

Vehicle IP

Ramon Carceller Valls

Enginyeria Tècnica de Telecomunicació

Especialitat en Telemàtica

Consultor: Jordi Bécares Ferrés

Data de lliurament 15 - Gener - 2013

Agraïments

M'agradaria donar les gràcies a totes aquelles persones que m'han ajudat d'una manera o altre a dur a terme aquets projecte, són moltes, i és molt difícil nomenar-los totes i cada una d'elles.

Però el que si que vull, és agrair l'ajut que he tingut per part dels diferents consultor que he tingut durant aquests anys i sobretot als diferents estudiants que a traves del fòrum hem compartit aula, on l'amistat ha arribat més enllà de l'aula d'estudi.

Voldria esmentar molt especialment al meu amic Francesc, que m'ha animat i aconsellat sempre que m'ha fet falta, sincerament moltes gràcies.

Finalment a la meva família i en especial a la meva dona M. Antonia. Sense la seva paciència infinita no hauria pogut estudiar la carrera.

Moltes gràcies a tots

Resum

El projecte que es descriu a continuació, s'emmarca dins **l'àrea dels sistemes encastats**. Tracta del desenvolupament d'un **vehicle**, que pugui esser dirigit des de qualsevol punt amb connexió a Internet.

Actualment disposem de varies tecnologies que ens permeten desenvolupar aquest projecte d'una manera més ràpida i econòmica.

Una d'aquestes tecnologies és la Wi-Fi, que ens permet establir connexions sense fils compatibles entre diferents dispositius. Amb la certificació de la WECA l'abril de l'any 2000 va sorgir la marca Wi-Fi, que garantia que tots els equips que tinguessin aquest segell poguessin treballar junts sense problemes.

Una altre tecnologia és el sistemes encastats que són sistemes de computació programables per realitzar funcions específiques. Els primers sistemes encastats van esser desenvolupats per IBM als anys 80, on una de les seves principals avantatges és el seu reduït cost. Els sistemes encastats són sistemes de processament d'informació que han estat incrustats a dins d'un producte superior, on la majoria d'aquest són transparents per l'usuari.

Així, partirem d'un microcontrolador incorporat al vehicle i el programarem perquè realitzi les nostres funcions específiques, com poden ser connectar-se a una xarxa amb l'ajuda de la tecnologia Wi-Fi.

Així doncs, utilitzant aquestes tecnologies es vol dissenyar un vehicle que a més a més de connectar-se a un punt d'accés, pugui llegir dades d'un servidor, analitzar aquestes dades i executar-les.

Les dades del servidor, seran les instruccions que donarà l'usuari per tal de dirigir el vehicle, el qual bàsicament pot anar endavant o endarrere, a la dreta o a l'esquerra, o estar parat. A més a més, l'usuari assigna la durada de cada moviment.

Índex de continguts

1.	Introducció	7
1.1.	Justificació	8
1.2.	Descripció del projecte	9
1.3.	Objectius del TFC.....	10
1.4.	Enfocament i mètode seguit.....	10
1.5.	Planificació del projecte.....	11
1.6.	Recursos emprats	13
1.7.	Productes obtinguts.....	17
1.8.	Breu descripció dels altres capítols de la memòria.....	18
2.	Antecedents.....	19
2.1.	Estat de l'art.....	19
2.2.	Estudi de mercat.....	21
3.	Descripció funcional	22
3.1.	Sistema total.....	22
3.1.1.	Diagrama de blocs de l'aplicació	23
3.2.	Recerca de xarxa	25
3.3.	Connectar Xarxa.....	25
3.4.	Llegir instruccions del servidor	25
3.5.	Executar instruccions	25
3.6.	Enviar dades.....	25
3.7.	Buscar el punt d'accés amb millor cobertura	25
3.8.	Servidor (Disseny de la interfície d'usuari)	26
3.9.	Control de motors	26
3.10.	Seguiment de l'evolució del programa	27
4.	Descripció detallada	28
4.1.	Sistema total.....	28
4.1.1.	Tasca vDemanaInstruccio.....	29
4.1.2.	Tasca vMotorDrive	29
4.2.	Recerca de xarxa "WiFLYSearchNetwork"	30

4.3.	Recerca de punt d'accés "WiFlysearchAccesspoint"	30
4.4.	Connectar-se a la xarxa "WiFlyConnectNetwork".....	31
4.5.	Llegir instruccions del servidor "ReadProperaInstruccio"	31
4.6.	Executar instruccions	32
4.6.1.	Executar instruccions per mitja de pins de sortida de la LPC.....	32
4.6.2.	Executar instruccions per mitja de la controladora de motors.....	33
4.7.	Enviar dades.....	33
4.8.	Buscar el punt d'accés amb millor cobertura	34
4.9.	Servidor (Disseny de la interfície d'usuari)	36
4.10.	Diagrama d'estats.....	37
4.11.	Esquema connexió	39
4.11.1.	Esquema de connexió directament des de la LPC.....	39
4.11.2.	Esquema de connexió per mitja de la controladora de motors.....	42
4.12.	Mostra les dades pel cutecom o PuTTY	43
5.	Nocions del sistema operatiu en temps real FreeRTOS.....	45
6.	Viabilitat tècnica.....	46
7.	Valoració econòmica	46
8.	Conclusions	48
8.1.	Conclusions	48
8.2.	Proposta de millores.....	48
8.3.	Autoavaluació	49
8.4.	Dificultats trobades	49
9.	Glossari	51
10.	Bibliografia	52
11.	Annexos.....	53
11.1.	Manual d'usuari	53
11.1.1.	Instal·lació, compilació i execució del codi de la LPC	53
11.1.2.	Instal·lació, compilació i execució del codi del Servidor	54
11.2.	Imatges del vehicle.....	55

· Índex de figures.

Figura 1. Esquema de comunicació 8

Figura 2. Planificació inicial 11

Figura 3. Planificació final 12

Figura 4. LPCxpresso LPC 1769..... 13

Figura 5. WiFly RN-XV 13

Figura 6. CP 21002 14

Figura 7. Serial Controlled Motor Driver..... 14

Figura 8. Relé de 6v 15

Figura 9. Optoacobladors CNY 74 – 2 15

Figura 10. Transistor BD 135 16

Figura 11. Resistències 330 Ω 16

Figura 12. Díode 1N4007 16

Figura 13. Porta-piles 17

Figura 14. Motor elèctric..... 17

Figura 15. Esquema de direcció..... 23

Figura 16. Diagrama de blocs de l'aplicació..... 24

Figura 17. Interfície d'usuari..... 26

Figura 18. Moviment de recerca del millor punt d'accés..... 36

Figura 19. Diagrama d'estats 38

Figura 20. Esquema de connexió per pins..... 41

Figura 21. Esquema de connexió per controladora 42

Figura 22. Captura de PuTTY 44

Figura 23. Pantalla de treball de LPCxpresso 53

Figura 24. Xassís vehicle 55

Figura 25. Muntatge control motors 55

Figura 26. Vehicle IP 56

. Índex de Taules.

Taula 1. Taula de veritat.....	40
Taula 2. Despeses de material.....	47
Taula 3. Despeses de mà d'obra	47
Taula 4. Despesa total	47

1. Introducció

El propòsit d'aquest projecte és la de dotar a un petit cotxe elèctric de la capacitat de llegir instruccions en un servidor, que prèviament ha introduït l'usuari, per tal de que aquest usuari el pugui dirigir via internet.

Les instruccions que donarà l'usuari per tal de dirigir el vehicle, seran bàsicament, anar endavant o endarrere, a la dreta o a l'esquerra, o està parat. A més a més, l'usuari assigna la durada de cada moviment. Els primers paràmetres ens donaran el sentit del vehicle, però degut a la lenta reacció que poden provocar les transmissions de dades via internet, s'ha cregut convenient que qualsevol instrucció estigui limitada en el temps.

Utilitzarem bàsicament els avantatges que ens dona el sistema encastats i de la tecnologia Wi-Fi. Així, el vehicle disposarà d'un microcontrolador, el LPC1769 on introduïrem el nostre codi, un dispositiu Wi-Fi, el WiFly que ens permetrà connectar a un punt d'accés disponible, dos motors, i una font d'alimentació.

La part principal desenvolupada en aquest projecte és el codi del microcontrolador, on les seves funcions són llegir les instruccions del servidor, analitzar les ordres, executar-les, i enviar dades de control.

A més a més, per tal que l'usuari pugui introduir les instruccions desitjades, s'ha dissenyat una pàgina en HTML dins d'un servidor Tomcat. La pàgina està pensada de tal manera, que l'usuari sols a de clicar en un dels botons que indica el moviment que farà el vehicle.

Un cop el procés de desenvolupament ha acabat, s'ha conjuntat en un vehicle que disposava de dos motors. Un motor per a anar endavant o endarrere i l'altre motor per a controlar la direcció.

Un dels punts crítics, serà si el vehicle s'allunya molt del punt d'accés i perdés la connexió. Per intentar evitar-ho, en cada moviment del vehicle, controla els certs paràmetres, que en el cas que siguin negatius, el vehicle busca un punt on el senyal és òptim.

Opcionalment, l'usuari pot demanar al vehicle que busqui aquest punt òptim, independent de la posició en que es trobi.

A més a més, el programa envia al servidor, certs paràmetres de control per tal que l'usuari estigui informat.

En la següent figura, podem observar un esquema on es pot veure pel punts on viatge la informació. També dona una idea global del projecte desenvolupat.

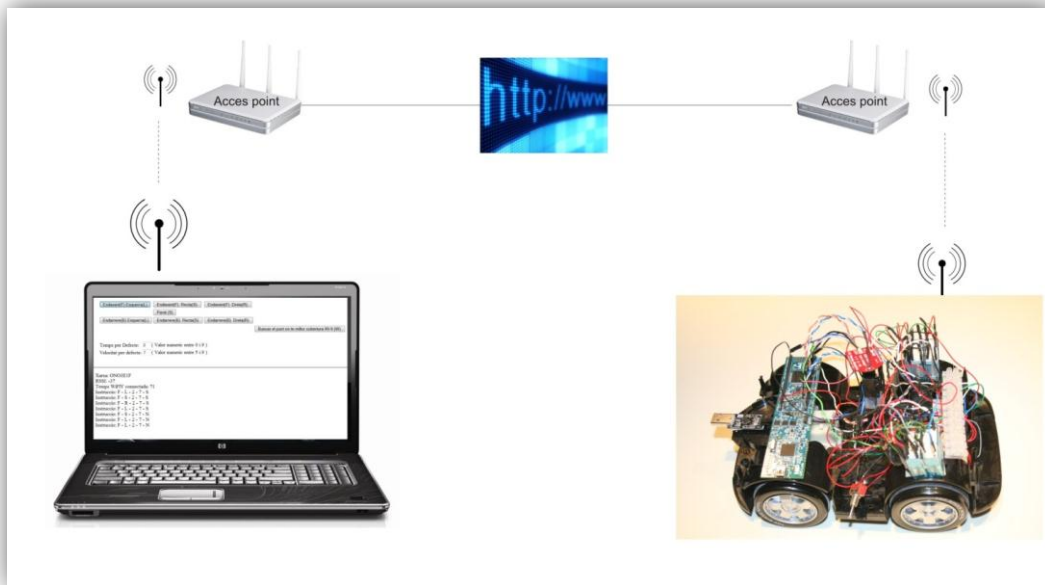


Figura 1. Esquema de comunicació

1.1. Justificació

Actualment es disposa de tecnologia suficient com per a desenvolupar les necessitats del dia a dia. Segurament es pot trobar un producte similar al aconseguit en aquest projecte, però dissenyar un prototip de vehicle governat des d'un servidor, s'ha volgut treballar des del repte personal en aprofundir i posar en pràctica els coneixements adquirits durant la carrera.

