototypen samt diskussioner kring problem, tillvägagångssätt och möjliga lösningar på problemen. De beslut som fattades, resonemang och grunden för dessa beslut finns redovisade i delen Utveckling av Automatiskt Låsstyrningssystem. Projektet avslutades med en fas med fokus på diskussion, analys samt reflektion om framtida utveckling av produkten.

Under arbetets gång har dagsrapporter och veckorapporter skrivits. Förutom en sammanfattning av veckan som gått innehåller veckorapporterna även diskussion kring målen för veckan, om målen blev uppnådda och eventuella anledningar till varför de inte blev det. Handledarna Göran Hult och Henrik Brenander hade tillgång till dessa veckorapporter via Google Drive. Man har även kunnat följa arbetets gång via en blogg på Broccolis hemsida (broccoli.se).

Hela projektet har utförts på Broccolis kontor i Göteborg. Då företaget är ett konsultbolag är de flesta anställda ute på uppdrag hos olika företag. Dock så har två av konsulterna, Thomas Hellström och Henning Colliander, varit tillgängliga på kontoret och bidragit med tips och idéer. Även Broccolis VD Björn Bergholm har varit tillgänglig för konsultation. All teknisk utrustning, komponenter och instrument som har används i projektet har Broccoli bidragit med.

2.2 Material

Material som har används i projektet finns angivet nedan. Länken till respektive inköpsställe samt dåvarande pris hittar ni delen 6.2, Materialreferenser. Den enda produkten där det saknas uppgifter om inköpsställe samt pris är en nyckel/fjärrkontroll från Volvo PV. Denna har används i ett tidigare projekt på Broccoli.

EEPROM microchip 93C66C-I/P

Mottagarmodul RX433, 433,92 MHz

RFID-läsare RDM632, 125kHz

RF-transceiver 868MHz TCM 320

STM32-H103 experimentkort från Olimex

Tagg passiv 125kHz, Sparkfun COM-10169

Tagg passiv 868MHz EURUHFT4930

Volvo vo-am433,92MHz 31110091 DELPHI nyckel/fjärrkontroll

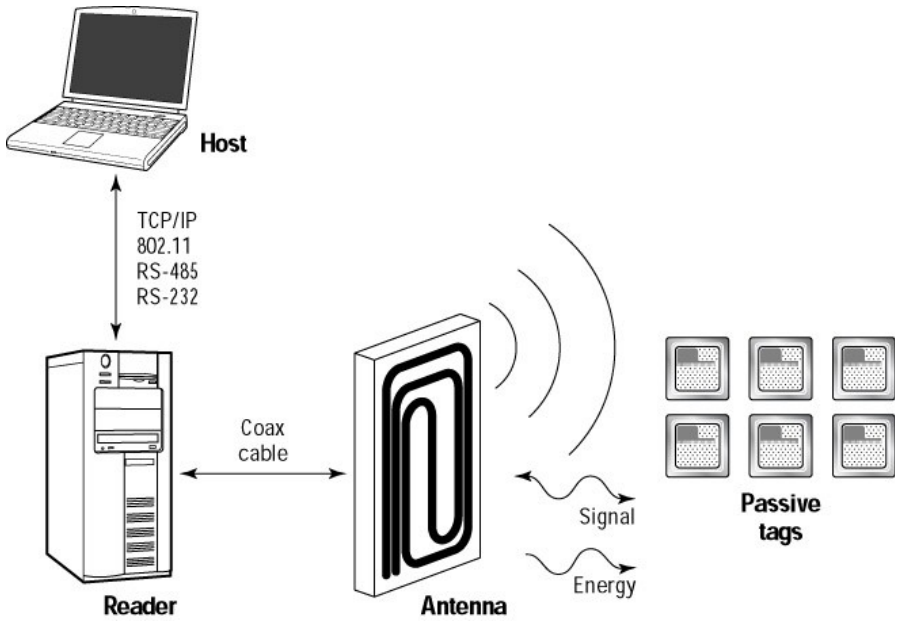
3. Teknisk bakgrund

I detta kapitel behandlas de olika tekniker som har använts i projektet. Bakgrund kring teknikerna samt funktion kommer diskuteras. Hur RFID fungerar kommer beskrivas samt genomgång av bakgrundsfakta till microcontrollern och kommunikationsprotokollet RS-232.

3.1 RFID

RFID står för Radio Frequency Identification och är ett kommunikationsverktyg där ingen fysisk eller synlig kontakt behövs. En transponder och en transceiver, även kallade tagg och läsare, kommunicerar trådlöst på en viss frekvens likt annan radiokommunikation. Läsare, tagg och antenn utgör grunden för RFID kommunikation. Läsaren skapar en signal som skickas via kabel ut till antennen. När den elektriska signalen har nått antennen skickas en analog signal trådlöst med en viss frekvens. Området som den elektromagnetiska signalen skickas till kallas för interregation zone (läszon på svenska). Inom detta område kan taggar upptäcka signalen (Glover & Bhatt, 2006).

Läsaren skickar inte bara ut signaler via antennen. Den lyssnar även på inkommande signaler från taggar. De analoga signaler som tas emot omvandlas till digitala signaler, ettor och nollor. Dessa skickas vidare till en enhet som skall hantera informationen, till exempel en dator eller en microcontroller (Glover & Bhatt, 2006).



Figur 3.1 RFID-system som skickar data (Sweeney, 2005)

Bilden ovan, figur 3.1, visar hur kommunikationen fungerar. Antennen skickar ut signaler/energi och taggen svarar när den känner av signalen. När antennen har tagit emot signalen från

taggen skickas informationen vidare till RFID-läsaren. Här görs signalen om till en digital motsvarighet som sedan skickas vidare till en värddator.

En RFID-tagg består av två delar: ett chip eller integrerad krets (IC-krets) och en antenn. På chippet finns ett nummer som är specifikt för just den taggen. Detta unika nummer kallas electronic product code (EPC). Många RFID-taggar använder EPC-protokollet vars två huvudsakliga uppgifter är att ange hur data ska delas upp och sparas på taggen samt bestämma vilket gränssnitt som ska användas, det vill säga hur läsare och tagg ska kommunicera med varandra. Chippet innehåller även instruktioner om vad den ska göra när den känner av en läsare. Via antennen får taggen energi som gör att den kan kommunicera och utbyta data med RFID-läsaren (Glover & Bhatt, 2006).

Taggar kan vara antingen passiva, aktiva eller semipassiva. Aktiva taggar har en egen strömkälla som används för att skicka information och driva komponenter. Den här typen av taggar är mer pålitliga än passiva taggar och har längre räckvidd, upp till flera hundra meter. Batteriet byts med några års mellanrum. Passiva taggar däremot har ingen egen strömkälla. De kommunicerar endast då de är i närheten av en läsare. Att vara i närheten av en läsare innebär att taggen befinner sig inom det elektromagnetiska fältet. Inom detta avstånd alstrar fältet såpass mycket effekt till taggen att den aktiveras och skickar ut sin information. Informationen som skickas är chippets unika nummer. Detta med en annan frekvens än den som läsaren skickade ut. Den semipassiva taggen är som namnet indikerar en blandning av aktiv och passiv tagg. Den har en egen strömkälla men denna driver endast chippet och inte sändningen av signaler (Sweeney, 2005).

RFID använder de licensfria frekvensbanden och finns på flera olika frekvenser. De vanligaste är lågfrekvens 125 kHz, högfrekvens 13,56 MHz eller ultrahögfrekvens som ligger på cirka 433 eller 866 MHz i Europa. Lågfrekvensläsare har en läszon på cirka 10 cm, högfrekvensläsare upp till 1 meter och ultrahögfrekvensläsare ett antal meter. Detta gäller vid optimala förhållanden utan störningar. Höga frekvenser klarar av snabbare dataöverföring och längre kommunikationsavstånd men är mer känsliga för yttre påverkan, till exempel så kan metall störa radiovågorna (Sweeney, 2005).

3.2 RFID och säkerhet

Säkerheten har inte varit prioriterad när RFID-tekniken utvecklades. Detta har medfört att RFID-system inte använder sig av autentiseringsmetoder i någon större grad, vilket skapar säkerhetsrisker. Ett av problemen är att det inte går att förhindra läsning av taggen. När en tagg kommer i kontakt med en läsare av samma typ så sänder taggen sitt ID och läsaren tar emot informationen. Detta sker även om det är otillåten läsning. På så sätt kan en angripare läsa av en tagg och se vad den innehåller för information. Ett annat problemen är att det är relativt enkelt att kopiera taggarna. Det finns RFID-kopiatorer att köpa där man enkelt kopierar en tagg och skriver taggens id-nummer till en annan tagg. Då kan kopian användas för att få åtkomst (Henrici, 2008).

Enligt Henrici (2008) så skulle obehörig läsning och eventuellt kopiering av taggen behöva förhindras för att öka säkerheten i RFID-system. Idén är att ytterligare information ska behövas för att få taggens ID-nummer. Läsaren ska behöva lämna ut denna information för att visa att den är behörig att läsa taggen. Ett annat sätt är att taggen innehåller gömd data som inte skickas till läsaren men som ändå används i autentiseringen. Dessa lösningar skulle öka säkerheten men det kräver samtidigt ett större lagringsutrymme på taggen vilket skulle öka kostnaden.

Figur 3.2 RFID-kopiator (Absoluteconceptz, 2013)

En RFID-kopiator går att köpa enkelt via nätet, figur 3.2 visar ett exempel på en sådan kopiator. Det finns sätt att förhindra kopiering av taggar. En metod är att använda rullande kod. Det innebär att taggen har ett antal ID-nummer lagrade och att taggens information ändras enligt detta rullande schema efter varje korrekt läsning. Skulle en angripare kopiera taggen så kommer den ändå inte att fungera nästa gång den skannas. Angriparen skulle då behöva tajma in exakt rätt tid då just det id-numret som är kopierat ska skannas. Taggar med dessa säkerhetsfunktioner kräver mer ström och är dyrare att utveckla (Wiki1, 2013).

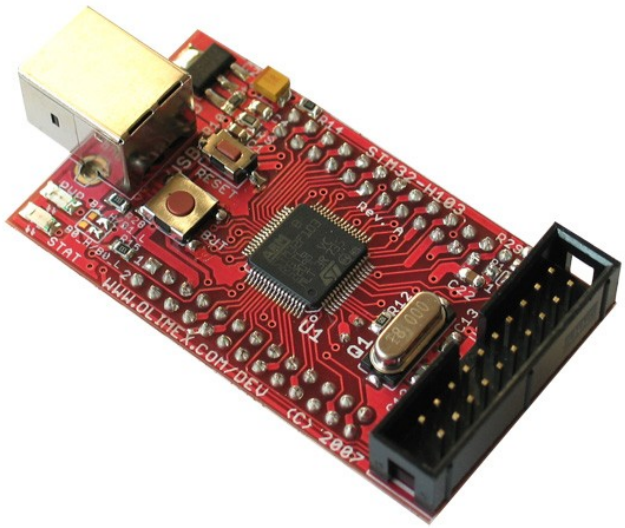
Ytterligare en säkerhetsbrist är att kryptering används av ytterst få RFID-system, all data skickas i klartext. RFID-taggar är uppdelade i klasser och endast klass 2-taggar har extra funktioner som till exempel kryptering (Glover & Bhatt). Det medför att en angripare kan tjuvlyssna på kommunikationen med hjälp av en RFID-läsare och till exempel få fram vad för information som skickas för att öppna en dörr. Anledningen till att kryptering inte används i större utsträckning är för att taggarna saknar processorkraft för att kryptera/avkryptera. Informationen behöver då krypteras innan det sparas på kortet. Det skulle också behövas mer lagringsutrymme på taggen och även mer processorkraft i läsaren där informationen avkrypteras (Henrici, 2008).

3.3 Microcontroller STM32F103

En microcontroller är en liten dator som är tillverkad i en integrerad krets. Microcontrollers används ofta till att konstruera inbyggda system där de kan styra olika processer. För att kommunicera har de ett antal ben som går att konfigurera som antingen input eller output. Program skrivs vanligtvis i C eller assembler på en dator för att sedan laddas till controllern. För att enkelt kunna programmera och använda sin processor används utvecklingskort som ofta har USB-kontakt för kommunikation med datorn och lättåtkomliga ben för smidig inkoppling av kortet till kretsen den skall användas i.

I det här projektet har microcontrollern STM32F103-RBT6 använts med en 32-bitars ARM Cortex-M3 RISC processor. Den arbetar med klockfrekvensen 72 MHz och har 128 kB programminne. Den kan kommunicera via USB, CAN, I2C, UART och SPI. Processorn är välanvänd och det finns gott om exempel på internet för att komma igång. För att snabbt kunna skriva kod och använda microcontrollern finns det ett bibliotek som heter Standard Peripherals Library vilket används för att hantera adresser och liknande.

Figur 3.3 visar utvecklingskortet STM32-H103 från Olimex med microcontrollern STM32F103RBT6. Kortet har förutom microcontrollern 52 ben, USB-kontakt och JTAG-kontakt.



