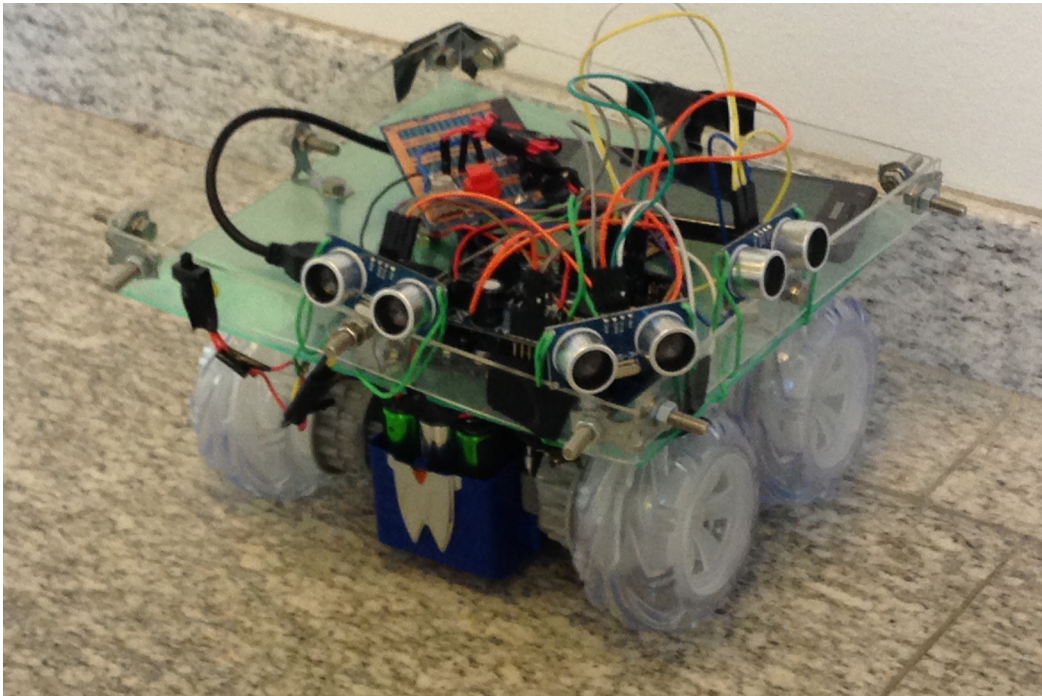


Service manual

WiFi-Seeker



Niklas Bergh
Rasmus Linusson
Araxi Mekhitarian
Sebastian Törnqvist
Max von Zedtwitz-Liebenstein

Innehållsförteckning

Kapitel 1: Översikt av robot.....	1
1.1 WiFi-Seekers syfte.....	1
1.2 WiFi-Seekers komponenter.....	1
1.3 WiFi-Seekers kodstruktur.....	1
Kapitel 2: Användning av robot.....	3
2.1 Inkoppling av komponenter.....	3
2.2 Starta och kör roboten.....	4
2.3 Läs av robotens genererade data.....	4
Kapitel 3: Specifikationer.....	6
3.1 Radiostyrd bil.....	6
3.2 Motorstyrning.....	7
3.3 Sensorer.....	7
3.4 Övrig hårdvara.....	8
3.5 Tabell över material.....	8

Kapitel 1: Översikt av robot

Kapitlets uppbyggnad:

- ✓ Beskrivning av robotens fysiska uppbyggnad
- ✓ Beskrivning av komponenter
- ✓ Beskrivning av kodens struktur

1.1 WiFi-Seekers syfte

WiFi-Seeker är en autonom robot som har till uppgift att avsöka en byggnad och kartlägga dess wifi-nätverk. Syftet är att identifiera vilka platser i byggnaden som har bäst täckning för wifi, samt vilka platser där täckningen är dålig. Denna data presenteras sedan för användaren i ett grafiskt format så att denne enkelt kan åtgärda eventuella täckningsproblem genom att flytta om accesspunkter eller ändra dess frekvens.