S'ha fet un projecte pràctic on s'obté un producte funcional, on el seu àmbit d'aplicació pot ser molt extens. Avui en dia, on per exemple, la majoria de mòbils tenen accés a internet, o la mobilitat que suposa els portàtils, permeten controlar a distància molts enginys que ens poden facilitar o donar seguretat en moltes tasques quotidianes.

De les diferents opcions de sistemes de comunicació s'ha optat per WiFi, que ens permet connectar el nostre prototip a qualsevol punt d'accés, afavorint el control remot des de qualsevol lloc.

S'han descartat altres tecnologies com pot ser Bluetooth, pensades en reemplaçar connexions per cable USB amb transferència de dades de poc volum, o infraroig que necessita visió directe a poca distància per establir connexió

Per controlar tot el sistema s'ha optat per un sistema encastat que són sistemes de computació programables per realitzar funcions dedicades, és a dir, de propòsit específic, a diferència dels

ordenadors que son de propòsit general. D'aquesta manera, amb un microcontrolador de mides i preus molt reduïts, introduir un programa auto executable, capaç de controlar el nostre sistema.

Amb una possible ampliació de les funcions d'aquest vehicle, si s'afegís una càmera de vídeo, les possibilitat d'aplicació augmentarien extraordinàriament deixant en mans de la imaginació de cadascú els límits del producte.

1.2. Descripció del projecte

Disposem d'una mota (microcontrolador LPC 1769) i un dispositiu de connexió Wi-Fi i el projecte consisteix en afegir un petit motor i rodes a la mota, la qual pugui anar endavant i endarrere on la direcció no es contempla com a requisit bàsic.

El sistema ha de ser capaç de rebre dades via Internet (a través del WiFi) per anar endavant i endarrere. També ha de disposar d'un sistema automàtic, el qual buscarà el punt on te millor cobertura WiFi (cerca de l'*access point*). En la figura 1, es pot veure un esquema del concepte d'aquest projecte.

Però el projecte final vol anar una mica més enllà dels requisits basics, introduint el control de la direcció, analitzar les xarxes disponibles i connectar-se a la de millor senyal, i finalment informar a l'usuari de diverses dades de control.

Per assolir aquests requisits, partirem d'un petit cotxe elèctric que disposa de dos motors, un motor per activar les rodes de tracció i un altre motor per controlar la direcció. A sobre d'aquest vehicle col·locarem la mota i els diferents components necessaris per controlar els motors i tenir connexió Wi-Fi.

La part principal del projecte serà l'elaboració del codi que s'introduirà en la mota per tal d'assolir els requisits mínims. L'entorn de programació serà LPCXPRESSO, i s'implementarà sobre un sistema operatiu en temps real com és el FreeRTOS . en aquest entorn és on desenvoluparan totes les funcions necessàries per tal finalitzar amb èxit l'execució del projecte.

A més, per tal de que l'usuari tingui un entorn amigable per introduir les instruccions al vehicle, es dissenyarà una pàgina en HTML que es pot penjar en un servidor Tomcat. D'aquesta manera l'usuari, si disposa de connexió a internet, podrà introduir les dades des de qualsevol lloc.

1.3. Objectius del TFC

Al tractar-se d'un projecte de final de carrera, es poden diferenciar dos tipus d'objectius. Un de caràcter més general on predomina la presa de contacte en l'àrea del sistemes encastats i tot el seu entorn. I l'altre objectiu més concret i el principal és la consecució del projecte en si.

Els objectius generals d'aquest projecte són:

- Introducció als sistemes encastats
- Entendre l'entorn de programació LPCXPresso
- Conèixer el funcionament dels sistemes interns de FreeRTOS
- Funcionament de la LPC 1769 amb un sistema operatiu en temps real
- Entendre el funcionament del mòdul WiFly

Els objectius específics d'aquest projecte són:

- Oferir una interfície via web on es permeti controlar el sistema
- Dotar al sistema d'una comunicació sense fils WiFi
- Dotar al sistema de la capacitat autònoma de connexió a la xarxa amb més cobertura
- Lectura periòdica de dades del servidor
- Creació de la plataforma del vehicle motoritzat
- Control del moviment del vehicle
- Dotar al sistema de la capacitat d'enviar dades al servidor

1.4. Enfocament i mètode seguit

S'ha desenvolupat el projecte seguint unes fases dissenyades.

Inicialment, en una **fase preliminar** a l'inici del projecte, s'ha treballat familiaritzant-se en el sistema operatiu i desenvolupant eines útils per al posterior desenvolupament del projecte. D'aquesta manera es podia tenir una idea més clara de les possibilitats i limitacions alhora de implementar les diferents funcions.

En la fase 0, inici pròpiament del projecte, s'ha fet la proposta i planificació. En aquesta fase, un cop escollit la proposta de control de motors, i saben els objectius a assolir, s'estudia una planificació orientativa per tal de complir els terminis establerts, de les diferents tasques que es desenvoluparan en aquest projecte, on es marcava la data límit de les diferents pacs com a referències dels terminis d'entrega

En la fase 1, fase més complexa, s'ha desenvolupat la part de software del projecte tant de les diferents tasques que executarà la LPC com de la creació del servidor i el seu entorn. En

aquesta fase s’ha donat prioritat en assolir els requisits bàsics. Un cop aconseguits, s’han implementat els objectius de millora del producte final.

En la fase 2, on es treballa més la part de hardware i muntatge final de l’estructura del vehicle. Un cop aconseguït el software necessari, en la fase anterior, en aquesta fase es fa el muntatge dels diferents components sobre l’estructura del vehicle, realitzant el test final del producte.

En l’ultima fase, la 3, però no menys important, es treballa en la memòria i presentació final del projecte. Aquesta fase s’inicia paral·lelament a les fases 1 i 2, prenent les anotacions necessàries per tal d’elaborar la memòria del projecte i el seu contingut.

1.5. Planificació del projecte

La planificació d’aquest projecte comença en la fase 0 comentada anteriorment, ja que la fase preliminar, ha estat una fase d’aprenentatge i consolidació de nous coneixements.

La planificació inicial que es proposa en la fase 0 es pot observar en la Figura 2. Planificació inicial. L’entrega de les diferents Pacs ens marcava la planificació a seguir.

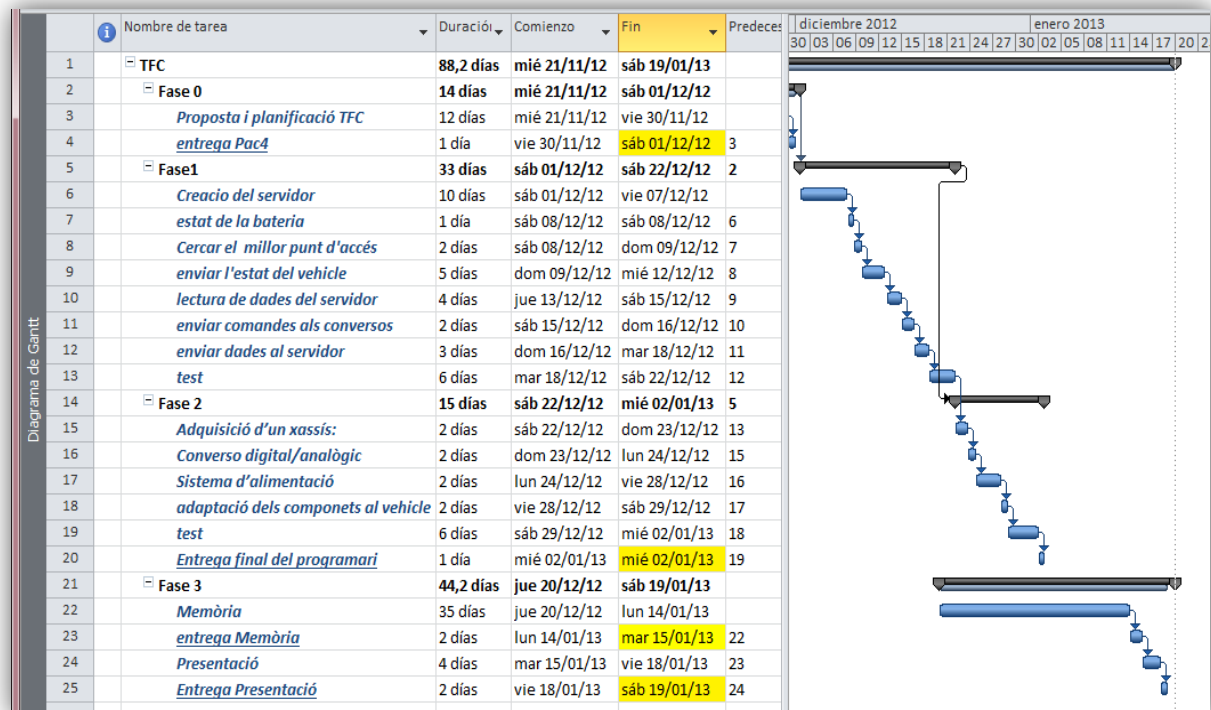


Figura 2. Planificació inicial

Aquesta planificació ha anat variant com ha conseqüència de varies causes. Primerament el consultor d'aquest projecte va fer uns suggeriments de millora que s'han intentat seguir, bàsicament en l'ordre de prioritats. També una sorpresa agradable va ser la d'una ràpida creació del servidor i el seu entorn. Això permetia compensar les dificultats que es va trobar en altres tasques del projecte, com va ser la comunicació entre la LPC i els motors, degut a una dificultat inesperada com va ser el deixar de funcionar una sortida de la placa LPC que controlava la controladora del motor. Això va provocar refer part del codi per controlar els motors amb optoacobladors i relés.

La proposta final es pot veure en Figura 3. Planificació final.