Figur 3.3 Olimex microcontroller STM32-H103 (Olimex, 2013)

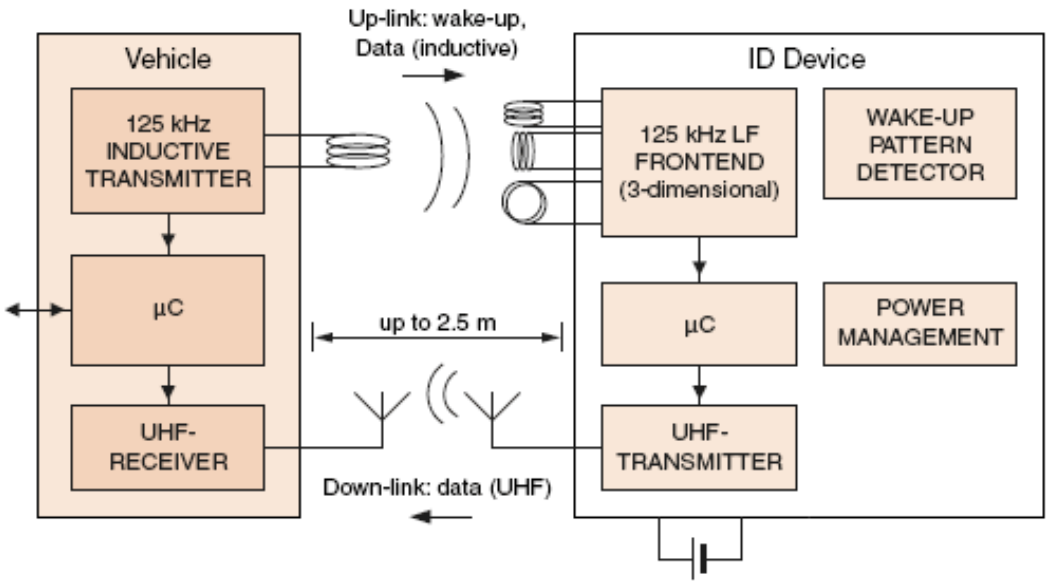
3.4 Passiva låssystem på eftermarknaden

Då projektet går ut på att utveckla ett passivt låssystem med hjälp av RFID är det viktigt att se vad som redan finns på marknaden. Ett sådant system går ofta under namnet passiv keyless entry, PKE, och innebär att föraren ska vara helt passiv. Det skall alltså inte behöva tryckas på något för att öppna fordonet.

3.4.1 Befintliga produkter

Det finns några passiva låssystem för fordon på marknaden och de flesta är uppbyggda på samma sätt. Kärnan i systemet är en microcontroller och läsare. Dessa ska kopplas ihop med bilens elsystem, såsom låssystem, larmsystem, lampor och tuta. Flera antenner kopplas också ihop med med den centrala enheten och placeras i framrutan, bakrutan och på sidorna av fordonet. Nyckeln och nyckellåset byts ut mot en knapp som startar bilen när den är nedtryckt, förutsatt att systemet känner av taggen.

Dessa system använder RFID UHF-teknik och aktiva taggar med rullande kod. RFID-modulen skickar ut en signal med frekvensen 125 kHz, LHF, och när taggen känner av signalen aktiveras den. Taggen är utrustad med en antenn som kan ta emot signalen. När taggen är aktiverad skickar den sitt ID med UHF. RFID-läsaren i fordonet tar emot taggens ID via en UHF-receiver och föraren får tillträde till fordonet. Bilden nedan (Figur 3.5) visar hur kommunikationen mellan RFID-modul och RFID-tagg fungerar.

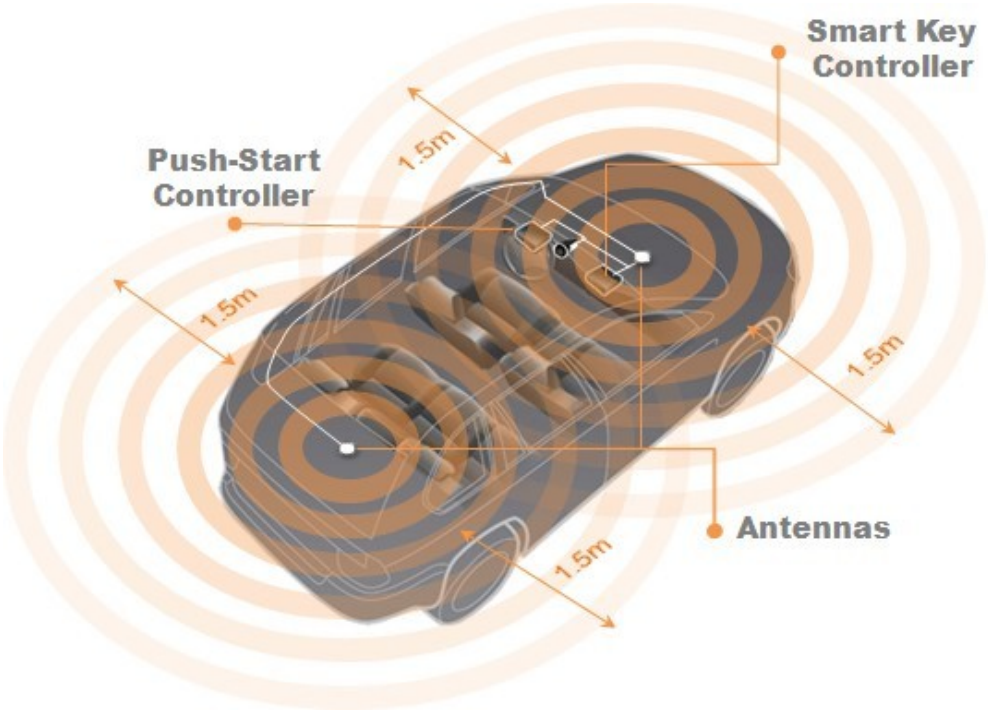


Figur 3.5 Kommunikation mellan RFID-modul och tagg (Advancedkeys2, 2013)

Räckvidden är beroende på var taggen är i förhållande till fordonet. Vanligtvis är det upp till 1,5 m. Då systemen är komplexa behövs bra kunskap om fordonet och dess elsystem för att installera systemet. Därför rekommenderas att arbetet utförs av kunniga mekaniker.

Lösningarna på marknaden varierar i pris och förmodligen även kvalité. Några av tillverkarna är:

- Advanced Keys: erbjuder kompletta system för 220 respektive 350 USD beroende på vilka funktioner man vill ha
- 2Go Keyless: erbjuder kompletta system för 400-700 USD beroende på funktioner och återsäljare.
- Directed: erbjuder ett komplett system för 1700 SEK.



Figur 3.6 Passivt låsstyrningssystem från Advanced Keys (Advancedkeys, 2013)

Bilden ovan (Figur 3.6) är ett exempel på ett passivt låsstyrningssystem som finns på eftermarknaden idag. De använder RFID UHF-teknik och räckvidden är upp till 1,5 m.

3.5.2 Skillnad mellan utvecklat system och marknadens

Det som skiljer produkterna som redan finns på marknaden från det system som ska utvecklas i detta projekt är att den utvecklade produkten endast kommer att styra låset till dörrarna. Motorn skall till exempel inte stängas av ifall den är igång utan det är endast dörren som skall låsas när föraren går iväg. En annan viktig skillnad är att lösningen inte kommer att kräva några ingrepp på fordonets befintliga elsystem. Det kommer istället vara en enklare lösning för styrning av låset. Mer om alternativ implementation kommer längre fram i avsnittet “Styrning av låssystem”.

4. Utveckling av automatiskt låsstyrningssystem

I följande kapitel förklaras utförande av projektet. Det kommer genomgås djupare beskrivningar av de olika delarna samt motivering av val. Även problem som uppstått kommer diskuteras och hur dessa har lösts.

4.1 Kravspecifikation

Som en del av detta projekt ska hela systemet kravställas med avseende på kostnad, funktion, säkerhet och tillförlitlighet. Vad gäller kostnaderna är målet att hålla dem nere. I avsnittet om befintliga system på marknaden kan man se att det går att köpa system som liknar det som utvecklas i detta projekt från 1700 kr och uppåt. Skillnaden är att de system som finns på marknaden även kräver en installation som behöver utföras av en yrkesman och kostnader för det tillkommer. Systemet som ska utvecklas i detta projekt kräver inte en så omfattande installation men det slutgiltiga priset bör vara konkurrenskraftigt.

Vad gäller funktion, säkerhet och tillförlitlighet så finns det krav beskrivna i tabellen som hittas i bilaga E. Det är en mall som används ofta inom produktutveckling. Mallen skapar en tydlig struktur där varje krav får ett ID-nummer, en titel, tillståndet före något har inträffat (preconditions), tillståndet efter något har inträffat (postconditions), beskrivning (description), vilka andra krav som behöver vara uppfyllda (predependency) samt en logisk förklaring till varför kravet behövs (rationale).

De nämnda kraven är att RFID-läsare och tagg ska kunna kommunicera (krav 1) samt att RFID-läsaren ska kunna kommunicera med microcontrollern (krav 2). Microcontrollern ska även kunna skilja mellan rätt och fel tagg (krav 3) och skicka lås- och lås upp-signal till fordonet (krav 4). Skulle microcontrollern ta emot två eller fler tagg-ID från RFID-läsaren så ska fordonet låsas upp om rätt tagg finns i närheten även om de andra taggarna inte har behörighet att låsa upp fordonet (krav 5). Vidare så ska fordonet låsas om rätt tagg lämnar läszonen (krav 6) och låsas upp om rätt tagg skannas av RFID-läsaren (krav 7). Slutligen skall en en tagg med fel ID inte kunna låsa upp fordonet (krav 8).

4.2 Utvecklingsmiljöer

En del av projektet är att skapa en struktur så att effektivt arbete kan genomföras. Valet av utvecklingsmiljöer är därför en viktig del. Det kommer att behövas en utvecklingsmiljö för att programmera utveckla mjukvara samt en miljö för att utveckla hårdvara.

4.2.1 Linux

Vad gäller valet av operativsystem så valde vi Linux, Ubuntu 12.04. Anledningen till det är de större möjligheter i valet av fri programvara vid arbete i Linux jämfört med Windows.

4.2.2 Eclipse och OpenOCD

Utvecklingsmiljöer som man kan använda för att programmera microcontrollern är den kommersiella CrossWorks och Open Source varianten där man använder Eclipse kombinerat med OpenOCD. Då Open Source-varianten är billigare föll valet på detta.

Eclipse är en utvecklingsmiljö som stödjer utvecklingen i många olika språk, bland annat C/C++ och Java. Genom att använda olika felmeddelanden hjälper Eclipse programmeraren att undvika syntaxfel och kompileringsfel. Eclipse kan enkelt installeras via hemsidan eclipse.org eller via Ubuntu Software Center.

OpenOCD står för Open On-Chip Debugger och är ett program som används för att programmera och debugga inbyggda system. Som med många andra Open Source system så är informationen om systemet bristfällig och det saknas användarmanualer. En enklare OpenOCD/Eclipse installationsguide skriven av en användare (Seng, 2013) fanns däremot tillgängligt.

Enligt den ovan nämnda guiden så ska OpenOCD och diverse bibliotek som är nödvändiga för att köra programmet installeras först. Det var inte problemfritt. Det behövs speciella versioner av vissa bibliotek för att installationen och systemet skulle fungera. Denna information inhämtades via diverse forum online. När de rätta biblioteken är installerade krävs att rätt JTAG-enhet installeras. JTAG är ett sätt att koppla in sig till microcontrollern och exekvera eller ladda mjukvara i processorn. JTAG:en som användes i projektet var en ARM-USB-TINY och även det var problematiskt att installera. Diverse filer behöver ändras då systemet per default är inställt att använda ARM-USB-OCD-H JTAG.

För att kunna ladda upp kod skriven i Eclipse till microcontrollern behövs ett flashprogram. Programmet som installationsguiden rekommenderar är stm32flash och det är det som användes i projektet. Stm32flash är ett program för GNU/Linux som ladda den kompilerade koden till microcontrollern. När man kompilerar ett program via Eclipse så kompileras det till den enheten som används om inget annat anges. Eftersom koden istället behöver kompileras till microcontrollern, så kallad cross-compiling, krävs ytterligare program bland annat GNU Binary Utilities, GNU C/C++ Compilers och GNU Debugger. Dessa kan laddas ner under namnet “Sourcery CodeBench”.

En annan viktig installation är biblioteket Standard Peripherals Library, ett bibliotek som används för att underlätta programmering av STM32 processorer. Med hjälp av Standard Peripherals Library kan programmeraren använda sig av färdiga funktioner och behöver inte sätta sig in i periferienheterna.

När allt är installerat behöver OpenOCD integreras med Eclipse för att kunna skriva kod, kompilera, flasha microcontrollern och köra och debugga. Sökvägar behöver skrivas och filer modifieras. Det tog ett tag innan kompilatorn och debuggern fungerade som tänkt men till slut fungerar systemet väldigt bra.