När WiFi-Seeker startas kommer roboten att följa väggen på dess vänstra sida med ett avstånd av cirka 20 cm. Roboten har sensorer som mäter avståndet till väggen och kan autonomt hålla ett avstånd av 20-25 cm från väggen när den kör framåt. Under givna intervall kommer WiFi-Seeker att stanna för att skanna accesspunkter. De data som roboten lagrar är en ögonblicksbild av alla accesspunkter som den kan identifiera. För varje accesspunkt lagras dess MAC-adress, nätverkets namn, signalstyrka och dess frekvens. Denna data kan sedan kommas åt genom att gå in på en webbserver som körs av roboten.

1.2 WiFi-Seekers komponenter

WiFi-Seeker är baserad på en radiostyrd bil där motorerna kopplats om från bilens styrkrets till en Arduino Uno. Roboten har två motorer som vardera driver ett hjulpar på högra respektive vänstra sidan. Koden som är inlagd på Arduinon kontrollerar motorernas hastighet individuellt med hjälp av PWM-styrning. För att roboten ska svänga, backar hjulparet på ena sidan samtidigt som den andra sidan körs framåt.

Till Arduinon är också tre stycken ultraljudsensorer kopplade. Det är dessa sensorer som ger roboten information om dess omgivning, så att den kan hålla rätt avstånd till väggen och undvika hinder. Styrsystemet läser in dessa sensorers status sekventiellt och tar därefter beslut om vilka signaler som går till motorerna. WiFi-Seeker är också utrustad med en potentiometer som kontrollerar dess hastighet. Om till exempel batterierna börjar bli dåliga kommer roboten att åka långsammare. För att kompensera för detta kan operatören öka hastigheten genom att justera potentiometern.

Till roboten hör också en smartphone (Samsung Galaxy SII). På smartphonen ligger en egenskriven app som skannar wifi-nätverken. Denna app lagrar data från skanningarna och gör den tillgänglig genom en webbserver på ett eget nätverk.

1.3 WiFi-Seekers kodstruktur

WiFi-Seeker kör kod på två typer av hårdvara, en Arduino Uno som styr roboten, och en smartphone som läser av wifi och gör denna data tillgänglig för operatören. Koden på Arduinon är uppbyggd av en huvudloop med två lägen som gör följande operationer sekventiellt:

- Läsa av knapp:
 - Hotspot-läge:
 - Bilen stannar och kopplar upp ett trådlöst nätverk vid namn "WiFi-Seeker".
 - Skanningsläge:
 - Läsa av ultraljudssensorerna en efter en och lagra deras data lokalt
 - Läsa in hastigheten från vridpotentiometern genom att parsea den analoga spänningen till ett värde mellan 0 och 255
 - Utifrån värdena på ultraljudssensorerna styrs motorerna med PWM-spänning så att roboten håller ett lämpligt avstånd från väggen och klarar av att åka runt hörn och hinder

Koden använder ett bibliotek som heter PWM.h. Biblioteket är tillgängligt över internet och används för att kunna skriva PWM spänning till Arduinons digitala utgångar. Utöver koden för Arduinon finns också kod för smartphonen i form av en Androidapplikation. Denna kod är skriven i Java och används för att skannar efter accesspunkter och göra data tillgänglig för operatören. Androidapplikationen är uppbyggd av ett antal klasser, en huvudklass och några hjälpklasser.

Huvudklassen innehåller en layout som visar ett antal olika knappar på smartphonens skärm. När robotens användare trycker på de olika knapparna så är det huvudklassens uppgift att anropa de funktioner som ska köras. Huvudsakligen finns två lägen i applikationen, skanningsläge och hotspot-läge. När telefonen är i skanningsläge kan den skanna efter accesspunkter och lagra information om dessa på minneskortet. För att kunna läsa av data som telefonen skannat har smartphonen möjlighet att hosta en webbserver som en extern klient kan ansluta sig till trådlöst. För att möjliggöra detta så har telefonen ett läge som heter hotspot-läge. I detta läge startar telefonen ett eget wifi-nätverk med namnet "WiFi-Seeker", samt hostar en webbserver på detta nätverk som användaren kan ansluta sig till. Under tiden som nätverket kör kan inte telefonen skanna efter andra accesspunkter, så när användaren vill återuppta skanningen trycker denne på en knapp i webbserverns gränssnitt.