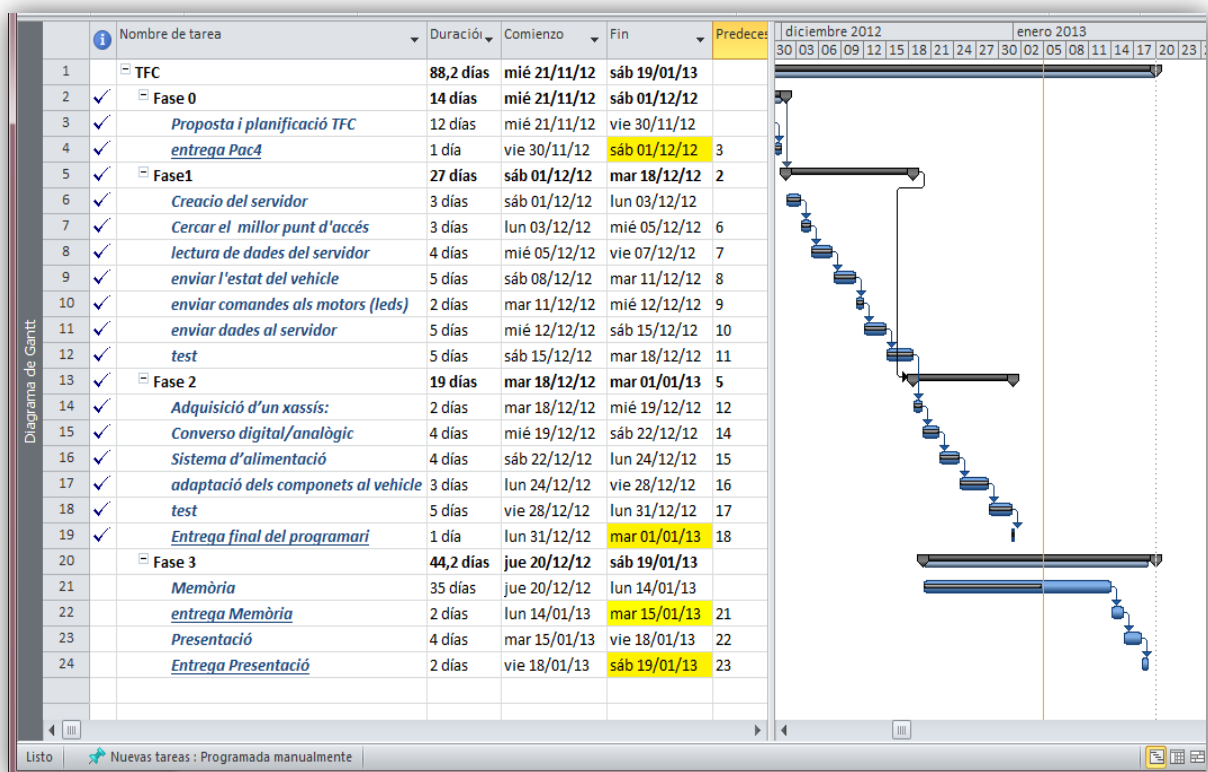


Figura 3. Planificació final

1.6. Recursos emprats

Els recursos utilitzats per a la realització d'aquets projecte són:

Hardware:

- LPCxpresso LPC 1769: microcontrolador que disposa entre altres de 64kb SRAM, 512 kb, Flash 4xUART, 3xI2C, SPI, 2xSSP, 2xCAN, PWM o USB 2.0.

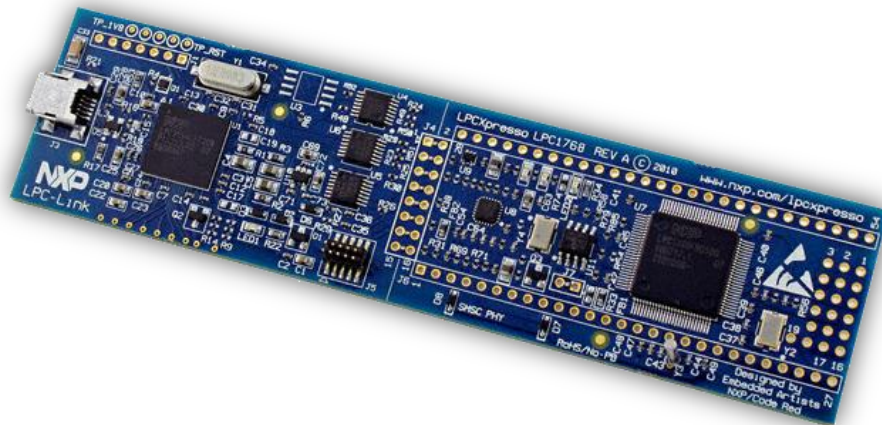


Figura 4. LPCxpresso LPC 1769

- WiFly RN-XV: és un mòdul Wi-Fi que incorpora un processador de 32 bits, 802.11b/g. Accepta un subministrament de 3.3 vdc, es pot configurar la potència de transmissió entre 12 dBm a 0dBm, disposa de 8 I/O digital de propòsit general, on per crear la connexió de dades requerim quatre PWR, TX, RX, i GND.



Figura 5. WiFly RN-XV

- CP 2102: es tracta d'un mòdul convertidor de sèrie USB 2.0 a TTL UART de 6 pins, 3,3 v RST, TXD, RXD, GND i 5v amb un buffer de transmissió de 640 byte



Figura 6. CP 21002

- Pc : Ordinador personal amb una placa base ASSUS 1366 P6T SE DDR3, amb un processador INTEL I7 2600K, 8GB RAM, sistema operatiu Windows 7
- Serial Controlled Motor Driver ROB-09571.: Permet el control de dos motors mitjançant el port 0 de la LPC. Permet controlar la velocitat dels motors, control d'avanç i retrocés. La velocitat de transmissió s'estableix a 1152000 bps (8-N-1)

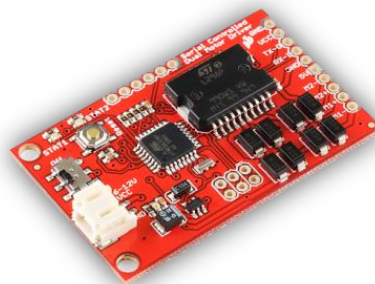


Figura 7. Serial Controlled Motor Driver

Software:

- Entorn LPCxpresso, on es disposa d'eines, aplicacions i llibreries de programació completes per a l'evolució i desenvolupament de microcontroladors de NXP.
- FreeRTOS, és un sistema operatiu per a dispositius de sistemes encastats en temps real que pot suportar sub-processos, tasques, exclusions mútues o semàfors entre d'altres
- Apache-tomcat 6.0.36. és un servidor web amb suport de servlets i JSPs. Funciona en qualsevol sistema operatiu que disposi de màquina virtual Java
- Eclipse IDE for java EE Developers, és un entorn de desenvolupament integrat de codi obert multi-plataforma, utilitzat en aquest projecte pel desenvolupament de la pàgina web destinat a l'usuari

- Java Runtime Environment (JRE) 6, és un conjunt d'utilitats que permet l'execució de programes Java, necessari pel servidor Tomcat
- Sistema operatiu del Pc Windows 7. Sistema operatiu que utilitza el ordinador personal on s'ha desenvolupat tot el projecte,
- PuTTY, (Port unique terminal type) és un client SSH, Telnet, rlogin i TCP raw amb llicència lliure.

Vehicle:

- Relés 6V, és un dispositiu electromecànic que funciona com un interruptor, consta de dos circuits, un circuit electromagnètic, amb un electroimant i un altre circuit on te diversos contactes, el qual aplicarem en el nostre circuit per controlar o be la polaritat del motor o be la seva activitat. Te l'avantatge de poder circular tensions diferents en cadascun dels circuits.



Figura 8. Relé de 6v

- optoacobladors CNY 74 – 2: els optoacobladors són components electrònics que aïllen elèctricament els circuits als qual estan connectats. Consten d'un encapsulat dintre del qual hi ha un RED (díode LED d'infraroig) i un component receptor de llum com un fotodíode o un fototransistor. En el nostre circuit de control de motors, aïllem el circuit de la LPC amb 3.3v del circuit motors de 6v, evitant posar en perill la LPC

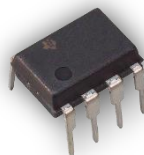


Figura 9. Optoacobladors CNY 74 – 2

- transistors BD 135: és un dispositiu electrònic semiconductor d'estat sòlid que compleix funcions d'amplificador, oscil·lador, commutació o rectificació. Consta de tres terminals

el col·lector la base i l'emissor. En els circuits digitals, s'utilitza com un interruptor elèctric ràpid, i en circuits analògics s'utilitza com a amplificador

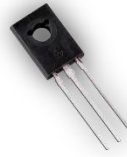


Figura 10. Transistor BD 135

- Resistències: la resistència elèctrica d'un component es pot definir com l'oposició que troba el corrent elèctric durant el seu recorregut. El seu valor es mesura en ohms i es designa amb la lletra grega omega majúscula Ω .



Figura 11. Resistències 330 Ω

- Díodes 1N4007: Un díode és un component electrònic de dos terminals que permet la circulació del corrent elèctric a través d'ell en un sol sentit. Ens serveix en el nostre circuit com a protecció davant una eventualitat canvi de polaritat el que podria comportar una averia en el transistor.



Figura 12. Díode 1N4007

- Sistema d'alimentació: degut a tenir tres circuits amb tensions diferents, s'utilitzarà tres porta-piles amb 3, 4 i 6 piles, que ens donaran 4,5v, 6v i 9v respectivament. La primera font d'alimentació ens servirà per alimentar el microcontrolador LPC, que la LPC, al mateix temps, alimenta amb 3.3v la WiFly. La segona font d'alimentació servirà per alimentar els relés de 6v. I la tercera font alimentarà als motors.



Figura 13. Porta-piles

- 2 motors: utilitzarem dos motors elèctrics de corrent continua. Un per controlar la direcció del vehicle, és a dir les dues rodes del davant. L'altre motors actuarà sobre les rodes posteriors del vehicle per avançar-lo o retrocedir-lo, depenent de la polaritat que li arribi.

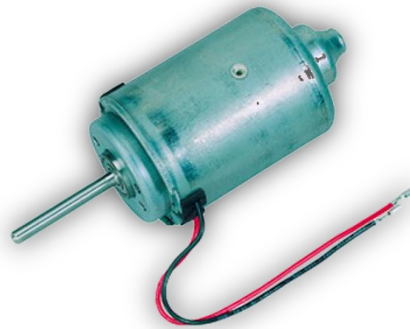


Figura 14. Motor elèctric

1.7. Productes obtinguts

Del desenvolupament d'aquest projecte, s'han obtingut 3 productes:

- Aplicació principal compostat per la LPC, la WiFly i la CP2102
És el que porta el pes del producte, on les funcions principals són:
 1. Llegir ordres del servidor
 2. Analitzar ordres i executar-les
 3. Enviar dades al servidor
 4. Enviar els missatges d'evolució del programa a la CP2102
- Vehicle.
Producte físic final que portarà incorporat tot el material necessari pel seu funcionament. Estarà dirigit per l'usuari, donant ordres en una pàgina d'un servidor.