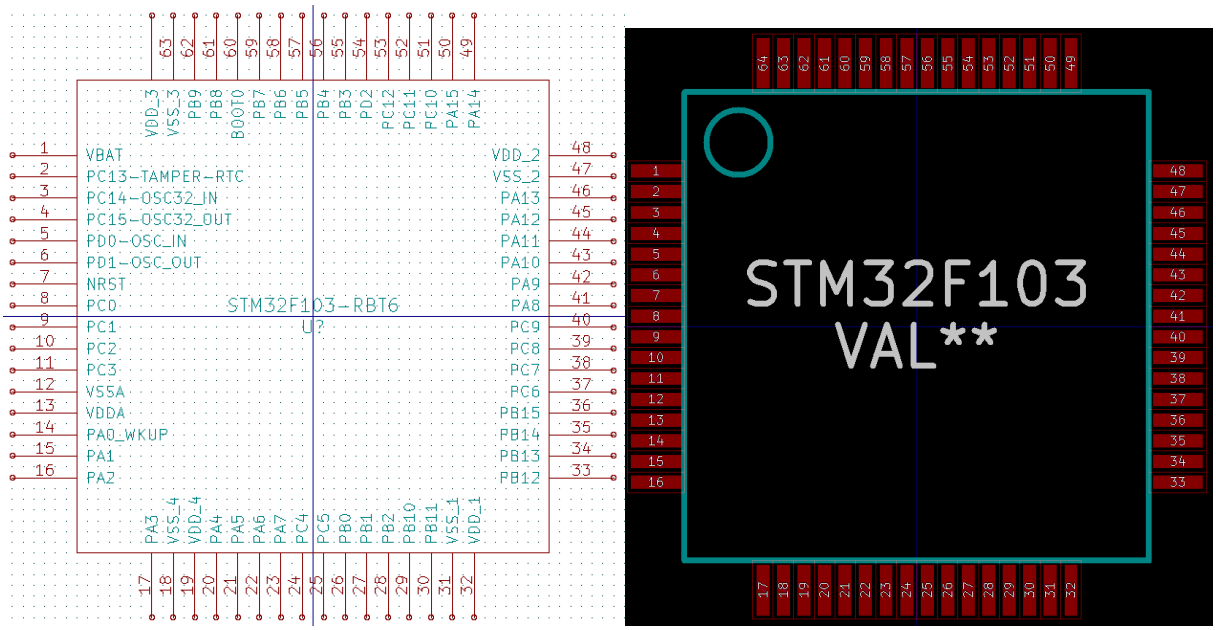
För att kunna programmera på ett smidigt sätt och hantera versionerna användes Bitbucket och Git. Git är ett versionshanteringsverktyg som ursprungligen skapades av Linus Torvalds för att hantera källkoden till Linuxkärnan. Versionshantering i sig innebär att tidigare versioner inte skrivs över utan att det bara är ändringarna som sparas. Det gör att tidigare versioner av till exempel källkodsfiler kan återskapas och ändringarna kan sparas. Ett sådant verktyg är väldigt användbart vid programmering. Anledningen till att valet blev Bitbucket var för att de tillåter privata projekt till skillnad från till exempel Github där det kostar att ha privata projekt.

4.2.3 KiCad

Det finns ett antal olika program för att konstruera layout för kretsar. I det här projektet stod det mellan de två programmen Eagle och KiCad. Eagle är ett program som är välanvänt av många i branschen. Det finns en gratisversion men det är då funktioner som saknas och betalversionerna är ganska kostsamma. Eftersom det till en början är oklart vilka funktioner som kommer behövas i projektet föll valet då istället på gratisprogrammet KiCad. KiCad har även det en stor användarbas och det finns gott om dokumentation att använda sig av på nätet.

Med hjälp av CAD-program såsom KiCad tas en design för tillverkning av PCB fram. I denna design är ledningsbanor dragna samt plats för borrhåll utmärkta. Med hjälp av ritningen kan sedan det färdiga kretskortet tillverkas genom lödning, borrning och montering av ledningsbanor.

Det finns i KiCad ett bibliotek med färdiga komponenter. Detta bibliotek är dock kraftigt begränsat. Det som istället behöver göras när en komponent som används inte finns med i biblioteket är att konstruera en egen. Detta görs i två steg, först skapas en schematisk ritning och sedan ett footprint som är det som används på PCB-layouten. I den schematiska ritningen läggs endast ben med namn och numrering till. Nedan till vänster syns den schematiska ritningen för processorn som använts i projektet. I fotavtrycket placeras däremot benen även enligt de avstånd som den fysiska processorn har. Information för fotavtryckets mått skall framgå ur databladet för komponenten.



Figur 4.1 Schematisk ritning (vänster) och footprint (höger) för processorn

När komponenterna finns att tillgå kan kretsen ritas upp. Även här ritas först en schematisk bild upp innan PCB-layouten konstrueras. I bilaga A och B finns bild på den schematiska ritningen samt den färdiga PCB-layouten. För att gå från schematisk krets till PCB-krets behöver varje komponent associeras med ett fotavtryck. Sedan placeras alla fotavtryck ut på lämpliga platser och hela kortet som kretsen ska monteras på behöver märkas ut. Med alla komponenter på rätt plats skall kopparbanor dras mellan komponenterna. Detta kan göras både manuellt och med en så kallad autorouter. Det kan vara en bra idé att använda en kombination av dessa två metoder genom att rita kritiska banor manuellt och låta programmet rita resten. Det sista som har gjorts är att fylla ut ett jordplan som sammankopplar alla komponenters jord.

Eftersom det som har tagits fram är en prototyp som inte har den färdiga funktionaliteten så finns inte behovet av att tillverka kretskortet än. Det är dock en start för fortsatt påbyggnad. För att sedan tillverka kortet finns det flera olika företag som kan utföra detta om man skickar in ritningen.

4.3 Valet av RFID

En av de första utmaningarna med projektet var att välja rätt RFID med avseende på avstånd, pris och säkerhet. Syftet med projektet var att skapa ett system där dörren till fordonet låses upp när en person med rätt bricka närmar sig. Dörren skall vara upplåst innan personen kommer fram till den. Likaså skall dörren låsas när personen går ifrån fordonet. Det innebär att ett RFID-system med några meters kommunikationsavstånd mellan läsare och tagg behöver användas.

Enligt information som finns tillgänglig så skulle det vara optimalt att använda RFID som är anpassad till ultrahög frekvens (UHF) som i EU är 868 MHz. Det innebär att kommunikationsavståndet mellan läsaren och en passiv tagg skulle kunna vara upp till 3 meter och mellan läsaren och en aktiv tagg upp mot 10 meter. Detta är under optimala förhållanden då störningarna är minimala. Aktiva kort har starkare signalstyrka på grund av sitt inbyggda batteri så läsningsavståndet skulle inte vara ett större hinder. Däremot blir kostnaden högre för aktiva taggar och de förbrukar även batteri vilket gör att dessa ibland behöver bytas.

Problemet med UHF är att utbudet är väldigt begränsat jämfört med RFID med lägre frekvenser. Denna lösning är även förhållandevis kostsam. En billig UHF läsare kostar ca 3000 - 10000 kr. Då gäller det även att hitta en som tål låg temperatur och som kan kommunicera med processorn via ett protokoll. Tittar vi på högfrekventa (HF) system så sjunker priserna på läsare och hamnar på 500 - 5000 kr. Även utbudet ökar. Nackdelen är att kommunikationsavståndet mellan läsaren och tagg sjunker till max 1 meter. Med den här lösningen så skulle chauffören i princip behöva stå vid dörren för att läsaren ska känna av taggen och fordonet ska låsas/låsas upp.

Valet stod mellan dessa system. Frågan här är vad kunden vill ha. Vill man ha ett dyrare system som är mer användarvänligt och där fordonet är upplåst när chauffören kommer fram till den. Eller vill man satsa på ett billigare system där chauffören får stå vid dörren för att den ska låsas upp. Avståndet skulle kunna minimeras ifall antennen sätts fast på förardörren, till exempel på fönsterrutan. Låsningen borde däremot inte vara ett problem, fordonet kommer att låsa sig när taggen hamnar utanför läszonen.

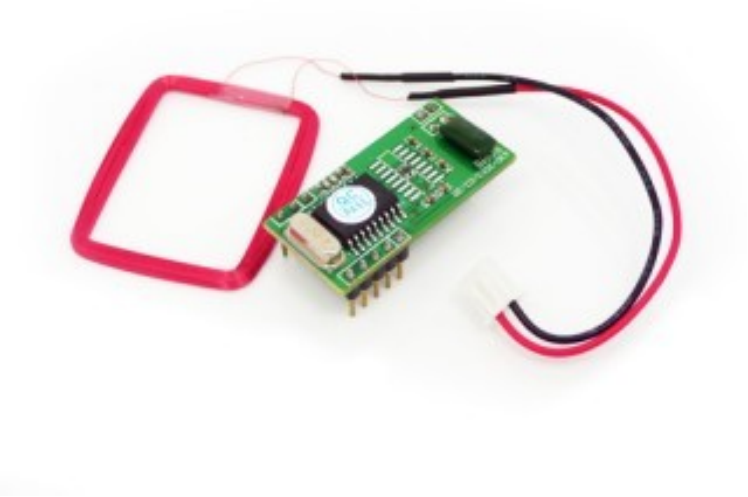
Efter samtal med Björn Bergholm som representerar kunden beslutades det att målet är ett system där dörren var upplåst när personen kom fram till den. Detta ska implementeras i fordon där systemet fungerar i bakgrunden utan att märkas. Det extra arbetsmoment som skulle uppstå då man låser/låser upp fordonet skulle inte accepteras.

4.3.1 Första RFID-modul - kort räckvidd

Efter att noggrant sökt efter en UHF RFID-läsare till ett lågt pris utan resultat beslutades det att välja en lågfrekvent läsare som kommunicerar via RS-232. Anledningen till det var att RS-232 är

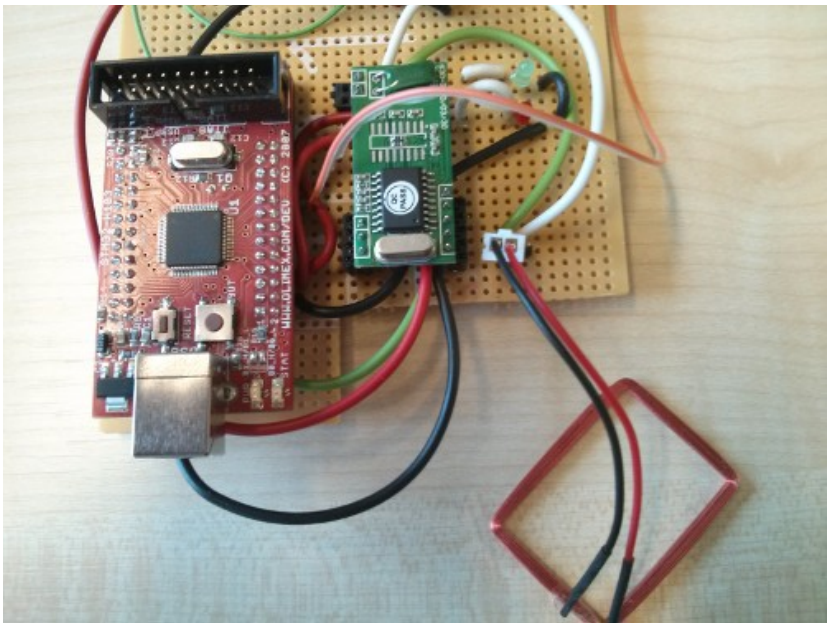
en av de vanligaste gränssnitten och finns för både låg- och ultrahögfrekvensläsare. Målet blev att utveckla en prototyp som fungerar på kort avstånd, få RFID att kommunicera med microcontrollern som i sin tur kommunicerar med fordonets låssystem. Med ett sådant fungerande system skulle man kunna byta ut lågfrekventläsaren mot en ultrahögfrekventläsare som kommunicerar via RS-232.

RFID som valdes att börja arbeta med var en RDM630 med RS-232 gränssnitt. Den har en frekvens på 125 kHz och fungerar på cirka 5 cm avstånd. Den har kopplingar endast för TX och RX och inga kontrollkopplingar vilket innebär att det inte kommer kunna användas några hårdvaruhandskakningar. Modulen ses i bilden nedan (figur 4.2).



Figur 4.2 RFID-modul RDM630 (Electrokit, 2013)

Arbetet inleddes med att kontrollera att taggarna kunde kommunicera med RFID-läsaren. Med hjälp av ett oscilloskop som var kopplat till RFID-läsarens sändarben (TX) kunde det ses att signaler skickades när taggen skannades. Nästa steg var att tillverka en första krets bestående av RFID-läsaren, microcontrollern samt en grön och en röd LED. Kretsen som användes syns nedan (figur 4.3).



Figur 4.3 krets innehållandes microcontrollern och RFID-modul

Databladet får RDM630 uppger att RFID-läsaren har en baudhastighet på 9600, en startbit, en stopbit och ingen paritetsbit (RDM630, 2013). Detta behöver översättas till kod så att microcontrollen kan ta emot rätt information.

4.4 Kommunikation mellan RFID-läsare och microcontroller

I denna del beskrivs tillvägagångssättet för att få igång kommunikationen mellan microcontrollern STM32F103 och RFID-läsaren. De problem som uppstod kommer att beskrivas samt lösningen av dessa problem.

4.4.1 Programmering av microcontrollern STM32F103

För att underlätta programmering av microcontrollern användes biblioteket Standard Peripherals Library. Biblioteket används av STM32 processorer för att på ett enklare sätt initiera till exempel klockor och input/output.

Problem dök upp i samband med att programmeringen av microcontrollern inleddes. Kommunikationen med läsaren fungerade inte. Kodexempel visade att klockorna behövde initieras men det var oklart vilka av klockorna som skulle initieras och hur. Första varianten av den skrivna koden innehöll en breakpoint när microcontrollern tog emot information, vilken information som helst. När koden provkördes visade oscilloskop att RFID-läsaren skickade signaler men microcontrollern tog inte emot någon information.