Huvudklassen innehåller kod som möjliggör mottagning av skanningsresultat för accesspunkter. För att möjliggöra kommunikation mellan Arduinon och smartphonen finns även kod för att hantera detta. I koden är detta implementerat som en funktion som tolkar olika kommandon som skickas från Arduinon eller över smartphonens trådlösa nätverk. Om till exempel knappen som är kopplad till Arduinon trycks ner, kan appen läsa av detta via ett USB-gränssnitt och tolka denna information som sändes från Arduinon. Om ett kommando kommer från hemsidan, över smartphonens trådlösa nätverk, tolkas det på liknande sätt i koden.

Knapparna i appen är till stor del mest för debugging och kommer därmed inte användas av den genomsnittsanvändaren. Knapparna som finns i appen är:

- "Check USB arduino connection". Denna knapp används för att initialisera USB-kopplingen mellan Arduinon och smartphonen
- "Manual wifi scan". Denna knapp används för att göra en manuell wifi-skanning.
- "Save wifi data". Efter att ett antal manuella skanningar är gjorda, kan man spara data på smartphonens SD-kort genom att trycka på denna knapp.
- "Hotspot On". Denna knapp slår på hotspot-läget så att telefonen kan kommunicera med en klient över smartphonens trådlösa nätverk.
- "Hotspot Off" Denna knapp slår av hotspot-läget.

Kapitel 2: Användning av robot

Kapitlets uppbyggnad:

- ✓ Inkoppling av komponenter
- ✓ Starta och kör roboten
- ✓ Läs av robotens genererade data

2.1 Inkoppling av komponenter

Innan roboten startas måste operatören kontrollera att alla komponenter är rätt inkopplade. För att göra detta enklare så presenteras här ett kopplingsschema:

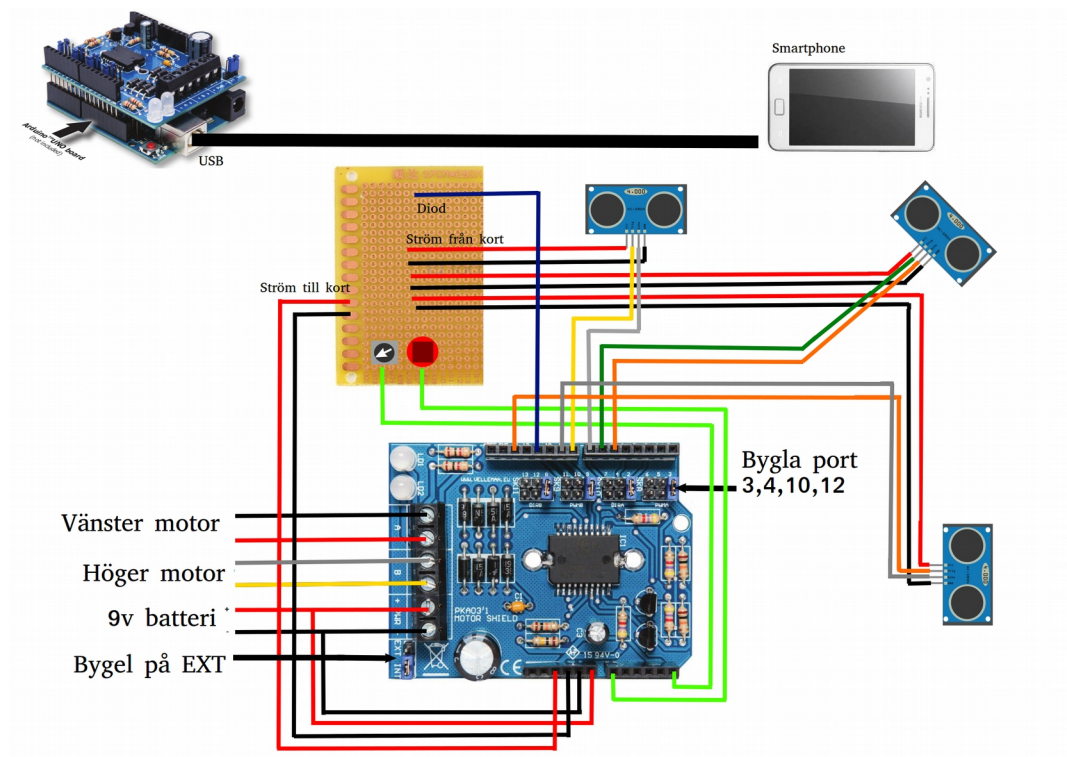


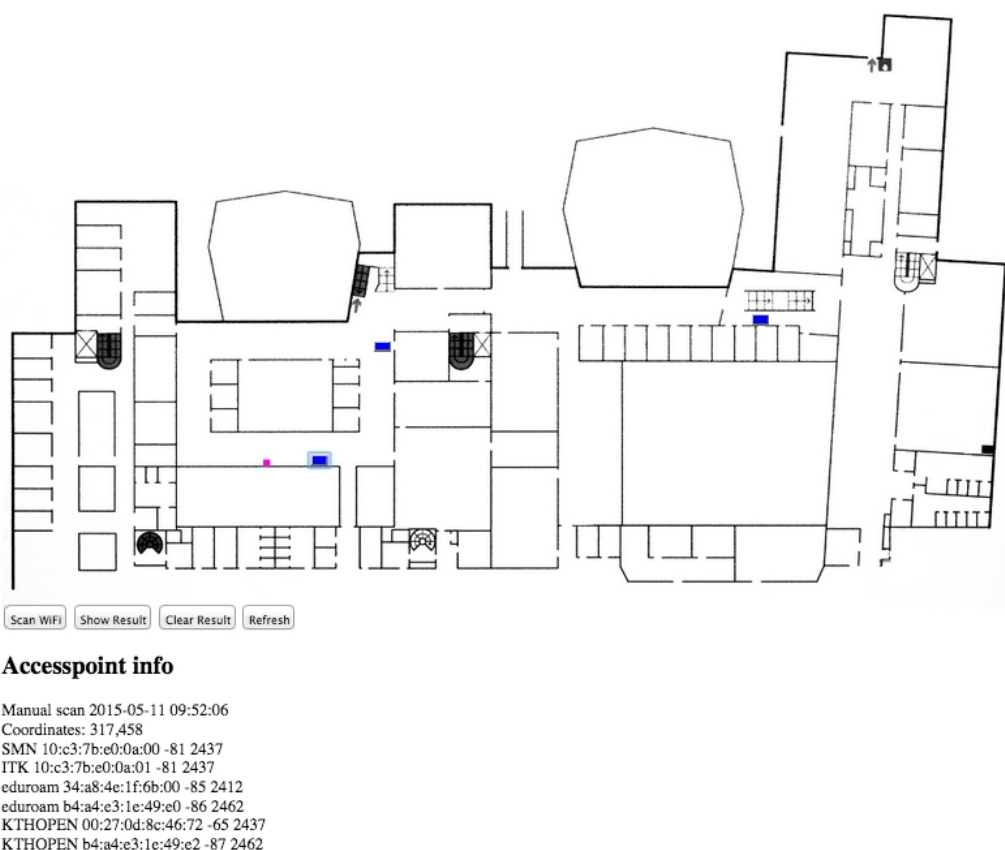
Illustration 1: Kopplingsschema robot

2.2 Starta och kör roboten

WiFi-Seeker är en autonom robot och är därför enkel att starta. Börja med att placera roboten med ultraljudssensorn på sidan mot en vägg med cirka 20 cm avstånd. Starta appen på smartphonen och klicka sedan in robotens position i webbgränssnittet. Koppla ihop smartphonen med roboten genom USB-kabeln samt tryck på "Check USB arduino connection" på displayen. Slå sedan på strömbrytaren. Roboten står nu stilla med webbservern igång. Vid nedtryckning av den röda knappen kommer roboten nu att autonomt följa väggen och köra runt i byggnaden. Om roboten träffar på öppna dörrar kommer den att köra in där och fortsätta kartläggningen av accesspunkterna för wifi. Om roboten stöter på ett hinder, till exempel en person som står längs väggen, en blomkruka eller pelare så kommer roboten svänga ut från väggen och köra runt hindret. Efter detta kommer roboten ta sig tillbaka till väggen och fortsätta kartläggningen.