- Centre de control en el servidor.
Pàgina on l'usuari podrà omplir un llistat de les ordres que executarà la LPC.
A més a més rebrà les dades de control de la LPC

1.8. Breu descripció dels altres capítols de la memòria

Primerament en antecedents podem copsar quin és l'estat de la tecnologia que es pot trobar actualment, una explicació dels sistemes encastats, com els podem classificar i els possibles protocols de comunicació.

A partir del 3 capítol, entrem ja de ple en el projecte en sí, explicant el disseny del sistema, el seu funcionament, donant unes primeres explicacions generals de les diferents funcions desenvolupades.

Aquestes explicacions, s'aprofundeixen en el capítol 4 , on es donant els detalls tècnics necessaris per comprendre el projecte. S'incorporen esquemes i figures per tal d'aclarir conceptes i idees.

En el capítol 5, s'intentarà donar idees sobre la viabilitat d'aquest sistema, quin són els punts forts i dèbils. Ja dins el capítol de valoració econòmica, es quantifica el projecte tenint en compte les hores invertides pel desenvolupament del projecte i també el material emprat. En aquest punt s'ha de tenir present que és un projecte de final de carrera, on l'aprenentatge té un paper molt important, però com ja es comentarà, l'elevat número d'hores invertides, eleva molt la valoració econòmica.

Finalment en els últims capítols, es fa una valoració del treball realitzat, on s'ha afegit unes propostes de millores per a que el producte obtingut pugui ser més atractiu. igualment s'inclou uns annexos per l'ajut de d'instal·lació dels diferents programes utilitzats.

2. Antecedents

2.1. Estat de l'art

Si mirem al nostre voltant, ens trobarem envoltats per sistemes informàtics. Cada any es construeixen milions de sistemes informàtics destinats a ordinadors personals o estacions de treball, però sorprenentment, es construeixen cada any milers de milions de sistemes informàtics incrustats dins grans aparells electrònics.

Podem trobar sistemes encastats en telèfons mòbils, càmeres digitals de foto o vídeo, calculadores o a més gran escala en automòbils, equips de telecomunicació, en electrònica aeronàutica, sistemes mèdics, és a dir, hi ha un ventall enorme d'aplicacions on els sistemes encastats que en la majoria de casos passen desapercibuts per la majoria de persones.

Una definició breu de sistemes encastats podria ser que tots els sistemes informàtics que no siguin un ordinador de propòsit general (amb monitor, teclat, etc.) són sistemes encastats. És una forma de treballar, organitzar o dur a terme una o diverses tasques d'acord amb un conjunt fix de regles, programa o pla. Un sistema encastat és un sistema que té un programari encastat en el maquinari, que fa que sigui un sistema dedicat per a una aplicació o part específica d'una aplicació o producte o part d'un sistema major.

Una definició general d'ús de sistemes encastats és que són dispositius utilitzats per controlar, supervisar o ajudar en l'operació dels equips, aparells i dispositius. "Incrustat" reflecteix el fet que són una part integral del sistema. En molts casos, la seva "incrustació" pot ser tal que la seva presència està lluny de ser evident per a l'observador casual.

Es podrien classificar en dos grans grups:

- Sistemes encastats en temps real. Real Time Operating System (RTOS)
Són sistemes operatius que garanteixen les respostes a cada esdeveniment dins d'un període de temps definit. Aquest tipus de sistemes s'utilitzen principalment en aplicacions on sigui crític el temps, com pot ser en mesuraments o sistemes de control. un exemple el podem trobar en el sector de l'automoció per controlar l'airbag o ABS.
El cor d'un sistema operatiu en temps és el nucli. Un nucli és el cor central del sistema operatiu, i s'encarrega de totes les tasques del sistema operatiu, com pot ser l'arrencada, la programació de tasques o les Biblioteques de funcions estàndard.
- Sistemes encastats sense temps real. Non Real Time Operating System
A diferència de l'anterior, en aquest cas no es garanteix el temps de resposta. Una de les aplicacions més habituals d'aquest tipus el podem trobar en el GPS o en routers. Utilitzen un sistema operatiu més ampli que els anteriors.

Els protocols més comuns utilitzats en els sistemes encastats són:

- I²C, és un bus de comunicació en sèrie molt utilitzat per microcontroladors
- RS-485, permet connectar entre si més de dos dispositius
- Bluetooth, especificació industrial per xarxes sense fil d'àrea personal (WPAN)
- Infraroig, permet la comunicació bidireccional entre dos extrems a poca distancia i visibles entre ells.
- ZigBee, protocol d'alt nivell de comunicació sense fil per radiodifusió digital de baix consum
- RS-232, més conegut com el port sèrie d'un PC
- USB, estàndard de connexió de perifèrics
- Transmissió de dades IP sobre connexions de sèrie(RS-232)
- Xarxa Tiny Embedded, transport de grans missatges per SAR

Les plataformes de desenvolupament que podem trobar en el mercaten quant a plaques amb microcontrolador, pins d'entrada/sortida, sortides de PWM, temporitzadors o entrades analògiques són:

- Atmel AVR: Toolchain lliure y gratuït on un dels principals fabricants i desenvolupadors de llibreries és Arduino
- Microchip PIC: trobem el projecte Pingüino on desenvolupen llibreries però que no estan tan desenvolupades i provades com Arduino. També és troba chipKIT encara que l'entorn de desenvolupament és de Arduino
- Plataforma ARM: on trobem l'arquitectura Cortex-M amb microcontroladors de baix consum, molt potents i preus molt competitius. En aquest grup podem trobar ST Micro o NXP que és la Antiga divisió d'electrònica de Philips.
- Plataforma Texas Instruments MSP430: on trobem la família Launchpad, encara que acostumant a tenir pocs mòdul de UART

2.2. Estudi de mercat

El prototip de vehicle dirigit a distancia via internet desenvolupat en aquest projecte segueix unes conductes molt especificques cosa que fa difícil trobar aplicacions similar en el mercat. De tota manera es troben aparells robotitzats on incorporen altres característiques com sensors, capturadores d'imatge, etc.

Molts d'aquest prototips s'han estat desenvolupant gràcies a la constant evolució de les tecnologies descrites anteriorment. Recordem que, per exemple tecnologies com WiFi o disposar de connexió a internet que ara són d'us quotidià, fa no massa anys, estava a l'abast de molt pocs.

El fet de poder moure un vehicle des d'una connexió a internet permet que els sectors on pot anar dirigit el sistema desenvolupat sigui molt ampli. Una aplicació podria entrar en el camp de la seguretat, on el vehicle es podria moure en zones perilloses sense posar en perill les persones que haguessin de controlar l'aparell.

Altres sectors on la seva evolució i desenvolupament és molt accentuada és el camp de la robòtica. En aquest camp també es pot incorporar solucions treballades en aquest projecte.

3. Descripció funcional

El disseny del sistema funcional d'aquest projecte el podem dividir en tres parts, on la part principal amb més pes és el codi que s'implementarà en la LPC que controla el funcionament de tot el sistema. La segona part important és el control del vehicle, des de que surt el senyal de la LPC fins l'accionament dels motors. I la tercera part és la interacció entre l'usuari i tot el sistema.

3.1. Sistema total

Per poder interactuar amb l'usuari és necessari disposar d'una connexió a internet. Per tant el primer pas que fa l'aplicació al iniciar-se, és buscar un punt d'accés per a poder-se connectar.

Està dissenyat per a buscar les xarxes disponibles, d'aquestes selecciona les xarxes que coneix la contrasenya, i d'aquestes es connecta a la que te millor senyal.

Si la recerca és negativa, s'activa una funció on el vehicle es desplaça pel tal de trobar una zona on la cobertura sigui suficient per tal de poder-s'hi connectar.

Un cop troba un punt d'accés realitza la connexió. En cas d'èxit, el programa continua amb la creació de dues tasques que s'executen simultàniament.

La primera tasca consisteix en demanar al servidor quina és la instrucció a seguir. Es pot trobar que l'usuari no hagi escrit cap instrucció. En aquest cas anirà fent consultes successives fins a trobar una instrucció nova. La instrucció que li arribarà del servidor contindrà la informació de la direcció, el sentit, i el temps d'execució d'aquesta instrucció i a la velocitat que l'ha d'executar.

La segona tasca és la d'analitzar cada una de les diferents instruccions que li arriben. De les instruccions analitzades, es poden trobar dos instruccions especials. La primera "Parat (S)", que deixa el vehicle en la posició actual. La segona instrucció especial és "Buscar el punt on te millor cobertura wifi (B)". Aquesta funció mou el vehicle per tal d'entrar dins una zona de cobertura de la Wifi on la seva RSSI (Indicador de força de senyal de recepció) estigui per sota del llindar establert.

La resta d'instruccions per moure el vehicle es poden observar en la següent figura, on del servidor rebrà les inicials en anglès del moviments a realitzar.

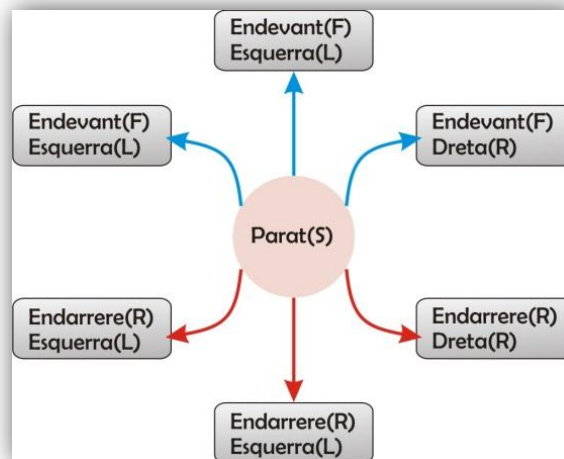


Figura 15. Esquema de direcció

Un cop analitzades les instruccions, es dona les ordres adients al motor per tal de realitzar els moviments demanats. En cada moviment que demana l'usuari, la LPC farà el seguiment del RSSI. En el cas que el valor fos molt petit o perdés del tot el contacte amb el punt d'accés, la LPC activaria una funció per tal de moure el vehicle buscant una zona amb bona senyal.

A més a més, en cada moviment la LPC també envia el valor RSSI trobat i el temps que la WiFly està connectada.

3.1.1. Diagrama de blocs de l'aplicació

Servidor: Serà l'element on interactuarà l'usuari. Haurà de clicar en algun dels botons que indicarà el moviment del vehicle. Al mateix temps rebre dades de la mota per tal que l'usuari estigui informat del moviment que realitzarà i també en quin punt d'accés esta connectat.

WiFly: Element que utilitzarà la mota per connectar-se al punt l'accés amb millor cobertura.

LPC1769: Element central del dispositiu, el qual conte el Firmware per controlar els següents processos:

- Buscar el punt d'accés amb millor cobertura i connectar-se.
- Llegir dades d'internet, via WiFly.
- Interpretar aquestes dades.
- Executar les ordres per tal de moure el vehicle amb els paràmetres rebuts.
- Enviar dades al servidor, via Wifly.

Grup activació motors: dispositiu electromecànic i electrònic que és activat per la LPC en el moment que es vol posar en marxa els motors del vehicle. Serveix també com a protecció de la LPC ja que la tensió de la LPC es diferent al circuit dels motors.