I ett tidigare skede i projektet skrevs kod som fick lampor att blinka och då behövdes registret General Purpose Input/Output (GPIO) initieras. I initieringen anges vilket register och vilken pinne som skall användas. Även klockan som var kopplad till det registret behövde initieras. När liknande USART-initiering genomfördes fungerade inte kommunikationen. Efter att ha studerat Standard Peripherals Library upptäcktes ytterligare en klocka. Även en USART-klocka behöver initieras. Efter initieringen av denna klocka blev resultatet positivt. Det visade sig att förutom att initiera det USART-registret som skulle användas och klockan kopplat till det registret behövde ytterligare en generell USART-klocka initieras för att programmet skulle köras.

Programmet fungerade delvis. Det körde och tog emot information av RFID-läsaren. Detta kunde ses genom att programmet stannade av när det nådde en breakpoint i koden. Microcontrollern fick visserligen input från RFID-läsaren men det var bara en byte och det saknades verktyg för att se vad det var som togs emot.

Debuggern i Eclipse fungerade inte fullt ut. Den visade inte vad för värden som variablerna hade, till exempel de värden som togs emot av RFID-läsaren. Det var nödvändigt att få igång den funktionen innan programmeringen av processorn kunde fortsätta. Mycket tid lades ner på att söka efter information för att få igång detta. Eftersom OpenOCD användes fanns det inga manualer. Lösningen hittades genom att söka på nätet på diverse forum och pussla ihop informationen tills debuggern fungerade. Arbetet med att få igång debuggern tog tid men utan den funktionen hade inte arbetet kunnat fortsätta.

Då debuggern var igång blev det tydligt vilka värden microcontrollern fick av RFID-läsaren. Breakpointen i debuggern störde sändningen, RFID-läsaren fortsatte att sända men microcontrollern tog inte emot några fler tecken eftersom processen var pausad. Det hade orsakat problem med att microcontrollern endast tog emot data första gången taggen skannades.

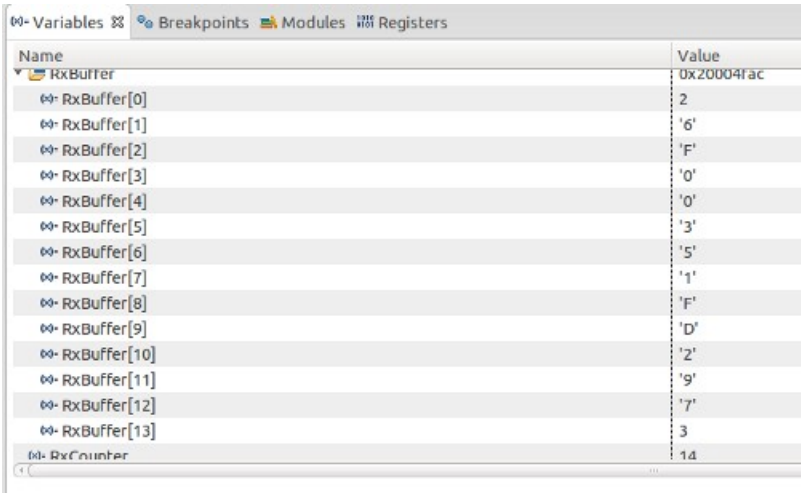
4.4.2 RS-232 Kommunikation

Som tidigare nämnt så kommunicerar RFID-läsaren RDM630 med microcontrollern via RS-232. Den har bland annat en pinne för att ta emot data (Rx), en för att skicka data (Tx), en för jord och en för 5V matningsspänning. Rx och Tx är kopplade till motsvarande pinnar på microcontrollern STM32-H103.

Då RS-232 använder asynkron kommunikation och har ett så kallat start/stop-dataformat så skickas data endast när RFID-läsaren får input från en tagg. När läsaren känner av taggen via antennen skickas taggens ID vidare till microcontrollern. RFID-läsaren skickar följande:

- 1 ASCII-tecken, en tvåa betyder start of text
- 10 ASCII-tecken, varav två slumptecken och åtta tecken som anger kortets ID
- 2 ASCII-tecken, checksumman
- 1 ASCII-tecken, en trea betyder end of text

I de första försöken användes en tagg med numret 3481554 och läsaren tog emot 14 bytes i form av decimala tal: 2, 54, 70, 48, 48, 51, 53, 49, 70, 68, 50, 57, 55, 3. Bilden nedan (figur 4.4) visar output från debuggern i Eclipse.



Figur 4.4 Output från debuggern

Det första tecknet (2) är ASCII-tecknet för “start of text”. Sedan följer två hexadecimala slumptecken (6F). De åtta kommande tecknen anger taggens ID som en hexadecimal siffra (00351FD2). 351FD2 omräknat till decimaltal blir 3481554 vilket är kortets id. Efter det följer två tecken (97). Dessa är checksumman som räknas ut genom en XOR-operation. Det sista tecknet (3) är ASCII-tecknet för “stop of text”.

Tecknen skickas i hexadecimal form. Varje tecken i sin tur består av:

0	x1	x2	x3	x4	x5	x6	x7	x8	1

1 startbit (nolla)

8 bitar som motsvarar ett ASCII-tecken

1 stoppbit (etta)

Baudhastigheten är 9600 bitar/sekund och det används inte någon hårdvaruhandskakning. Checksumman räknas ut genom en XOR operation, disjunktion, mellan de tio tecken som motsvarar taggens ID. XOR innebär enkelt beskrivet att två olika tecken ger en etta och två lika tecken en nolla. Se tabell nedan (tabell 4.1).

Input A	Input B	Output
0	0	0
0	1	1
1	0	1
1	1	0

Tabell 4.1

Nedan följer ett exempel på vad man får för hexadecimalt tal om man gör en XOR-operation mellan de tio tecken i taggens ID.

$$6F_{16} \oplus 00_{16} \oplus 35_{16} \oplus 1F_{16} \oplus D2_{16} = 97_{16}$$

Checksumman stämmer alltså med det avlästa värdet.

När kommunikationen mellan RFID-läsaren och microcontrollern fungerade anpassades koden för att lättare kunna se att endast rätt tagg kunde få access. Detta görs genom att använda de två lampor som är fastlödda på experimentkortet. Den gröna lampan skall lysa vid rätt tagg och den röda vid fel. I första steget hårdkodades taggens nummer, vilket fick rätt tagg att tända den gröna. I andra steget användes istället en knapp med funktionen att para ihop tagg och läsare

då knappen trycks ned. Innan ihopparningen lyser inte grön lampa vid kontakt med någon av taggarna. Efter ihopparning av en tagg så lyser däremot den gröna lampan när den taggen avläses.

Som det fungerar nu så behöver taggen paras ihop varje gång läsaren startas. Denna lösning kommer inte att fungera med den färdiga produkten. Därför kommer ett icke-flyktigt minne, EEPROM, att kopplas samman med microcontrollern. Taggarnas ID skall sparas på det dataminnet även när systemet är spänningslöst. Mer om detta i nästa avsnitt.

4.4.3 Externt minne - EEPROM

För att kunna ha ett system där tagg-ID sparas även när microcontrollern är avstängd behövs ett externt icke-flyktigt minne. Till det syftet kommer ett EEPROM-minne att användas, mer bestämt Microchip 93C66C 4Kbit. EEPROM står för Electrically Erasable Programmable Read-Only Memory och är ett minne som kan raderas elektroniskt. De använder sig oftast av seriell överföring och de vanligaste gränssnitten är SPI och I2C.

Eftersom SPI är enklare att använda än I2C kommer SPI att användas i kommunikationen mellan EEPROM-minnet och microcontrollern. Serial Peripheral Interface Bus eller SPI är ett full-duplex synkront seriellt gränssnitt som använder master/slave principen. Det innebär att en enhet kontrollerar en eller flera andra enheter. I det här fallet kommer microcontrollern att vara master och EEPROM-minnet att vara slav. Då SPI är ett synkront gränssnitt finns det en klocka som synkroniserar dataöverföringen mellan sändare och mottagare (Bengtsson, 2009).

När SPI används kopplas minnet till microcontrollern med minst sex ledningar. Den första används för att skicka klocksignalen från microcontrollern till minnet (SCK). Den andra för att skicka data från mastern till slaven (MOSI) och den tredje för att skicka data från slaven till master (MISO). En ledning används för matningsspänning och ytterligare en för att jorda enheten. Den sista ledningen används för att ange om minnet ska vara aktivt eller i stand-by läge, så kallad chip select (CS). Om en etta ges är minnet aktivt medan en nolla sätter minnet i standby-läge. Ytterligare en sjunde pinne används för just modellen 93C66C för att ange om 8-bitars eller 16-bitars organisation ska användas. En etta på denna port indikerar att 16-bitars register kommer att användas medan en nolla anger att 8-bitars ord kommer att användas (Microchip, 2011). EEPROM-minnet som används i projektet är 4 kbit stort och då ASCII-tecken används räcker det med 8-bitars långa minnesplatser. Det innebär att det kan maximalt skrivas 512 tecken i minnet.

EEPROM-minnet som används i detta projekt ska till skillnad från andra EEPROM ha en hög signal på CS när den ska skicka data. På grund av detta kan det uppstå svårigheter vid användning av EEPROM-minnet.

När man läser från, skriver till eller raderar minnet så ska en kod skickas först som anger vad man vill göra. En så kallad OP-kod. Koden består av tre till fem bitar, till exempel 101 (skriva) och 110 (läsa). Sedan följer adressen till vilken man skriver/läser från. Efter adressen skickas data (Microchip, 2011).

Det ger följande vid exempelvis skrivning: OP-koden 101 följt av 9 bitar adress följt av 8 bitar data.

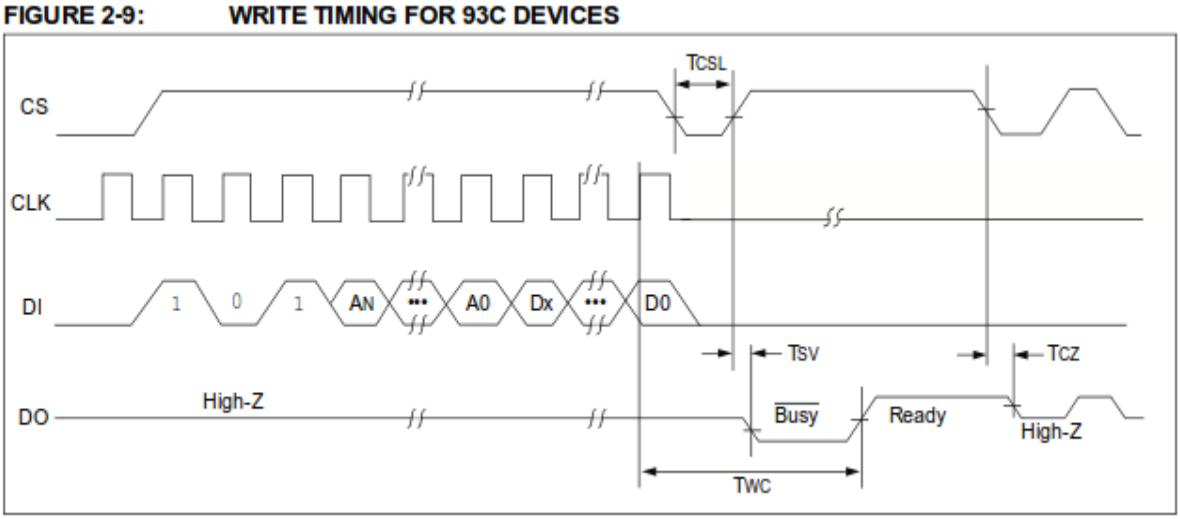
1	0	1	x	x	x	x	x	x	a1	a2	a3	a4	a5	a6	a7	a8	a9	d1	d2	d3	d4	d5	d6	d7	d8

Det innebär att bitarna kommer att behöva skiftas eftersom det endast går att skicka hela bytes och inte enstaka bitar. Adress- och databitarna behöver förskjutas för att det ska bli rätt.

1	0	1	a1	a2	a3	a4	a5	a6	a7	a8	a9	d1	d2	d3	d4	d5	d6	d7	d8	0	0	0	0

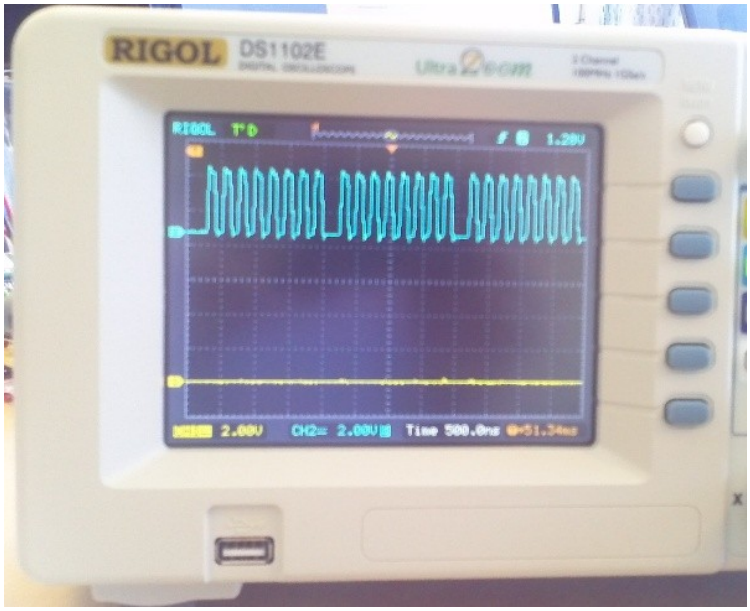
När microcontrollern skickade OP-koden för läsning och adress där läsningen skulle ske så stannar klockan av. Lösning på det problemet är att fortsätta att skicka dummy-nollor för att hålla igång klockan. Då går det att läsa från EEPROM-minnet. Det som microcontrollern tog emot var endast ettor vilket innebar att skrivningen till minnesplatsen inte gått rätt till.