2.3 Läs av robotens genererade data

Roboten kan generera data på två olika sätt. Dels kommer roboten, när den startat, att automatiskt läsa av accesspunkter för wifi och spara dessa. Roboten har också möjligheten att läsa av accesspunkter manuellt när användaren vill det. För att skanna manuellt samt för att läsa av redan skannade accesspunkter måste operatören ansluta till webbservern som körs på smartphonen.



Figur 1: Skärmdump på webbsida med mätningar

Knapparna och deras funktioner:

- Scan wifi – vid knapptryckning kommer webbservern på smartphonen att kopplas ner så att smartphonen kan skanna efter accesspunkter för wifi
- Show Result – vid knapptryckning omdirigeras användaren till en sida med rådata från mätningarna
- Clear Result – Rensar mätdata
- Refresh – Hämtar startsidan igen

För att ansluta till webbservern måste operatören först ansluta sig till telefonens trådlösa nätverk. Det nätverket heter ”WiFi-Seeker”. Efter detta måste operatören besöka ip-adressen 192.168.43.1:8080 i en webbläsare. Webbssidan visar en karta över byggnaden som roboten kör i. På kartan syns också en rosa ruta som visar robotens nuvarande position. Den kan uppdateras genom att klicka på kartan och sedan med en knapptryckning bekräfta att robotens position ska uppdateras.

När koordinaterna är visade går det att trycka på en knapp som heter ”Scan wifi”. När skanningen är klar kommer smartphonen starta upp webbservern och resultaten presenteras på kartan som blåa klickbara rutor. Genom att klicka på rutorna på kartan kommer mätdata att visas från skanningen som genomfördes på den platsen.

Kapitel 3: Specifikationer

Kapitlets uppbyggnad:

- ✓ Radiostyrd bil
- ✓ Motorstyrning
- ✓ Sensorer
- ✓ Övrig hårdvara

3.1 Radiostyrd bil

Roboten är baserad på en radiostyrd bil, ”RC snurrbil” art.nr 45-743 från Biltema. Bilen är utrustad med två motorer som vardera driver höger och vänster hjulpar. Motoraxlarna överför kraften från motorerna till båda hjulen med hjälp av kugghjul i plast. Tillhörande bilen är en fjärrkontroll med två spakar. Tillståndet hos strömbrytarna i fjärrkontrollen representerar utsignalen som den kan ge. Varje spak ger antingen maxfart framåt eller maxfart bakåt på höger eller vänster hjulpar. På bilen finns ett kretskort som har till uppgift att ta emot radiosignalerna från fjärrkontrollen och ge motsvarande utsignaler till bilens motorer. Kretskortet kan även slå på och styra LED's som sitter runtomkring på bilen.

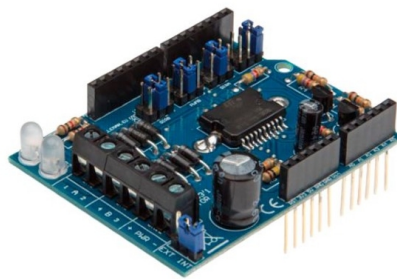


Figur 2: RC snurrbil

En stor begränsning i bilens styrsystem i sin ursprungliga form, är att det inte tillåter att motorerna kör på någon annan hastighet än maximal framåt eller bakåt. För att komma tillrätta med problemet beslutade sig gruppen därför för att försöka PWM-styra motorerna. Ursprungligen var planen att koppla in sig på fjärrkontrollens kretskort och försöka styra strömbrytarna där. Det visade sig dock att det inte gick att försvaga utsignalerna och få en motsvarande försvagning i hastighet på bilen. Detta troligen för att logiken i bilens mottagarkrets tolkar de svagare signalerna som att fjärrkontrollen befinner sig på ett större avstånd från bilen och därför fortfarande kör på maximal fart. Nästa steg i planen blev därför att koppla in sig direkt på bilens motorer. Det gjordes genom att klippa av de fyra sladdarna som går från bilens styrcykel och istället styra motorerna från en Arduino Uno.