Vehicle: suport mòbil, on estarà ubicat el microcontrolador, WiFly, relés i sistema d'alimentació. Disposa d'un motor pel moviment endavant/endarrere i un altre motor que controla la direcció. Rebrà les ordres del microcontrolador, procedents de l'usuari, podent anar a dreta i esquerra, endavant i enrere o esta parat.

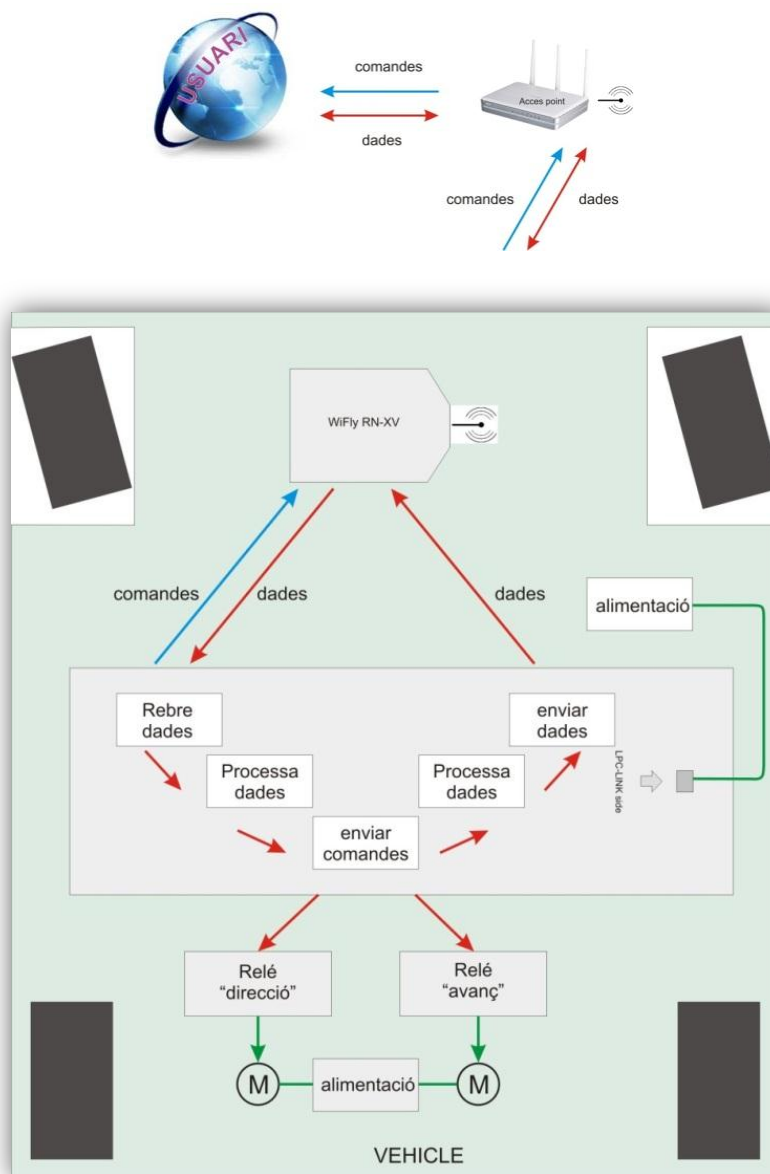


Figura 16. Diagrama de blocs de l'aplicació

3.2. Recerca de xarxa

Aquesta funció fa un escaneig de totes les xarxes disponibles. Si troba alguna de les xarxes de que es disposa de la contrasenya, compara el que te millor RSSI. Aquesta funció ens diu finalment quina és la xarxa escollida i el valor del RSSI

3.3. Connectar Xarxa

Un cop trobada la xarxa on ens podem connectar, mitjançant la WiFly, enviem les ordres oportunes, entre elles la contrasenya per tal de tenir accés a internet

3.4. Llegir instruccions del servidor

Amb l'ajuda d'aquesta funció, obrim el servidor on estarà ubicades les instruccions. En el nostre cas el servidor estarà en el nostre propi Pc. A continuació es demana que ens digui quina és la següent instrucció que s'ha de realitzar. El servidor ens respondrà amb una nova instrucció, o si no n'hi ha cap, ens dirà que la llista esta buida.

3.5. Executar instruccions

Un cop tenim una instrucció per seguir, analitzem de quina instrucció es tracte. Com ja s'ha comentat anteriorment, ens poden arribar dos instruccions especials. Estar parat o buscar el punt on te millor cobertura.

Aquestes instruccions segueixen un camí diferent a les altres instruccions. En aquest cas, estar parat, deixa el vehicle en la mateixa posició. La segona opció te la missió de trobar un punt on el seu RSSI estigui per sobre del llindar preestablert.

La resta d'instruccions segueixen les indicacions de l'usuari, és a dir, el vehicle pot anar endavant o endarrere, cap a la dreta o l'esquerra a voluntat de l'usuari.

3.6. Enviar dades

De les instruccions anteriorment descrites, sols les que tenen moviment controlat per l'usuari, s'enviaran les dades de control. En aquest cas s'envia el valor del RSSI que es captura al final de cada moviment i també s'envia el temps que fa que la WiFly està en servei.

3.7. Buscar el punt d'accés amb millor cobertura

Aquesta funció, fa moviments cap a endavant i cap endarrere per trobar en quina direcció va augmentant el valor. Si per exemple es cap endavant, va avançant fins que el valor disminueix. A continuació el vehicle gira 90° i repeteix la operació.

3.8. Servidor (Disseny de la interfície d'usuari)

L'aplicació dissenyada per a que l'usuari pugui donar les instruccions al vehicle, consta de dos parts. La part superior de la finestra, tal com es pot veure en la Figura 17. Interfície d'usuari, estan ubicats els botons per a controlar els moviments del vehicle. A més a més, en aquesta part trobem els valors per defecte que se li donaran al temps i a la velocitat d'execució. També es troba el boto per a buscar d'una manera automàtica, el punt on te millor cobertura Wi-Fi.

En la part inferior d'aquesta finestra, trobem l'informació que la LPC envia al servidor, i també el llistat de les instruccions demades per part de l'usuari.

En aquest llistat format per les inicials de les intruccions, i els valors del temps i velocitat, s'afegueix una lletra de control, que informa a l'usuari si l'instrucció ha estat llegida per la LPC (S), o encara està pendent de lectura (N)

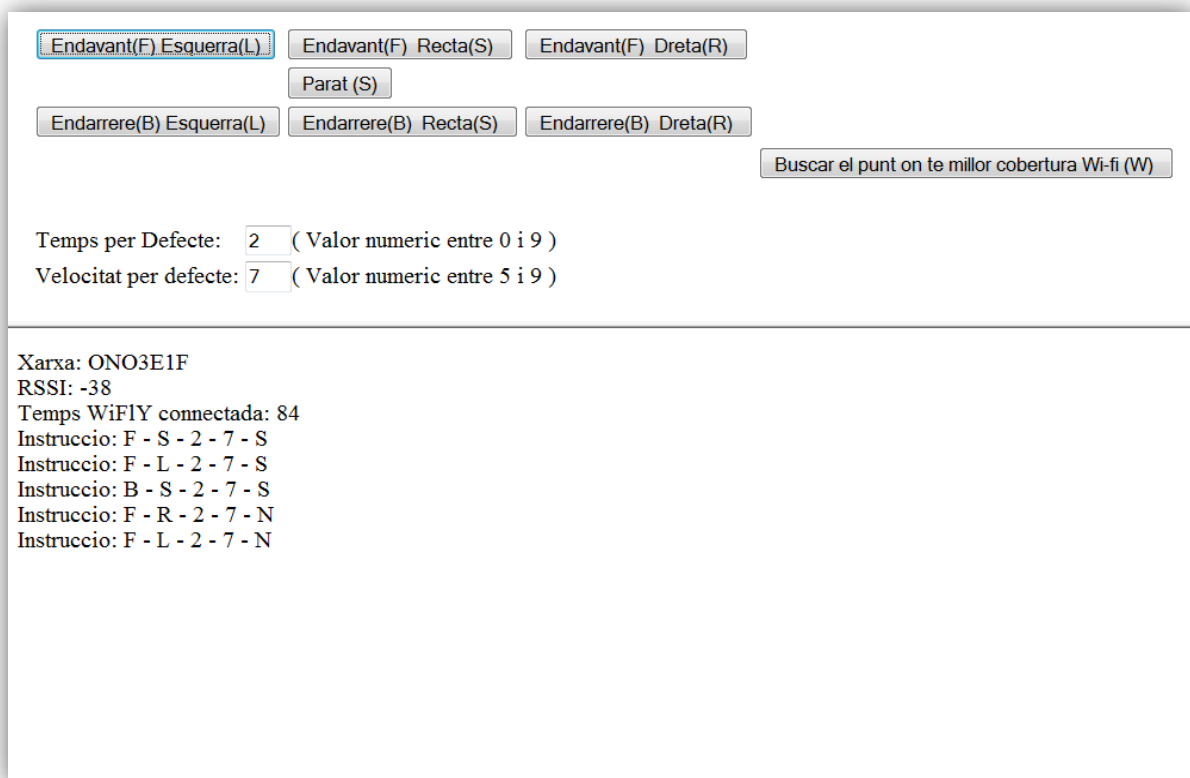


Figura 17. Interfície d'usuari

3.9. Control de motors

Un cop la LPC te clar el moviment que ha de realitzar el vehicle, ha de passar algun tipus de senyal o activació del motors.

Inicialment el projecte estava pensat en utilitzar el port 0 de la LPC on amb una placa controladora de motors, podia transmetre i rebre informació. Aquesta controladora de motors,

un cop rebia una instrucció de la LPC, l'executava directament al motor. Els paràmetres que havia de passar la LPC a la controladora era quin motor volia activar, en quin sentit i amb quina potencia. A mes a mes, per part de la controladora, enviava el resulta de l'execució de l'acció.

Però per una averia de la LPC, aquest sistema d'activació de motors no s'ha pogut portar a terme en aquest projecte.

L'opció finalment optada pel control de motors ha esta la d'utilitzar 4 pins de sortida de la LPC que poden donar un senyal d'actiu o inactiu. Així per cada motor utilitzarem 2 pins, un per activar o desactivar el motor i l'altre pin per donar la polaritat o sentit de gir del motor.

Si en el cas de la controladora de motor, la mateixa font d'alimentació de la placa servia per alimentar els motors, en aquest cas no és possible alimentar els motors directament amb el senyal que dona la LPC.

Com a conseqüència s'ha desenvolupat un circuit paral·lel per cada motor, que pogués ser activat amb els pins de la LPC. Per tal de protegir el circuit de la LPC del circuit de control de motors, s'ha dissenyat un petit muntatge amb diferents components electrònics. D'aquesta manera, el voltatge més elevat que necessita els motors, no pot afectar la placa LPC sensible a aquest voltatges.