Enligt databladet (Microchip, 2011) skall MISO, eller data output (DO) som det kallas i databladet vara etta för att indikera att skrivningen till minnet har fungerat och att den är redo att ta emot nästa kommando. Figuren 4.5 visar hur det ska se ut.



Figur 4.5 Skrivschema för 93C66C

Jämförs detta med outputen från oscilloskopet så kan man se att data output är noll (figur 4.6). Det innebär att det inte går att skriva till minnet och det enda som minnet returnerar är ettor eftersom det är skrivet på minnet när det är raderat. Lösningen är väldigt simpelt. EEPROM-minnet är skrivskyddat per default och behöver ställas om till skrivläge. En viss bitsekvens behöver skickas till minnet för att kunna skriva till det.



Figur 4.6 Output från oscilloskopet

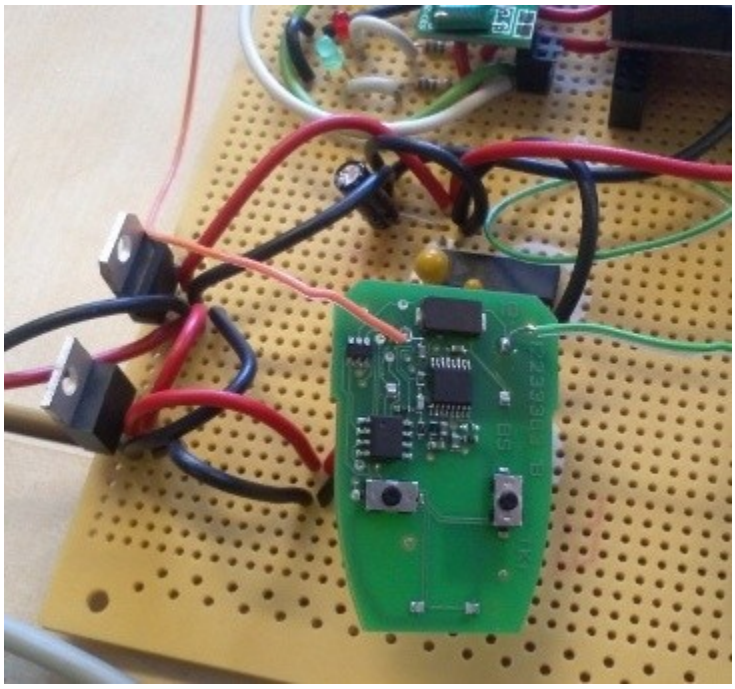
När data läses från EEPROM-minnet så är ordet förskjutet med tre bitar. Tecknet som är 8 bitar är splittrad över två bytes. Lösningen var att skriva en funktion som skiftar data tre bitar. Skiftningen blir nödvändig även vid sändning.

4.5 Styrning av låssystem

För att kontrollera bilens låssystem finns det olika tillvägagångssätt. Det första alternativ som kom upp var att koppla in systemet på fordonets CAN-buss. Då hade microcontrollern skickat information som säger till fordonet att låsa eller låsa upp dörren. Denna lösning övergavs eftersom det är problematisk att koppla in sig på CAN-bussen i moderna fordon. Det hade även krävts olika programkod för olika fordonsmodeller eftersom systemen skiljer sig däremellan.

Det andra alternativet till lösning var att gå runt bilens kommunikationssystem och direkt styra låssystemet. Detta hade gjorts genom relästyrning av låsmekanismen. I jämförelse med att koppla in sig på CAN-bussen hade detta varit ett enklare val då det endast är spänningssättning av låsmekanismen som krävs och ingen avancerad kommunikation. Men även detta kan skapa konflikt på elsystemet då det påkopplade systemet låser upp fordonet när det själv tycker att det ska vara låst. Det kan då bli odefinierat om låset skall vara låst eller upplåst.

Lösningen som istället implementerats i systemet är att styra fjärrkontrollen till det befintliga fjärrlåset. Eftersom fjärrlåset redan är inbyggt i fordonet kommer inga konflikter uppstå och lösningen går att anpassa till alla fordonsmodeller så länge det finns ett fjärrlås installerat. Genom att koppla bort knapparna för lås- och upplåsning på fjärrkontrollen har det istället kopplats på styrsignaler från microcontrollern. Bilden nedan (figur 4.7) visar fjärrkontrollen där knapparna har ersatts av styrning från microcontrollern.

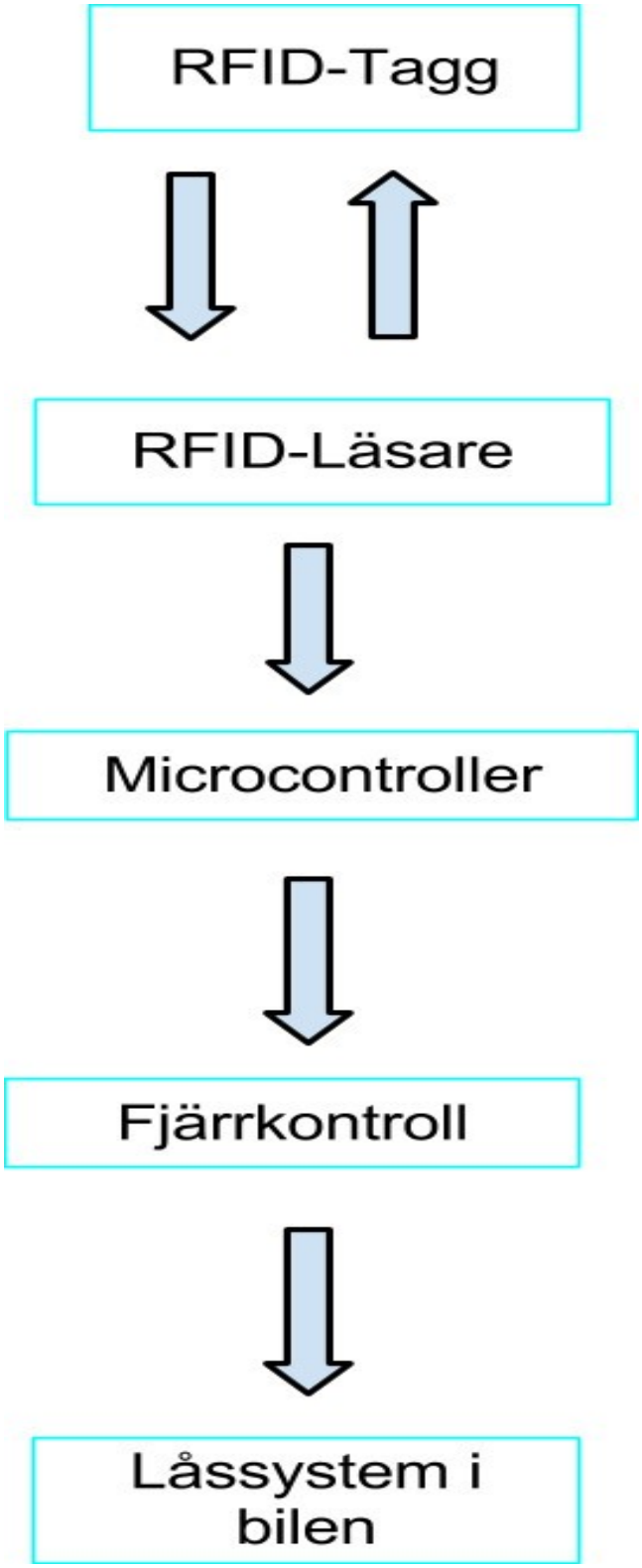


Figur 4.7 Avskalad fjärrkontroll

För att kontrollera att fjärrkontrollen fungerar som tänkt behövs en mottagarmodul. Optimalt hade varit att testa detta i ett riktigt fordon men eftersom det inte var möjligt användes istället en mottagarmodul (figur 4.8). När RFID-läsaren känner av taggen skickar den taggens nummer till microcontrollern som kontrollerar det. Om det är rätt ID så skickas det en etta till fjärrkontrollen för upplåsning. Fjärrkontrollen skickar i sin tur en trådlös signal till mottagaren på frekvensen 433,92 MHz. Hade mottagaren varit placerad i ett fordon och ihopkopplad med det centrala låssystemet så hade fordonet låsts upp. Så länge RFID-läsaren känner av taggen förblir fordonet upplåst. När den inte längre känner av taggen väntar den i ett av programmeraren bestämt tidsintervall och om inte rätt tagg känns av under denna tid så låses fordonet.

Figur 4.8 Mottagare 433,92 MHz

Figur 4.9 nedan visar hur kommunikationen fungerar i den slutgiltiga produkten. Taggen och RFID-läsaren kommunicerar två vägar med varandra medan kommunikationen mellan RFID-läsaren och microcontrollern är enkelriktad. Microcontrollern tar endast emot data från RFID-läsaren men skickar inget tillbaka. Även kommunikationen mellan microcontrollern och fjärrkontrollen är enkelriktad. Fjärrkontrollen får in en etta på antigen lås- eller låsuppknappen. Signalen skickas sedan till fordonets låssystem, även där är kommunikationen enkelriktad.



Figur 4.9 Blockschema som visar kommunikationen i systemet

4.6 RFID - lång räckvidd

Då utbudet av RFID-moduler som verkar inom UHF är både kostsamt och ganska begränsat blev valet att använda en RF-modul. Med RF-moduler erbjuds ett större utbud och priset sjunker i jämförelse med motsvarande RFID-moduler. Istället bli det mer att göra själv och det är inte lika “paketerat” som en RFID-modul. Detta väger dock upp prisskillnaden och det låga utbudet.

4.6.1 Valet av RF-modul

Radiovågor är en typ av elektromagnetisk strålning, en vågrörelse som fortplantas i tid och rum. Det är den mest lågfrekventa formen. Radiofrekvens är svängningar som ligger mellan 300kHz och 300 MHz.

Modulering är olika metoder som används för att överföra information via radiofrekvenser. De vanligaste är frekvensmodulering och amplitudmodulering. Vid frekvensmodulering (FM) ändrar man frekvensen beroende på originalsignalen. Vid amplitudmodulering (AM) är det istället amplituden som ändras. Amplitudmodulering är känsligare för störningar än frekvensmodulering. Dessa är två former av analog modulering. Motsvarigheterna för digital modulering är Frequency-shift keying (FSK) och Amplitude-shift keying (ASK).

Skillnaden mellan analog och digital modulering är att digital modulering endast varierar mellan två värden och på så sätt översätter binära ettor och nollor till analoga signaler. FSK varierar mellan två olika frekvenser och ASK varierar mellan två olika amplituder.

I sökandet efter RF-modul fanns både FSK- eller ASK-modulerade moduler. Ett tredje alternativ var RF-moduler med både FSK- och ASK-modulering. Då målet var att upprätta kommunikation mellan RF-modulen och taggen behövde dessa vara modulerade på samma sätt. Passiva taggar är oftast ASK-modulerade och aktiva taggar är oftast FSK-modulerade. Då frekvensmodulering är mindre känslig mot störningar hade det varit optimalt att använda FSK vilket innebär användning av aktiva taggar. Dessa skiljer sig rätt mycket i pris. Passiva taggar kostar cirka 20 kronor medan aktiva cirka 300 kronor. Ett av kraven med projektet var att hitta en så billig lösning som möjligt och då skulle passiva taggar passa bättre, förutsatt att de fungerar på några meters avstånd. Däremot blir system känsligt för störningar.

Efter att ha presenterat de olika varianterna för Björn Bergholm fattades beslutet att satsa på ett system med passiva taggar. Det som beställdes var en ASK-modulerad RF-transceiver med inbyggd antenn och en passiv tagg. Både taggen och modulen är anpassade till den europeiska UHF-frekvensen 868MHz. Tanken är att få systemet att fungera med dessa produkter samt testa om det räcker med passiva taggar. Ett FSK-modulerat system med aktiva taggar kan komma att bli aktuellt om räckvidden inte är tillräckligt lång eller om det blir för omfattande störningar.

4.6.2 Implementering av RF-modul

Till skillnad från den första RFID-modulen som användes så är inte RF-modulen anpassad för att direkt fungera som RFID. Detta innebär till exempel att signalen som aktiverar taggen inte automatiskt sänds ut från modulen. Denna signal behöver modulen bli tillsagd att skicka ut. Även hantering av mottagen data från taggen behöver manuellt konfigureras.

Det behövs tvåvägskommunikation för att använda modulen till att både skicka och ta emot data. Eftersom ingen information angående full duplex i aktuell processor har hittats initieras därför processorns USART i halv duplex.

För att modulen ska sända radiomeddelanden krävs det att den även får ett antal bytes utöver meddelandet. Den kräver två bytes för synkronisering, en byte för både funktionen och längden på meddelandet, en byte som säger läget för modulen, bytes för meddelandet och ID:t samt två bytes för status och kontrollsumma. Detta står även beskrivet i användarmanualen för modulen (TCM320, 2013).

Funktionen med att använda RF-modulen som RFID har inte hunnit implementeras inom exjobbets ramar. Efter att meddelande skickas så tas ingen signal emot från taggen. Det finns ett flertal tänkbara felkällor där ett första är att modulen inte skickar ut någon RF-signal. Ett annat tänkbart fel kan vara att taggen sänder data på ett format som modulen inte stödjer. Dessa fel är väldigt svåra att urskilja då det i projektet saknas mätutrustning för att kunna se om signaler skickas.