3.2 Motorstyrning

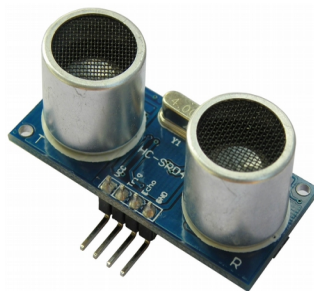
Systemet på WiFi-Seeker som styr hur roboten rör sig körs på en Arduino Uno. För att kunna kontrollera motorens hastighet var planen att koppla de fyra sladdarna från bilens två motorer direkt till Arduinons digitala utgångar. Därefter skulle kod skrivas för att skicka PWM-signaler till utgångarna. För att generera PWM-signaler så användes ett Arduino-bibliotek med namnet PWM.h som hittades på internet. Biblioteket kommer med fördefinierade funktioner som låter användaren generera en PWM-signal med värden mellan 0 och 255. Dessa representerar signalens duty-cycle. Ett problem uppstod dock med denna metod, det visade sig nämligen att motorerna drog mer ström än vad Arduinons digitala utgångar kunde leverera. Lösningen blev att använda ett kretskort, byggt för Arduino, som kunde leverera strömmen som önskades. WiFi-Seeker använder sig av "Arduino Motor & Power Shield", art.nr 87129 från Kjell & Company. Den låter roboten använda ström direkt från batterier och kan därmed ge motorerna mer kraft än vad Arduinon själv kan generera.



Figur 3: Arduino Power & Motor Shield

3.3 Sensorer

För att roboten ska kunna observera sin omgivning används tre ultraljudssensorer, HC-SR04. Sensorerna har 4 pinnar, två för strömförsörjning och två för dataöverföring. Sensorerna kopplas så att strömförsörjningspinnarna går till kretskortets +5V och jord, och de två dataöverföringspinnarna till digitala utgångar på Arduinon enligt kopplingsschemat i figur 1. För att läsa av avståndet skickas en 15 mikrosekunder lång puls till den dataöverföringspinnen som heter "trigger". Sensorn svarar med en puls på pinnen som heter "echo". Genom att mäta hur lång pulsen från sensorn är, kan man få ett värde på hur långt avstånd det är till objektet som ultraljudssensorns ljudsignal reflekterats mot. I koden som körs i Arduinon genomförs detta genom att skicka en puls på 10 mikrosekunder till trigger-pinnen. Sedan anropas den inbyggda funktionen pulseIn, vilket är en funktion som lyssnar på en pinne och returnerar längden på den första pulsen den hör.



Figur 4: Ultraljudssensorn HC-SR04

3.4 Övrig hårdvara

Utöver Arduinon, motorstyrningskortet, sensorerna och bilen finns även ett kretskort med komponenter fastlödda. Komponenterna som sitter på kortet är en knapp, en potentiometer, en LED, samt fyra strömförsörjningssladdar och fyra jordsladdar. Potentiometern är kopplad till Arduinons analoga ingång och fungerar som en hastighetsregulator. I början av loopen i Arduinons kod läses värdet på potentiometern in och omvandlas till ett värde mellan 0 och 255. Detta värde är sedan samma värde som används av PWM-funktionen när motorns hastighet ska bestämmas. Knappen som sitter på kortet används dels till att starta roboten och växla mellan hotspot-läge och skanningsläge.

3.5 Tabell över material

Produkt	Namn	Antal	Leverantör
RC bil	RC snurrbil	1	Biltema
Arduino	Arduino Uno	1	Fredrik Lundevall
Motorstyrningskort	Arduino Motor & Power Shield	1	Kjell och company
Smartphone	Samsung Galaxy SII	1	Rasmus Linusson
Ultraljudssensor	HC-SR04	3	Rasmus Linusson
Lödningskort med komponenter	-	1	Fredrik Lundevall
Batterihållare	6*AA	1	Kjell och company
Plexiglas	34-1359	1	Clas ohlson
Skravar och muttrar	-	1	Clas ohlson
OTG-adapter	-	1	Kjell och company
USB A-B adapter Ha-Ha	-	1	Dustin Home