3.10. Seguiment de l'evolució del programa

Un dels port de que disposa la LPC, s'utilitza per enviar missatges, que es poden llegir a traves de la CP2102 mitjançant PuTTY. D'aquesta manera, es pot fer el seguiment del programa connectant un cable USB a la CP2102 que porta el vehicle. Si s'utilitza un programa com el PuTTY o el Cutecom, surt per pantalla diversos missatges introduïts en el codi per tal de veure l'evolució del programa.

Aquest seguiment no s'utilitza habitualment en la utilització del vehicle, és a dir, el vehicle en condicions de treball normals, no necessita cap cable de cap tipus, ja que recordem, disposa de recursos d'alimentació autònoms suficients per activar i moure el sistema, i la comunicació és fa via WIFI.

El cable, que es pot connectar opcionalment, pot servir per fer un control de seguiment puntual, però no d'una manera continua, ja que pot limitar els moviments del vehicle degut a la seva llargada.

4. Descripció detallada

L'aplicació central d'aquest projecte s'ha desenvolupat en l'entorn de programació LPCXPRESSO amb llenguatge de programació C.

Treballem amb FreeRTOS que és un sistema operatiu per microcontroladors que permet multithreading, és a dir, permet fils d'execució independents amb una memòria assignada a cadascun d'ells i existeix un scheduler que va canviant de context i assignant uns cicles de rellotge a unes tasques i altres, permetent sistemes multitasca.

En quant a les entrades i sortides disponibles de la LPC, s'utilitza el port 3 per enviar i rebre dades de la WIFly, que serà el recurs més utilitzat durant l'execució del programa.

Tal com s'ha comentat en l'apartat anterior, en el port 0 de la LPC s'ha connectat la CP2102. Així, opcionalment, mitjançant el PuTTY, podem veure els diferents missatges del programa. En els següents apartats, indicarem els missatges de seguiment.

L'últim port disponible, el port 1, estava destinat al control dels motors, però una averia de la sortida TX va obligar a canviar la manera de activar-los. Finalment s'ha optat per re-utilitzar els pins que indicaven amb un LED el moviment del vehicle.

4.1. Sistema total

El primer pas que fa l'aplicació principal al iniciar-se, és buscar un punt d'accés per a poder-se connectar. Per a fer això, inicia la WIFly i executa la funció `WiFLYSearchNetwork` que ens retorna el valor de la RSSI trobat i el nom de la xarxa. Per a fer les proves es disposa de dos punts d'accés, amb les contrasenyes corresponents. Per tant l'aplicació busca quin dels dos punts d'accés està disponible, i en cas que els dos ho estessin, es connecta al de millor RSSI.

En cas que la recerca fos nul·la, es fa moure el vehicle amb `WiFlysearchAccespoint` fins a trobar una zona de cobertura. A continuació es torna a comprovar si s'ha trobat una xarxa on poder-se connectar. I així successivament fins a trobar un punt correcte.

Un cop troba un punt d'accés realitza la connexió amb la funció `WiFlyConnectNetwork`. En cas d'èxit, ens retorna un valor "1" permeten al programa continuar, enviant al servidor el nom de la xarxa on esta connectat i a la CP2102 el nom de la xarxa i el valor RSSI

A continuació crea una cua, on es guardaran les instruccions que es llegeixin del servidor que posteriorment s'aniran executant. També es genera un semàfor per tal d'assegurar que els recursos limitats no siguin utilitzats per més d'una funció al mateix temps.

El següent pas es creen dues tasques. Aquestes tasques, gràcies a FreeRTOS treballaran d'una manera simultània, tenint totes dues la mateixa prioritat.

La primera tasca consisteix en la lectura de les instruccions que hi ha en el servidor que l'usuari ha introduït prèviament. I posar aquestes instruccions a la cua.

La segona tasca és la de llegir les diferents instruccions que estan a la cua , analitzar-les i executar-les. En cada moviment que fa el vehicle analitza el RSSI i l'envia al servidor

4.1.1. Tasca vDemanaInstruccio

Un cop el programa entra en aquesta tasca, sol·licita el semàfor per tal d'utilitzar la WiFly i obtenir informació del servidor. La funció és `trobatInstr =ReadProperalInstruccio(&instr);`

La informació que es rep del servidor, detallat en el punt 4.4, es transforma en una estructura (instr) de 4 camps, direcció, sentit, temps i velocitat, que s'enviarà a la cua, en la qual es poden emmagatzemar fins a 5 instruccions.

Per poder fer el seguiment mitjançant la CP2102, llegirem cada un d'aquets camps i l'enviarem pel port 0

En cas de no haver-hi cap instrucció disponible, el servidor indica "buida". En aquest cas, la funció retorna `trobatInstr = -1`, indicant que no hi ha valors disponibles. En conseqüència no s'envia res a la cua i s'envia pel port 0 que s'espera un temps.

4.1.2. Tasca vMotorDrive

Primerament aquesta tasca, intenta llegir una instrucció de la cua. Si hi ha èxit, analitza la informació per saber les ordres que s'han de donar al motor.

De les instruccions analitzades, es poden trobar dos instruccions especials. La primera "Parat(S)", que deixa el vehicle en la posició actual. La segona instrucció especial és "Buscar el punt on te millor cobertura WiFi (W)" . Aquesta funció mou el vehicle per tal d'entrar dins una zona de cobertura de la WiFi on la seva RSSI estigui per sota del llindar establert, (funció que es comentarà amb més detall en el punt 4.7). Per executar aquesta funció, el programa sol·licita el semàfor i no el deixa fins que l'operació s'ha acabat.

Si la instrucció llegida no és cap de les dues anteriors, continua el programa analitzat quin moviment ha de realitzar el vehicle. En aquest cas tenim sis possibilitats:

- Endavant recta (FS)
- Endavant esquerra (FR)
- Endavant dreta (FL)
- Endarrere recta (BS)
- Endarrere esquerra (BR)

- Endarrere dreta (BL)

Per cada una d'aquestes possibilitats, el programa activa una funció específica per a cada moviment, indicant-li la durada del moviment.

Seguidament es sol·licita el semàfor per a poder utilitzar el recurs de la WiFly i poder llegir el valor de la RSSI. Si degut al moviment realitzat el vehicle surt de l'àrea de cobertura del punt d'accés, el valor rebut serà -3. En aquest cas aturarà totes les tasques per tal de buscar un punt d'accés com es feia a l'inici del programa. Una altra possibilitat podria ser rebre un valor del RSSI inferior al llindar mínim establert a -150db. En aquest cas s'activaria la funció de trobar buscar el punt on te millor cobertura WiFi (W)

Arribat a aquest punt s'envia al servidor el valor RSSI trobat. A més a més es demana a la WiFly el temps que està en marxa des de l'última inicialització. Igualment, com en les altres operacions, en aquestes també es demana el semàfor per tal de no utilitzar dos funcions el mateix recurs.

4.2. Recerca de xarxa "WiFLYSearchNetwork"

Per a fer totes les proves de xarxes, s'han utilitzat el router disponible de la connexió ADSL local i la connexió WiFi que pot generar un telèfon mòbil. Per tant en el codi s'ha introduït el nom d'aquestes xarxes juntament amb les contrasenyes corresponents. En el cas d'utilitzar aquest vehicle en algun altre escenari, s'hauria d'introduir tant el nom com la contrasenya adient.

Mitjançant la funció `WiFlyScan` busquem totes les xarxes que captura la WiFly. A continuació fem un anàlisi sintàctic (*parser*) del capturar per la LPC, per saber si de les xarxes trobades ha trobat una o més d'una de les xarxes que s'utilitzen per fer les proves.

En cas que trobes les dues, analitza quin valor RSSI és el més alt, retornant el nom i el valor RSSI al programa principal.

En cas de no trobar cap xarxa de les dues, retorna el valor Null com a nom de xarxa i valor de RSSI -1000 al programa principal, per tal d'informar-li.

4.3. Recerca de punt d'accés "WiFlysearchAccesspoint"

Aquesta funció s'activa en el moment que el vehicle no troba cap xarxa per connectar-se o si en algun moviment que ha realitzat ha perdut la cobertura del punt d'accés.

El moviment que segueix aquesta recerca, consisteix en un moviment en forma de creu. A cada un del passos que descriurem a continuació sol·licita a la WiFly el valor de la RSSI. En el cas que el valor sigui diferent -3 (valor que ens indica que no ha trobat cap xarxa) la funció finalitzaria.

Així, primerament avança en línia recta cap endavant una distància d . Comprova els punts d'accés i en cas negatiu recula una distància $2d$. Torna a fer la comprovació del punts d'accés i en cas negatiu retorna al punt d'origen i gira 90° . Torna a avançar una distància d comprovant els punts d'accés. En cas negatiu, recula una distància $2d$.

Arribat en aquest punt el vehicle haurà fet un moviment de creu. Si la recerca ha estat negativa, torna a repetir el mateix procés anterior però ara doblant la distància. I així successivament.

4.4. Connectar-se a la xarxa “WiFlyConnectNetwork”

El programa principal passa a la funció el nom de la xarxa que s'ha de connectar. D'aquesta manera, fent una comparació, pot executar la part de la funció exclusiva per a cada xarxa.

Inicialment, al no saber en quin estat està la WiFly, la inicialitzem, li indiquem el nom de la xarxa i la contrasenya i li donem l'ordre de connectar-se.

Capturem el que respon la WiFly per assegurar que la connexió s'ha realitzat amb èxit. Si la connexió és correcta finalitza la funció tot enviant al port 0 l'èxit de la connexió i retornant un valor de control 1 al programa principal. En cas negatiu, el procés es repeteix fins a 3 vegades indicant pel port 0 el número d'intents que s'estan executant. En cas que no s'arribés a connectar, li passaria al programa principal un valor de control -1.

4.5. Llegir instruccions del servidor “ReadProperaInstruccio”

Per a poder llegir una nova instrucció del servidor, el primer que fa és obrir el servidor Tomcat que està instal·lat com a localhost per port 8080. A continuació fent un `GET /ControlWeb/properaInstruccio` per demanar la propera instrucció tot esperant la resposta que es memoritzarà al buffer del port 3.

Es fa una comprovació inicial per saber si la informació rebuda conté instruccions vàlides. En cas afirmatiu, es fa un anàlisi de la informació rebuda.

Aquesta informació rebuda, que per exemple, podria ser “ F - S - 2 – 7”, l'anem separant i guardant a cada camp correspon a l'estructura d'una instrucció. Així, en aquest cas es guardaria el valor F en el camp direcció, el valor S en el camp sentit, el valor 2 en el camp temps i el valor 7 en el camp velocitat. Aquesta informació serà analitzada, posteriorment, per altres funcions.

Finalitza la funció tancant la connexió amb el servidor i retornant un valor de control on l'hi indica si ha pogut llegir una instrucció vàlida o no.

4.6. Executar instruccions

En aquest apartat, s'explicarà com es dona la ordre des de la LPC, per activar els motors.