Efter flera försök visade det sig tillslut att RF-modulen som användes var ett felköp. Det är inte en “vanlig” RF-modul. Det är en produkt med ett speciellt applikationslager utvecklat av EnOcean Alliance. Kommunikationen på applikationslagret sker med så kallade EnOcean-radiotelegram vilket innebär att modulen endast kan kommunicera med andra EnOcean Alliance-produkter. Det går inte att använda produkten då taggar inte använder det protokollet. Deras produkter är för övrigt anpassade för att tända, släcka och dimma lampor trådlöst. Den använder sig av radiostandaren ISO/IEC - 14543-3-10, en standard utvecklat för trådlösa applikationer med minimal strömförbrukning.

5. Diskussion och slutsatser

Syftet med detta projekt var att utveckla ett passivt automatiskt låsstyrningssystem för fordon med hjälp av RFID-teknologi. Målet var att dörrarna till fordonet ska låsas upp när RFID-läsaren skannar av rätt tagg och låsas när den inte längre gör det. Vidare så skulle endast rätt tagg ha möjlighet att låsa upp fordonet.

De specifikationer som fanns med vid projektets start var bland annat att kravställa systemet med avseende på funktion, säkerhet och tillförlitlighet samt att välja rätt RFID med avseende på dessa krav. Dessa krav specificerades i början av projektet och modifierades under arbetets gång, (bilaga E). Åtta övergripande krav finns i kravlistan. Med undantag för krav 1 som rör avståndet är kraven uppfyllda, men samtidigt är krav 1 en förutsättning för de andra kraven så systemet anses inte fungerande tills det kravet är uppfyllt. Tittar vi på de andra kraven så kan RFID-läsare och microcontroller kommunicera (krav 2). Microcontrollern kan skilja mellan rätt och fel tagg (krav 3) och kan skicka lås/lås upp-signal till fordonet via fjärrkontrollen (krav 4). Rätt tagg låser upp fordonet även om fel tagg finns i närheten (krav 5). Fordonet låses när RFID-läsaren inte längre känner av taggen (krav 6) och låses upp när den känner av den (krav 7). Systemet är även säkert i viss bemärkelse då fel tagg inte kan låsa upp fordonet (krav 8).

Att välja en RFID-modul utifrån dessa krav visade sig vara problematiskt. Det största problemet var att hitta en billig RFID-modul som fungerar på långt avstånd, minst 2 meter. Detta problem löstes inte under projektets gång. Att överhuvudtaget söka efter komponenter och köpa komponenter som fungerar ihop med andra produkter var något som vi inte hade någon erfarenhet utav. Det visade sig att det tar väldigt långt tid att hitta produkter som är kompatibla med varandra och som har rätt funktion. Många produkter saknar datablad och andra har kortfattade och bristfälliga datablad. Detta försvårar valet av produkt då det är riskfyllt att köpa något utan att kunna hitta någon information om hur produkten fungerar. Detta fick vi erfara under projektets gång. Först i samband med sökandet efter en RFID-modul och sedan i sökandet efter RF-modul. Bristfälliga datablad ledde delvis till att fel RF-modul köptes in.

En annan specifikation var att välja microcontroller. Vi hade tidigare arbetat med ATmega328 (Arduino) och PIC. Hade vi valt mellan dessa hade det blivit en PIC-processor eftersom Arduino inte använder rent C som utvecklingsspråk och är delvis begränsad i sin funktion. Ingenjörerna på Broccoli som vi kom i kontakt med introducerade oss till ARM-processorn STM32F103 och då det redan fanns exemplar av microcontrollern på kontoret valde vi den. STM32F103 är ingen nybörjarprocessor utan kräver en del av utvecklaren. Bland annat ska olika klockor ställas in för att olika funktioner i processorn ska fungera.

En av specifikationerna för projektet var också att välja utvecklingsmiljö för utveckling av mjukvara. Efter att ha konsulterat anställda på Broccoli som har erfarenhet av att arbeta med microcontrollern STM32F103 reducerades valen till antigen ett kommersiellt program eller en

Open Source-variant. Valet blev Open Source-varianten där Eclipse används kombinerat med OpenOCD.

Det uppstod vissa svårigheter att få igång OpenOCD och få det att fungera med Eclipse. Dels för att det saknades dokumentation och dels för att den dokumentationen som fanns inte var uppdaterad. Det finns många fördelar att arbeta med Open Source-system, bland annat tillgängligheten och att man som utvecklare kan vidareutveckla programmen och anpassa dem till sitt syfte. Nackdelen är att det krävs en del av användaren, både kunskapsmässigt och tidsmässigt. En annan nackdel är det bristande dokumentationen.

En del av projektet var även att testa mjukvaran. Vad gäller den punkten så har endast funktionalitetstester genomförts där vi har tittat på att programmet fungerar som det är tänkt. På grund av tidsbrist har inte omfattande enhetstester skrivits.

Ett mål har varit att få fram ett färdigt kretskort av prototypen. Först beslutades vilken programvara som skulle användas. Sedan tidigare kände vi till programmet Eagle som är väldigt vanligt och vi blev tipsade om att det fanns ett program som heter KiCad av Henrik Brenander. Eftersom Eagles gratisversion har vissa begränsningar som till exempel antal lager och storlek på kretskortet så blev valet det fria programmet KiCad, eftersom vi inte visste om begränsningarna i Eagles gratisversion skulle bli något problem.

KiCad kändes ganska osmidigt i början men när man börjat lära sig funktionerna flyter det på bra och är ganska trevligt att arbeta i. Vi har tagit fram en PCB-layout för kretsen som skulle gå att beställa i dagsläget från någon kretskortstillverkare. Men eftersom vi inte uppnått kravet på avståndet så är det ingen vinning i att beställa kortet. Det finns istället möjligheter till framtida påbyggnad av den nuvarande layouten för att så småningom beställa ett kretskort.

Att bygga en prototyp var en del av specifikationen av projektet och större delen av tiden lades på det. Inledningsvis innebar det praktiska arbetet att lära känna microcontrollern. Så småningom när RFID-modulen och taggen var levererad låg fokuset istället på att få igång kommunikationen mellan modulen och microcontrollern. När kommunikationen väl fungerade blev nästa steg att utveckla mjukvara som skulle hantera informationen från läsaren samt få igång kommunikationen med fordonet för att kunna låsa/låsa upp. Idén att fokusera på det befintliga fjärrlåset underlättade arbetet avsevärt och i detta skede flöt arbetet på.

Nästa steg var att implementera systemet på lång räckvidd. En modul köptes in men det var inte en vanlig RF-modul, det vill säga en AD-omvandlare som skickar ut en bärvåg för att kunna få tillbaka taggens ID. Modulen som köptes in har ett applikationslagr där kommunikationen sker med så kallade EnOcean radio telegram. Arbete lades ner på detta innan vi upptäckte att

applikationsprotokollet var problemet och då var det begränsat med tid för att börja med en ny modul eller annan idé.

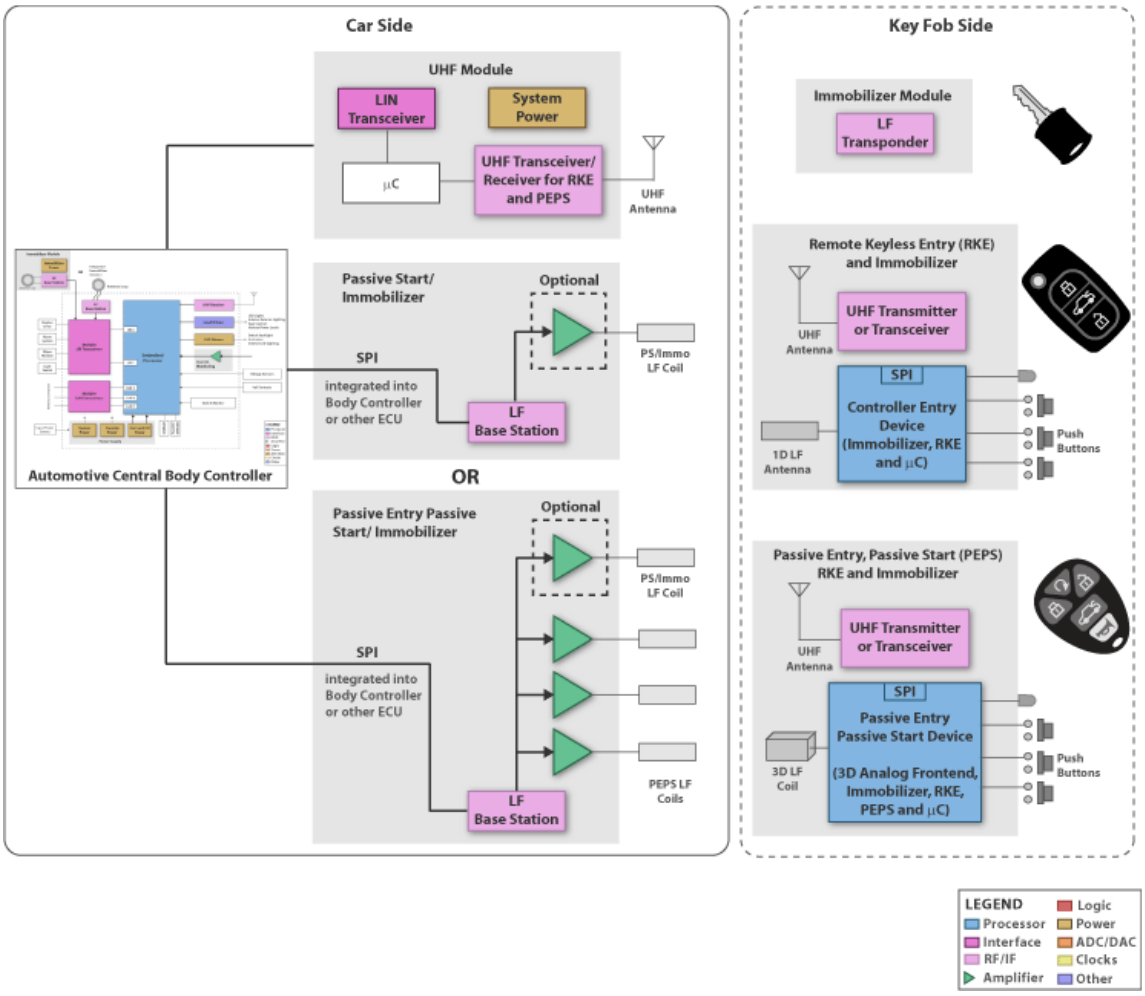
Sammanfattningsvis, att bygga en prototyp på långt avstånd tog längre tid än förväntat. Istället byggdes en funktionsprototyp som visar att ett det är möjligt att konstruera ett sådant system även om det i det här fallet endast fungerar på kort avstånd.

5.1 Fortsatt utveckling

Då det i projektet endast har tagits fram en prototyp som fungerar på kort avstånd behöver det implementeras funktionen för läsning på längre avstånd. För att åstadkomma detta finns det olika vägar att gå. Ett alternativ är att använda sig av färdiga UHF-moduler för RFID som verkar på långt avstånd. Fördelen med denna lösning är enkelheten med implementation i den redan framtagna prototypen. Nackdelen blir den stora kostnad varje enhet kommer få. Detta eftersom sådana moduler har en prislapp på 2000-3000 kronor och uppåt. Så ser det ut på marknaden i dagsläget men skulle utbudet av UFH-moduler öka och priserna sjunka så skulle denna lösning kunna användas.

För att få ned kostnaden på varje tillverkad enhet skulle utveckling av egen hårdvara istället för användning av moduler vara önskvärt. Det finns på marknaden integrerade kretsar tillverkade speciellt för både RF och RFID. För konstruktion med hjälp av dessa krävs kunskaper om RF-elektronik och det kommer bli mer tidskrävande, men även billigare.

En stor tillverkare av integrerade kretsar är Texas Instruments och de tillverkar många produkter som fungerar i ett RFID-system. Med deras produkter skulle ett RFID-system med aktiva taggar kunna utvecklas. Det finns bra dokumentation på Texas Instruments hemsida och bilden nedan visar hur ett system med deras produkter kan se ut. Nedersta systemet “passive entry” på bilden (figur 5.1) kan vara aktuellt i detta projekt. I system används bland annat RF-transceivern CC1101-Q1. Den används både i den aktiva taggen och i läsaren som sitter i bilen. De flesta RF-transceivers kostar runt 30-60 kronor vilket gör att priset minskar kraftigt jämfört med att använda en färdig UHF-modul.



Figur 5.1 Förslag på låssystem från Texas Instruments

Förslag till fortsatt utveckling är att konstruera ett aktivt system med en RF-transceiver. Ett aktivt system kan vara fördelaktigt eftersom det går att använda sig av rullande kod vilket ökar säkerheten på produkten. Med ett aktivt system blir det även lättare att tillmötesgå avståndskravet eftersom taggen kan sända ut signaler med högre effekt. Det kommer krävas att taggen är energisnål så livslängden på batteriet blir så långt som möjligt. Detta påverkas till stor del på valet av processor och RF-transceiver. Det är därför viktigt att dessa är strömsnåla. En aktiv RFID-tagg skall bara skicka ut sin signal efter att den först fått en signal från läsaren, vilket också medför att energiåtgången minskar samt att onödiga signaler tas bort.