Com ja s'ha comentat a l'inici d'aquest capítol, el projecte estava enfocat per que la LPC controlés els motors mitjançant un controladora de motors. Aquesta controladora que es va provar inicialment amb bons resultats és la Serial Controlled Motor Driver ROB 09571.

Però degut a una averia de la Tx del port 1 de la LPC, el projecte no s'ha pogut acabar utilitzant aquesta controladora. De totes maneres en el codi del programa, també hi ha la part del control dels motors mitjançant la controladora, encara que està comentat perquè no afecta a la bona execució del global del programa. Està demanada una LPC nova per tal de poder finalitzar el projecte tal com s'havia pensat inicialment. Per tant, es considera oportú explicar en aquest apartat, l'execució de les ordres cap al motor, mitjançant les dos maneres possibles.

4.6.1. Executar instruccions per mitja de pins de sortida de la LPC

Aquesta és l'opció de com el projecte final executa les instruccions. És més fàcil de codificar però pots controlar menys paràmetres. A més a més s'ha d'afegir una part de components electrònics entre els pins i els motors, per tal de protegir la LPC, que complica el muntatge final del vehicle.

La LPC disposa de varis pins que es podem controlar donant un senyal de 3.3v de sortida. Dels que disposa la placa, s'han escollit els següents:

- Pin P0(23): indicador gir esquerra
- Pin P0(24): indicador endavant
- Pin P0(25): indicador gir dreta
- Pin P0(26): indicador endarrere

En cada una de les instruccions possibles que rep el motor (FS), (FR), (FL) , (BS), (BR) i (BL) n'explicarem dues com a representatives de les 6 possibles.

El primer cas és quant el vehicle va endavant o endarrere però en línia recta (FS) i (BS). En aquest cas la funció sols activa el pin corresponent, espera el temps que la instrucció li indica i desactiva el pin.

El segon cas, que afecta les altres quatre possibilitats, la funció ha d'activar dos pins al mateix temps. Un per la direcció i l'altre pel motor d'avanç. Si sols actives la direcció, el vehicle no es mouria, ja que les rodes de direcció no tenen tracció.

De cada un dels pins que s'activa surt un voltatge de 3.3v. Amb aquest voltatge seria perillós i no suficient per a alimentar els motors directament. Per això s'ha optat per intercalar, entre els pins i el motors, un optoacoblador, un transistor, resistències, díodes i dos relé, per tal que una

font d'alimentació externa alimenti els motors. D'aquesta manera es protegeix la LPC de qualsevol sobrecarrega del circuit. L'explicació del muntatge està detallat en el capítol 4.11.1

Un cop ha acabat d'executar l'acció corresponent, s'envia pel port 0, la notificació del moviment que acaba de realitzar. (recordem que pel port 0, sols es pot capturar els missatges si es connecta, amb un cable USB, la CP2102 amb el PC).

4.6.2. Executar instruccions per mitja de la controladora de motors

A través de la TX i RX del port 1 es pot controlar els motors mitjançant la *Serial Controlled Motor Driver*. No és la versió final del projecte, però sí que conte el codi necessari per ser activat en el moment en que es disposi d'una nova LPC (recordem que l'actual té una averia en el sortida Tx).

Per controlar el motor s'ha d'enviar tres paràmetres a la controladora de motors. El primer paràmetre és el motor a controlar, que pot ser 1 o 2. El segon paràmetre és el sentit de gir del motor, és a dir la polaritat del motor, f o r. I el tercer paràmetre és la potencia que li dona al motor, valors entre 0 i 9, utilitzant el valor 0 per parar el motor.

En aquesta versió de control de motors, s'han utilitzat els pins comentats en l'apartat anterior per posar uns leds al vehicle per indicar la direcció i sentit que seguirà el vehicle. S'activen de la mateixa manera que l'apartat anterior, però canviant els motos per leds.

El procés que segueix cada funció per activar els motors , comença amb l'activació del pin corresponent, continua enviant l'ordre a la controladora de motors, espera el temps indicat per l'usuari, i finalment desactiva tant el led com envia l'ordre de parar el motor a la controladora.

La controladora de motors envia missatges per tal de poder controlar si el moviment s'ha efectuat correctament o una alerta en el cas que el motor tingui molta resistència al avançament. Però al treballar les dues plaques amb tensions diferents, s'ha de col·locar un divisor de tensions entre la Rx de la LPC i la TX de la controladora de motors, com a protecció de la LPC.

Igualment que en l'apartat anterior, s'envia pel port 0, la notificació del moviment que acaba de realitzar.

4.7. Enviar dades

Les dades de control que s'envien al servidor per tal que l'usuari estigui informat són les següents:

- Nom de la xarxa on està connectat.
Sols s'envia una vegada a l'inici de la connexió, mitjançant la funció `SendServerNetwork(Network)` . Com que les tasques encara no estan actives,

aquest procés no demana l'ús del semàfor. La funció obre una connexió amb el servidor tomcat. A continuació li envia el nom de la xarxa amb la comanda `GET /ControlWeb/informaXarxa?xarxa=` més el nom de xarxa, Finalment informa pel port 0 que s'ha enviat el nom de la xarxa i tanca la connexió amb el servidor.

- Valor RSSI

El valor RSSI que informa la WiFly en cada moviment del vehicle, és enviat al servidor. La funció `SendServerRSSI(RSSI)` ha de sol·licitar prèviament el semàfor, ja que utilitza el recurs de la wifly, que altres processos el poden emprar.

Com en el cas anterior, aquesta funció obra una connexió amb el servidor i li envia la RSSI amb la comanda `GET /ControlWeb/informaXarxa?rssi=` més el valor de RSSI. A continuació envia a la CP2102 el missatge que ja s'ha enviat el valor i a continuació tanca la connexió amb el servidor. El valor RSSI ens pot donar una idea aproximada de la distància a que es troba el vehicle respecta el punt d'accés.

- Temps de la WiFly

Igual que en el valor de RSSI, en cada moviment que fa el vehicle, demana a la WiFly el temps que està en marxa des de l'última inicialització. Aquest valor s'envia al servidor amb la funció `SendServerWiTime(Witime)`. Per enviar la dada, prèviament també ha de demanar el semàfor, obrir una connexió amb el servidor i enviar la dada mitjançant la comanda `GET /ControlWeb/informaXarxa?WiTime=` més el valor del temps passat per paràmetre. La resta de la funció segueix els mateixos passos que les altres dues funcions anteriors.

4.8. Buscar el punt d'accés amb millor cobertura

La funció que busca el punt d'accés amb millor cobertura, és un dels requisits bàsics d'aquest projecte. Aquesta funció pot ser activada per l'usuari des de la pàgina del servidor o també pot ser activada en el moment que el vehicle s'allunya molt del punt d'accés.

La funció en qüestió és `trobat = MotorAutomaticBack (MinimRSSI)`, on s'ha de passar per paràmetre el valor mínim de RSSI que s'exigeix que trobi i retorna un dígit de control confirmant o negat l'èxit de la operació.

La primera acció és demanar a la WiFly el valor de la RSSI que te en aquell moment. En el cas que estés fora de cobertura, activaria la funció `WiFlysearchcAccespoint` per buscar un punt amb un mínim de cobertura.

A partir d'aquest moment el procés entra en un bucle que no surt fins que troba el punt desitjat o fins que ha realitzat l'operació que descriurem a continuació 10 vegades. (si no es troba amb

10 vegades es pot considerar que el vehicle està molt lluny de l'àrea de cobertura de punt d'accés).

El procés de recerca consisteix en avançar el vehicle en petits moviments controlats per el valor temps. En cada moviment compara el valor RSSI que ha trobat en l'última consulta respecte al valor trobat en realitzar el nou moviment. Si el valor és més alt, és a dir, millora el senyal, el moviment es repeteix, si no, fa els desplaçaments en sentit contrari.

Si en qualsevol de les lectures de RSSI troba un valor per sobre del exigít (punt trobat), el programa finalitza.

Per fer aquests moviments, comencem amb la funció `MotorForwardStraight`. Aquesta funció realitza els moviments cap endavant i es va repetint mentre millora el senyal (indicador que es va en bona direcció). En el cas que en el primer moviment que fes el valor disminuís ens indicaria que es va en sentit contrari. Si no ha trobat el punt buscat, l'últim moviment que haurà fet el valor RSSI serà més petit que el penúltim moviment. Per tant activem la funció `MotorAutomaticBack` que també fa el petits moviments però en sentit contrari, és a dir endarrere.

Arribat aquest punt, el vehicle estarà parat en un punt pràcticament tangent a una circumferència amb centre al punt d'accés.

Si la zona buscada encara no s'ha trobat, fa un gir de 90°. D'aquesta manera quedarà encarat en la direcció del punt d'accés, però no sabrà si el sentit és el correcte.

Per tant torna a repetir el procés d'anar endavant mentre millori el senyal o si en el primer moviment el senyal empitjora, canviarà de sentit.

En la Figura 18. Moviment de recerca del millor punt d'accés, podem observar un exemple del moviment que faria un vehicle per trobar el millor punt d'accés.

En aquest exemple el vehicle parteix del punt A. El primer moviment "1" va en sentit contrari. Per tant en la següent lectura del valor RSSI veurà que és inferior al primer i en conseqüència, el vehicle es mourà en sentit contrari. Es continuarà movent (punts 2,3,4,5) fins al final del punt 5, on el valor de la RSSI serà inferior que el llegit en el punt 4.

A continuació farà un gir de 90° (moviment 6), arribant al punt C. Aquí torna a repetir el procés. Primer fa un moviment endavant "7" i al adonar-se que el valor disminueix, canvia de sentit, fent els moviments 8, 9, 10. On finalment troba una zona(punt C) on el valor RSSI està per sobre del valor demanat.