De system som finns ute på marknaden idag använder ett liknande system med mottagare och aktiv tagg där taggen aktiveras med en signal på 125kHz för att sedan skicka sin ID på frekvensen 868MHz till en mottagare i fordonet. En idé är att köpa in ett befintligt system och analysera hur det är uppbyggt.

6. Referenser

Bengtsson, L. (2009) *Microcontrollers.* Lund: Studentlitteratur

Buchanan, W. (2000) *Computer busses design and application.* Boca Raton: CRC

Glover, B. & Bhatt H. (2006) *RFID Essentials.* Sebastopol: O'Reilly Media

Henrici, D. (2008) RFID *Security and Privacy.* Berlin: Springer Berlin Heidelberg

Microchip (2011) *Datablad Microchip 93C66C*
http://ww1.microchip.com/downloads/en/DeviceDoc/21795E.pdf
(Hämtat 2013-04-23)

RDM632 (2013) *Datasheet RDM632*
http://www.seeedstudio.com/depot/datasheet/RDM630-Spec..pdf
(Hämtat 2013-04-03)

Reynders, D., Mackey, S. & Wright, E. (2005) *Practical Industrial Data Communications.* Oxford: Butterworth-Heinemann Ltd

Seng (2013) *Installing a toolchain for Cortex-M3/STM32 on Ubuntu Version 0.8.2.*
http://www.seng.de/downloads/HowTo_ToolChain_STM32_Ubuntu.pdf
(Besökt 2013-03-20)

Sweeney, P. J. (2005) *RFID for Dummies*. Indianapolis: Wiley Publishing

TCM320 (2013) *User Manual TCM320*
http://www.enocean.com/en/enocean_modules/TCM_300__TCM_320_User_Manual_V1.31_06.pdf/
(Besökt 2013-04-29)

Wiki1 (2013) *Radio Frequency Identification.*
http://sv.wikipedia.org/wiki/Radio_Frequency_Identification
(Besökt 2013-04-15)

Wiki2 (2013) *Baud.*
http://sv.wikipedia.org/wiki/Baud
(Besökt 2013-04-09)

Wiki3 (2013) RS-232
http://sv.wikipedia.org/wiki/RS-232
(Besökt 2013-04-05)

6.1 Bildreferenser

Advancedkeys (2013)
ttp://www.advancedkeys.com/Prod_AK104B.html
(Hämtat 2013-04-19)

Advancedkeys2 (2013)
http://www.advancedkeys.com/technology_security.htm
(Hämtat 2013-05-13)

Absoluteconceptz (2013) RFID-kopiator
http://www.absoluteconceptz.com/shop/gadgets/rfid-access-card-copierduplicator-with-writable-rfid-card-and-keychain-standalone-operation/
(Hämtat 2013-04-19)

Electrokit (2013) RDM630
http://www.electrokit.com/rfid-modul-125khz-wiegand.47193
(Hämtat 2013-04-19)

Microchip (2011) *Datablad Microchip 93C66C*
http://ww1.microchip.com/downloads/en/DeviceDoc/21795E.pdf
(Hämtat 2013-04-23)

Sweeney, P. J. (2005) RFID-system som skickar data
Figur 2-1 i *RFID for Dummies*. Indianapolis: Wiley Publishing

Olimex (2013) Olimex microcontroller STM32-H103
https://www.olimex.com/Products/ARM/ST/STM32-H103/
(Hämtat 2013-04-19)

USConvereters (2013-04-19) DB-9
http://www.usconverters.com/index.php?main_page=page&id=61&chapter=0
(Hämtat 2013-04-19)

Texas Instruments (2013) Funktion med Texas Instruments produkter
http://www.ti.com/solution/car_access_system
(Hämtat 2013-05-13)

6.2 Materialreferenser

EEPROM microchip 93C66C-I/P - 5,49 kr
http://se.farnell.com/microchip/93c66c-i-p/eeprom-4kbit-microwire-8dip/dp/2098002?in_merch=New%20Products&in_merch=Utvalda%20nya%20produkter&MER=i-9b10-00002068
(Köpt, 2013-04-19)

Mottagarmodul RX433, 433,92 MHz - 79 kr
http://www.kjell.com/sortiment/el/elektronik/fjarrstyrning/433-mhz-mottagarmodul-p88900#ProductDetailedInformation
(Köpt, 2013-04-09)

RFID-läsare RDM632, 125kHz - 199 kr
http://www.electrokit.com/rfid-modul-125khz-uart.47191
(Köpt, 2013-04-03)

RF-transceiver 868MHz TCM 320 - 275,32 kr
http://se.farnell.com/enocean/tcm-320/module-soc-rftrx-w8051-mcu-868mhz/dp/2134197
(Köpt, 2013-04-19)

STM32-H103 experimentkort från Olimex - 239 kr
http://www.electrokit.com/stm32h103-utvecklingskort-cortexm3.49727
(Köpt, 2013-03-18)

Tagg passiv 125kHz, Sparkfun COM-10169 - 19 kr
http://www.electrokit.com/rfid-accesskort-125khz.45264
(Köpt, 2013-04-03)

Tagg passiv 868MHz EURUHFT4930 - 26,09 kr
http://se.farnell.com/avonwood/euruhft4930/rfid-tag-uhf-868mhz-96bit/dp/1702061
(Köpt, 2013-04-19)

Bilaga A: RS-232

RS-232 är en standard som befinner sig på lager ett i OSI-modellen, det fysiska lagret. Syftet med RS-232 var ursprungligen att koppla ihop en DTE med en DCE, vanligtvis en dataterminal med ett modem (Reynders m.fl., 2005). Idag används det istället mest för att koppla ihop externa enheter och datorer, bland annat inom tillverkningsindustrin (Wiki3). Standarden använder seriell duplex-kommunikation där det skickas en bit i taget. En kabel används för att skicka och en annan för att ta emot data. RS-232 standardiserades 1962 av EIA och har används flitigt sedan dess (Buchanan, 2000).

RS-232 använder asynkron kommunikation och har ett så kallat start/stop-dataformat. Data konverteras från parallell till asynkron start/stop-seriell form med hjälp av en krets som kallas UART. Ett tecken skickas i taget och det finns en viss fördröjning mellan dem. Fördröjningen (eller idle) är alltid en etta. När sändaren ska skicka data till mottagaren börjar den med att skicka en startbit för att informera mottagaren att tecken kommer att skickas, startbiten är alltid en nolla. Sedan skickas ett 7-bitars ASCII-tecken följt av en paritetsbit och slutligen två ettor (Buchanan, 2000).

Paritetsbiten i slutet av varje tecken används som en kontrollbit för att kontrollera att det inte har blivit något fel i sändningen. Enkelt förklarat så räknar man antal ettor i tecknet och tittar på om antalet är jämnt eller udda. Det innebär att man kan upptäcka eventuella fel men samtidigt att antalet fel behöver vara udda för att det ska upptäckas. Är antalet fel jämnt så kommer paritetsbiten att förbli oförändrad och mottagaren kommer inte att upptäcka att det har blivit fel i sändningen (Buchanan, 2000).

Att skicka data via RS-232 går relativt långsamt. Elva bitar behövs för att skicka ett tecken på sju bitar. Ett sätt att mäta hastigheten är med hjälp av baudhastigheten. Det beskriver antal signalelement, symboler, som skickas per sekund. En sådan symbol kan innehålla olika mycket information (Wiki2, 2013). Asynkron kommunikation med start/stop innebär att både sändare och mottagare måste ha samma bithastighet. Tajmingen beror på baudhastigheten och kan beskrivas med hjälp av följande formel:

Tidsperioden för varje bit i sekunder = 1/baudhastigheten

Innan dataöverföringen påbörjar används i vissa fall hårdvaruhandskakning, i andra fall mjukvaruhandskakning eller ingen handskakning alls. Hårdvaruhandskakning innebär att man använder kontrollkopplingarna för att meddela när enheterna kan skicka data och när de kan ta emot data. Detta gör man genom att skicka en etta på en viss pinne. Mjukvaruhandskakningar innebär att enheterna skickar speciella kontrolltecken till exempel DC1 (Xon) - DC4 (Xoff). I de fall då ingen handskakning används är det viktigt att mottagaren kan läsa ett tecken innan nästa skickas. Exempelvis så kan mottagaren buffra tecken i en speciell del av minnet innan de läses.

Sändaren skickar data på TD pinnen och mottagaren tar emot den på RD pinnen. Det innebär att TD pinnen på den ena enheten kopplas till RD pinnen på den andra och tvärtom (Buchanan, 2000).

Enligt Reynders m.fl. (2005) består RS-232 standarden av tre huvudsakliga delar, dessa beskriver:

- Egenskaper hos de elektriska signalerna
- Egenskaperna hos det mekaniska gränssnittet
- Funktionen, hur kretsarna kommunicerar

En RS-232 sändare skickar signaler med spänning mellan +/- 5 och +/- 25 volt

Logisk 1 : -5 till -25 V

Logisk 0: + 5 till +25 V

Odefinierad: -5 till +5 V

En RS-232 mottagare tar emot signaler med spänning mellan +/- 3 och +/- 25 volt

Logisk 1 : -3 till -25 V

Logisk 0: + 3 till +25 V

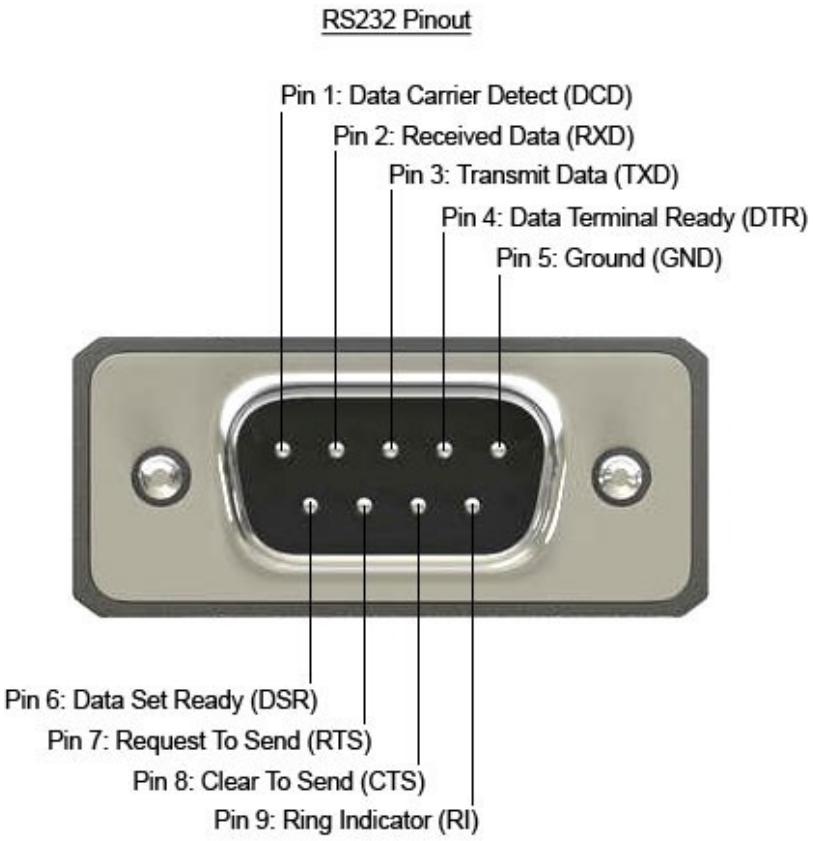
Odefinierad: -3 till +3 V

Anledningen till att sändaren skickar en lite högre spänning är för att ge bättre störmarginal med långa kablar så spänningen sjunker innan den når mottagaren. En nivåkonverterare kan behövas för att justera nivåerna om signalerna ska skickas till en microcontroller då en microcontroller använder TTL logik, det vill säga från 0 till +5 V (Reynders m.fl., 2005).

Även då signalspänningen varierar mellan +/- 3 till +/- 25 volt ligger det vanligtvis runt +/- 12 volt (Buchanan, 2000).

RS-232 definierar även de mekaniska egenskaperna i kommunikationen mellan en DTE och DCE, det vill säga vilken slags kopplingsnod som kan användas. Den vanligaste kopplingen som används är en DB-25, då används 25 kopplingar för att koppla ihop två enheter. Det är också vanligt att använda en DB-9 (Figur 3.4) där nio kopplingar kopplar samman enheterna. Det är numera standarden. Dessa kopplingar är indelade i fyra grupper: data, kontroll, tajming och speciella sekundära funktioner. Datakopplingar används för att skicka och ta emot data. Kontrollkopplingar används vid hårdvaruhandskakning. De reglerar datakommunikationen genom att till exempel begära att skicka data till den andra enheten. Tajmingkopplingar hanterar

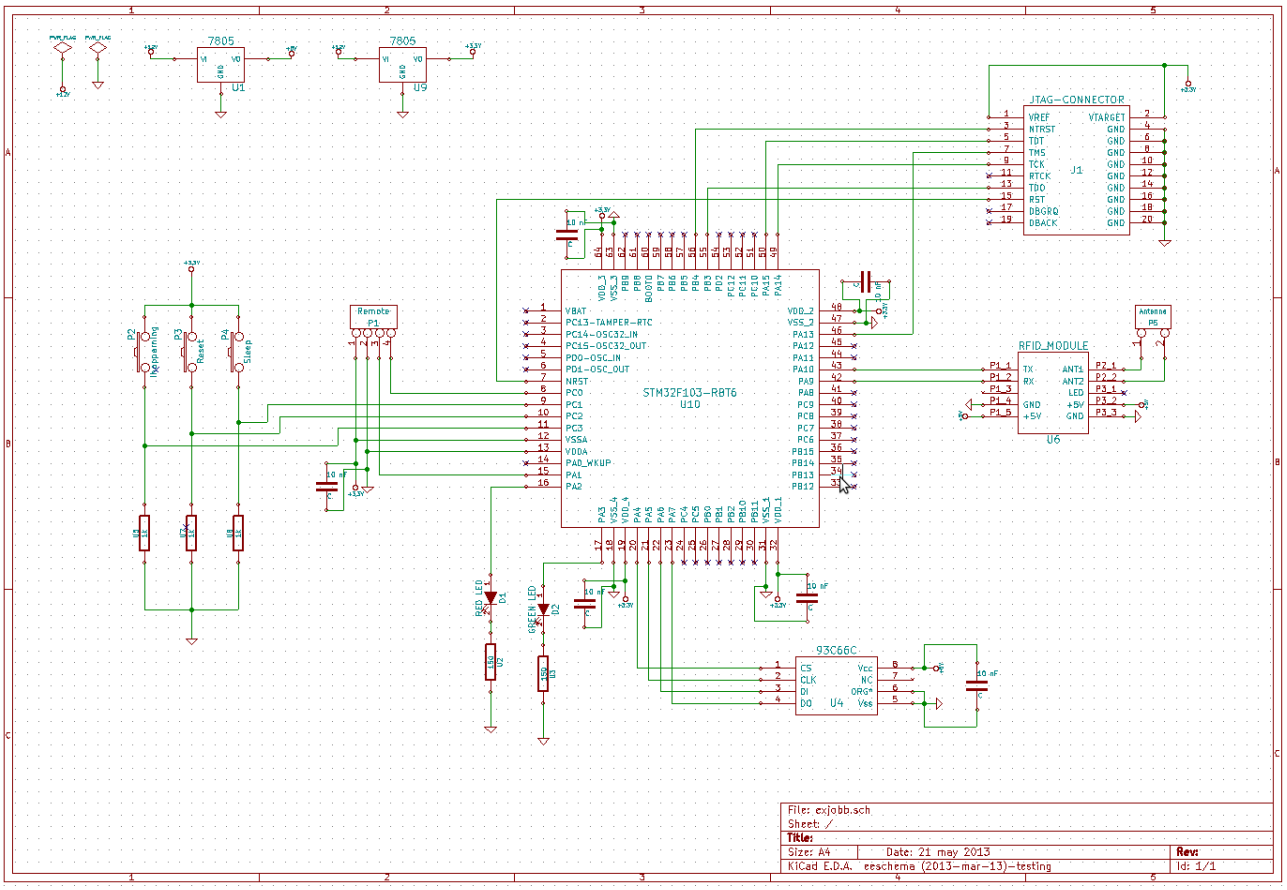
tajmingen och de sekundära kopplingarna hanterar övriga funktioner (Reynders m.fl., 2005). DB-25 erbjuder full RS-232 funktionalitet, men DB-9 används vanligtvis idag eftersom enheterna har blivit mindre och det behövs mindre kopplingsdon (Buchanan, 2000).



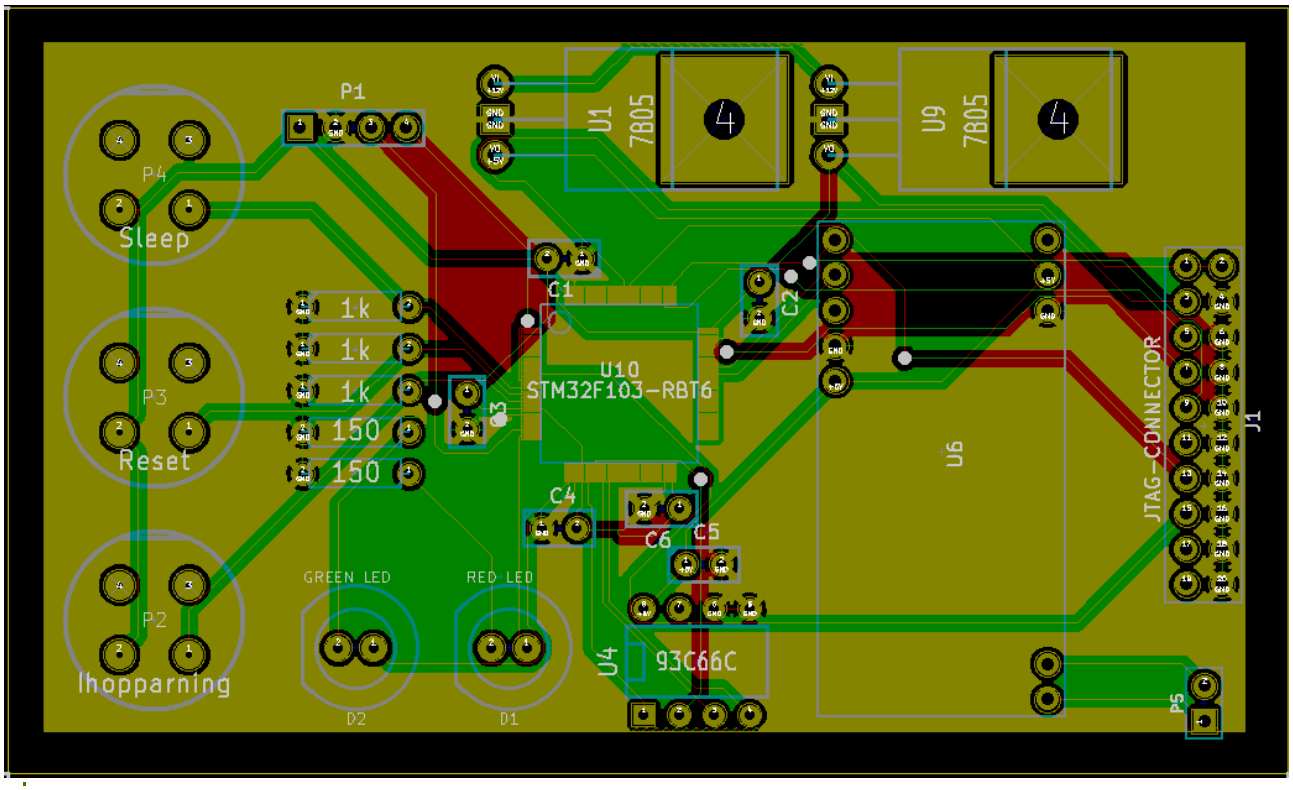
Figur A.1 DB-9 (USConverters, 2013)

Den funktionella beskrivningen säger hur dessa pinnar används i kommunikationen mellan två enheter. Till exempel så jordar man DTE och DCE så att de har samma jord. Kopplingen som skickar data används för att skicka data från DTE till DCE och pinnen som tar emot data används för att skicka data från DCE till DTE. Vidare så används en av kontrollkopplingarna när DTE begär att skicka data och en annan när DCE är redo att ta emot (Reynders m.fl., 2005).

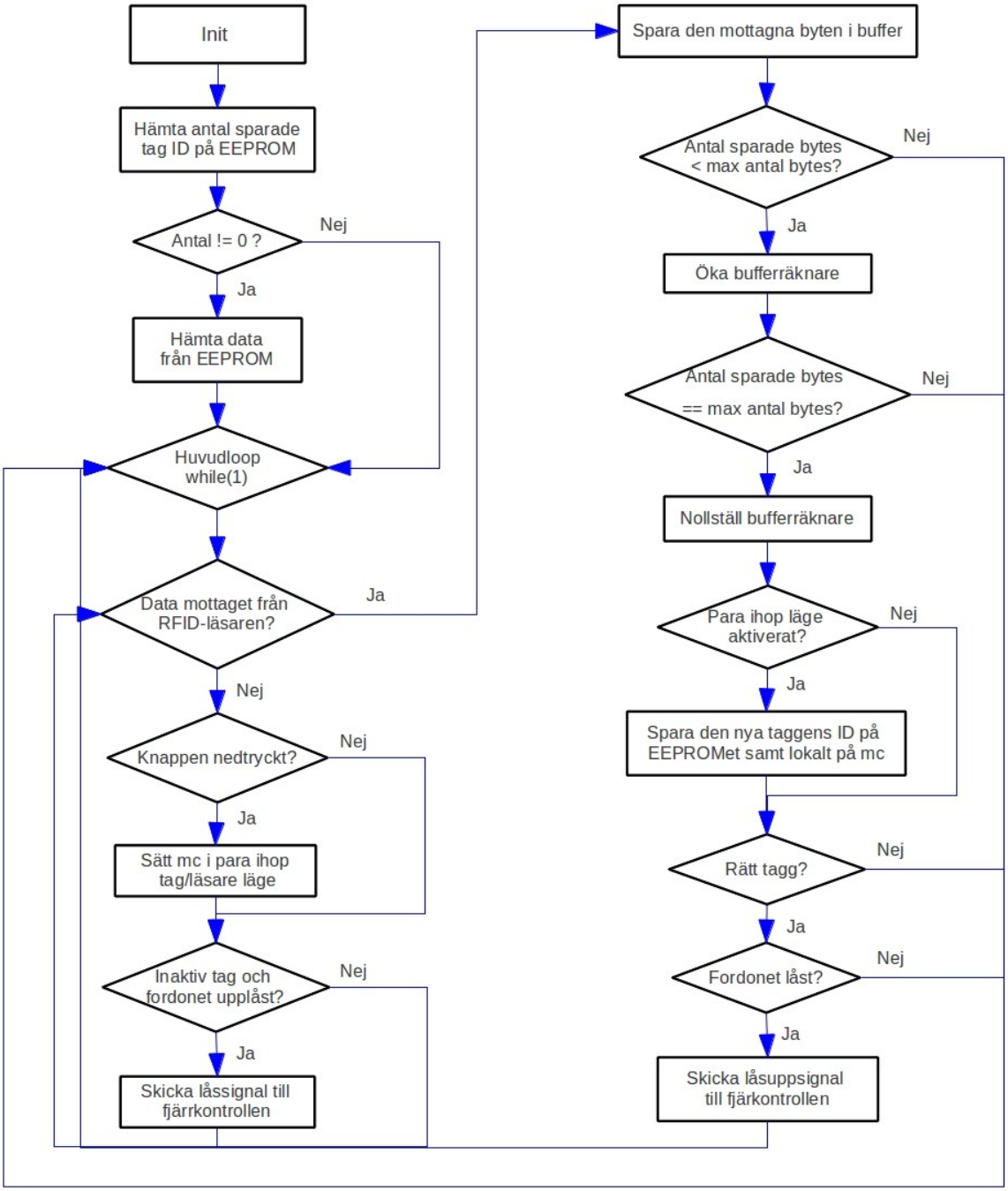
Bilaga B: Kretsschema i KiCad



Bilaga C: PCB-layout i KiCad



Bilaga D: Flödesschema



ID	Title	Preconditions	Postconditions	Description	Dependencies	Rationale
1	Tagg/Läsare	Taggen hamnar innanför läs zonen	Signal (ID) skickas till RFID-läsaren	Signal med taggens ID ska skickas till läsaren när den kommer innanför läs zonen	-	Läsare och tagg behöver kunna kommunicera på 2-3 m avstånd för att system ska fungera
2	Läsare/ microcontroller	RFID-läsaren får input från antennen	Signal skickas till microcontrollern	RFID-läsaren ska kunna skicka taggens ID till microcontrollen	Krav ID: 1	Läsare och microcontroller behöver kunna kommunicera för att system ska fungera
3	Microcontroller	Microcontrollern får input från RFID-läsaren	Microcontrollern kontrollerar om taggen är rätt	Microcontrollern ska kunna hantera och skilja mellan rätt och fel tagg	Krav ID: 1, 2	Kravet är en förutsättning för att systemet ska vara säkert och fungera
4	Microcontroller/ fjärrkontroll	Microcontrollen har kontrollerat om taggen är rätt/fel	Signal skickas till fjärrkontrollen	Microcontrollen skickar signal till lås/lås upp knappen	Krav ID: 1, 2, 3	Mc behöver kunna kommunicera med fjärrkontrollern för att kontrollera låset
5	Dual input	Microcontrollern får input från flera taggar	Om rätt tagg scannas ska fordonet låsas upp	Rätt tagg ska låsa upp fordonet även om fel tagg finns i närheten	Krav ID: 1, 2, 3, 4	Kravet behövs för att systemet ska vara tillförlitligt, rätt tag ska låsa upp fordonet
6	Fordonet ska låsas	Taggen är i läs zonen och lämnar läszonen	Dörrarna till bilen låses	Systet ska vara pålitligt, ska låsas när taggen är utanför läs zonen	Krav ID: 1, 2, 3, 4	Kravet är avgörande i ett tillförlitligt system
7	Fordonet ska låsas upp	Taggen närmar sig läsaren, kommer innanför läs zonen	Dörrarna till bilen låses upp	Bilen ska låsas upp innan chauffören når dörren, ska vara pålitligt	Krav ID: 1, 2, 3, 4	Kravet är avgörande i ett tillförlitligt system
8	Säkert systen	En annan brick hamnar innanför läs zonen	Ingen signal skickas till låset, dörren förblir låst.	En annan bricka ska inte kunna öppna upp dörren.	Krav ID: 1, 2, 3, 4	Kravet behövs för att system ska vara säkert. Ska inte gå att låsa upp med annan bricka