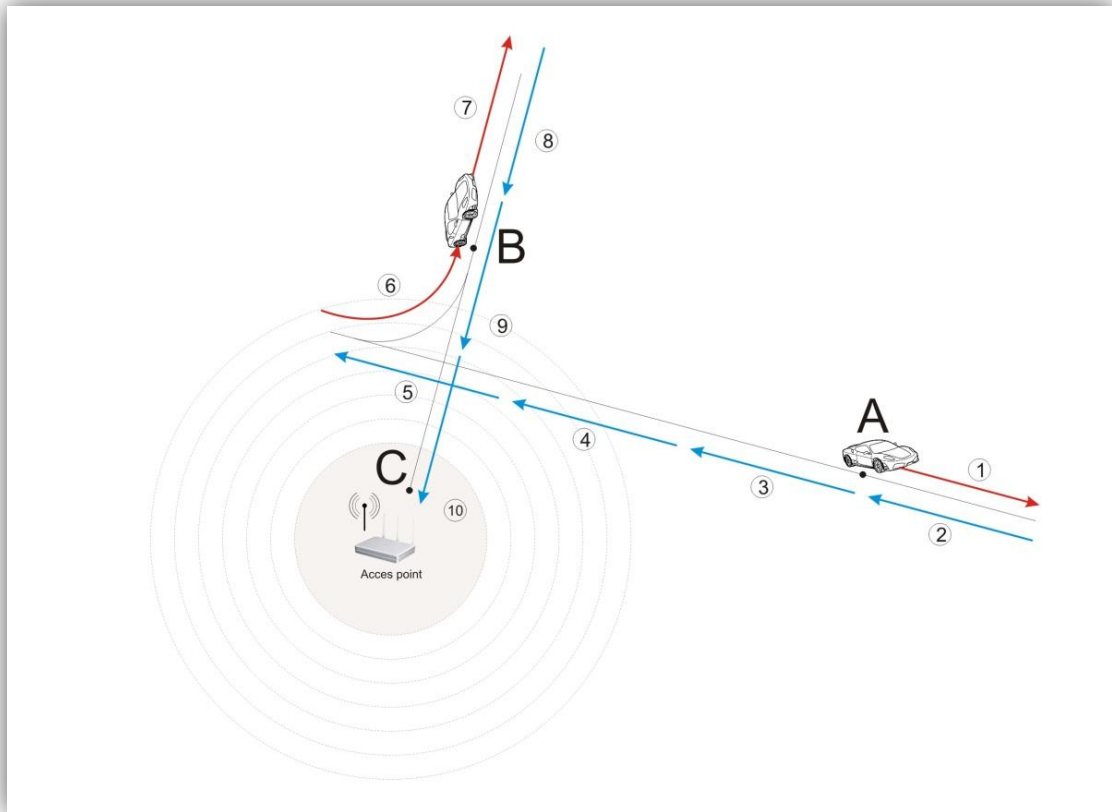


Figura 18. Moviment de recerca del millor punt d'accés

4.9. Servidor (Disseny de la interfície d'usuari)

La comunicació entre l'usuari i l'aplicació es realitza mitjançant un codi realitzat en l'Eclipse i una pagina html, tot penjat en un servidor Tomcat. Per fer les proves, aquest servidor està instal·lat en la màquina local. D'aquesta manera, l'accés a aquesta pàgina es pot fer mitjançant l'enllaç <http://localhost:8080/ControlWeb/index.html>.

En el codi podem trobar diferents servlet.

En el primer implementem un servlet per guardar les instruccions que demanarà l'usuari, on es guardarà la direcció, sentit, temps i velocitat. A més afegirem un dígit de control "N", que ens indicarà que la instrucció no ha estat demanada pel vehicle.

Un segon servlet (properaInstruccio) permet a la LCP llegir la següent instrucció disponible. Cada vegada que es llegeix una instrucció, es modifica el dígit de control posant una "S" en lloc de la "N".

Es disposa d'un altre servlet que guardarà les dades que la LPC li enviarà. Aquestes dades seran: nom de la xarxa, valor de RSSI i el temps que la xarxa esta en ús. Esta pensat de tal manera que es pot actualitzar qualsevol del paràmetres, sense que els restants es modifiquin. També és fàcil introduir nous paràmetres de control, on en l'aparat de millores s'esmenten algunes possibilitats.

Un altre servlet està pensat per informar de les dades de control que ha rebut de la LPC. Ja sigui perquè és a l'inici del programa o be perquè LPC no ha informat alguns dels valors, aquest servlet informa amb valor null si no disposa de cap valor.

S'han dissenyat dos pagines HTML. Una d'aquestes disposa dels botons que utilitzarà l'usuari per controlar el vehicle. També disposarà del valor de temps i velocitats que indicarà l'usuari, donant la pàgina uns valors per defecte.

L'altre pàgina contindrà les dades dels valors de control i un llistat de totes les instruccions demanades per l'usuari, ja sigui que hagin estat llegides per la LPC o no.

Finalment s'ha fet un frameset per reunir les dues pàgines creades en una de sola. D'aquesta manera és més fàcil la visualització de tota la informació per part de l'usuari.

4.10. Diagrama d'estats

El diagrama d'estats es pot observar en la següent Figura 19. Diagrama d'estats

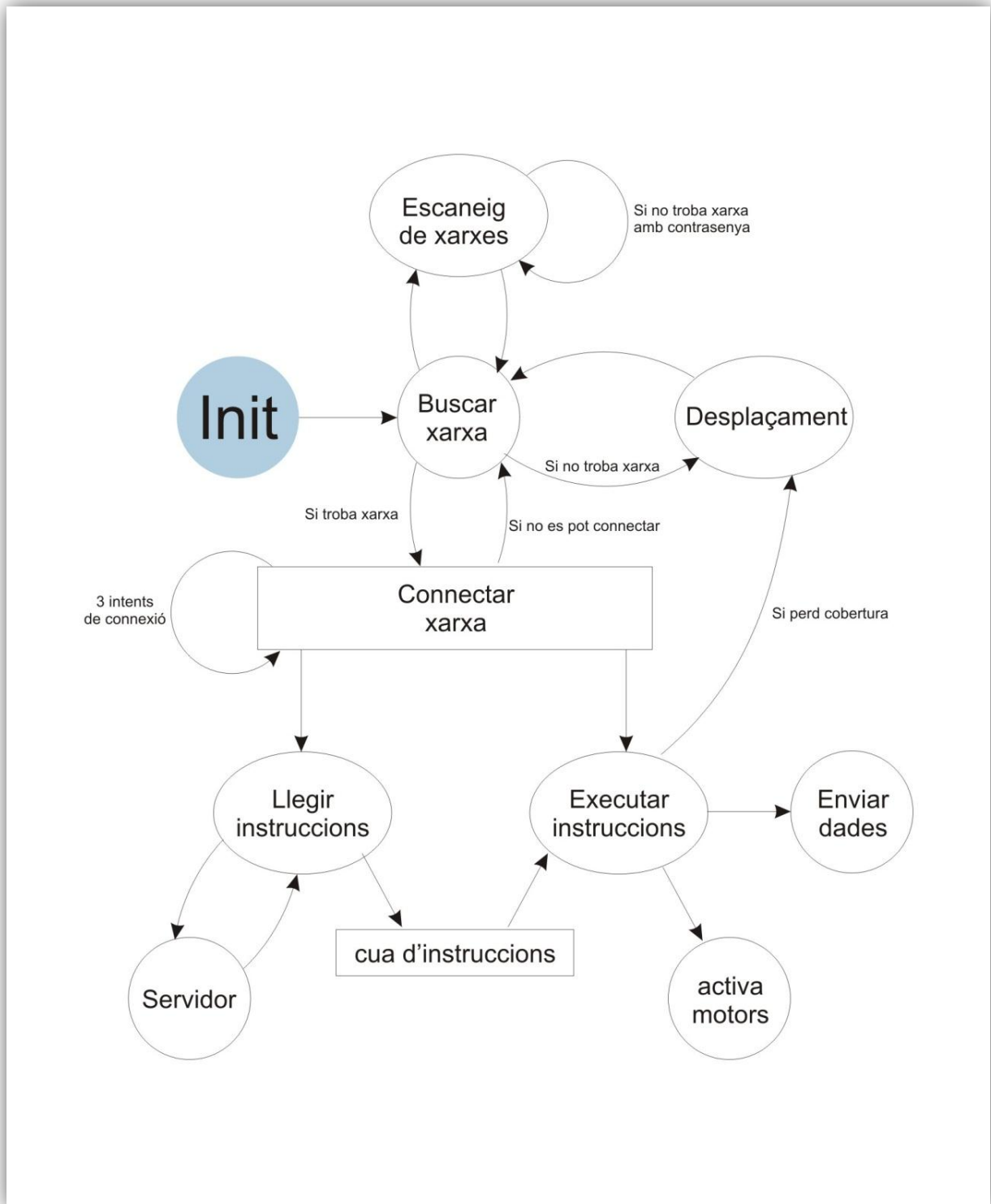


Figura 19. Diagrama d'estats

4.11. Esquema connexió

En aquest apartat s'han inclòs els dos esquemes possibles de connexió del microcontrolador LPC respecte als diferents recursos emprats, tenint en compte la utilització de la controladora de motors o mitjançant els diferents components electromecànics i electrònics per controlar els motors

4.11.1. Esquema de connexió directament des de la LPC

En la Figura 20. Esquema de connexió per pins podem veure la connexió utilitzada pel funcionament final d'aquest projecte. Podem observar la major complexitat alhora de controlar els motors.

Per a la sortida dels pins de la LPC surt un voltatge de 3.3v. Aquest voltatge és insuficient per a activar els motors. Per tant s'ha dissenyat un circuit d'adaptació per poder connectar els motors directament a la LPC. Els motors són considerats elements de potencia i els activarem per mitja de relés. El relés seran controlats per la LPC activant-los o desactivant-los per a controlar el vehicle en totes les direccions. Per a controlar els relés ho separarem de la LPC mitjançant un optoacoblador, que gràcies a la llum que genera un LED intern al ser activat i a un fototransistor que la capta, actuen com un interruptor sense contacte entre els dos circuits.

L'optoacoblador escollit és el CNY47-2, optoacoblador de propòsit general.

Fixem 10 mA que és la corrent típica d'un LED. Com la tensió de la LPC de sortida al LED del optoacoblador és de 3.3v col·locarem una resistència de:

$$\frac{3.3v}{10 mA} = 330 \Omega$$

El corrent màxim del fototransistor del optoacoblador és de 50 mA, segons la fulla de característiques del fabricant *datasheet*, és inferior a la corrent que necessitem per excitar a la bobina del relé (segon *datasheet* són 100 mA). Per tant ens caldrà col·locar un transistor en muntatge Darlington. Aquesta combinació ens serveix per proporcionar un gran guany de corrent.

El transistor escollit és BD 135, un transistor de propòsit general.

Per activar el relé de 6 v, necessitem 100 mA. El límit d'intensitat del col·lector del transistor escollit és 1,5 A, suficient per controlar el relé.

Per assegurar-nos que no danyarem el fototransistor, la corrent del col·lector del BD 135 és de 100 mA i el guany h_{FE} és 40, ens queda:

$$\frac{100 mA}{40} = 2.5 mA$$

Que serà la corrent que circularà per la base del transistor que serà proporcionada pel col·lector del optoacoblador. Aquesta corrent de 2.5 mA l'aguanta perfectament l'optoacoblador.

Finalment la col·locació de dos relés ve degut a poder controlar el sentit de gir (relé 1) i l'activitat del motor (relé 2) tal com es pot veure en la següent taula de veritat:

Relé 1		Relé 2		MOTOR
0	Gir positiu	0	inactiu	inactiu
1	Gir negatiu	0	inactiu	inactiu
0	Gir positiu	1	actiu	Gir positiu
1	Gir negatiu	1	actiu	Gir negatiu

Taula 1. Taula de veritat

La corrent emmagatzemada en la bobina del relé, quan es des-excita canvia de polaritat. Aquest canvi pot afectar parts dels components electrònics. Per evitar-ho es col·loca un díode que provoca que aquesta corrent es descarregui sobre ella mateixa.

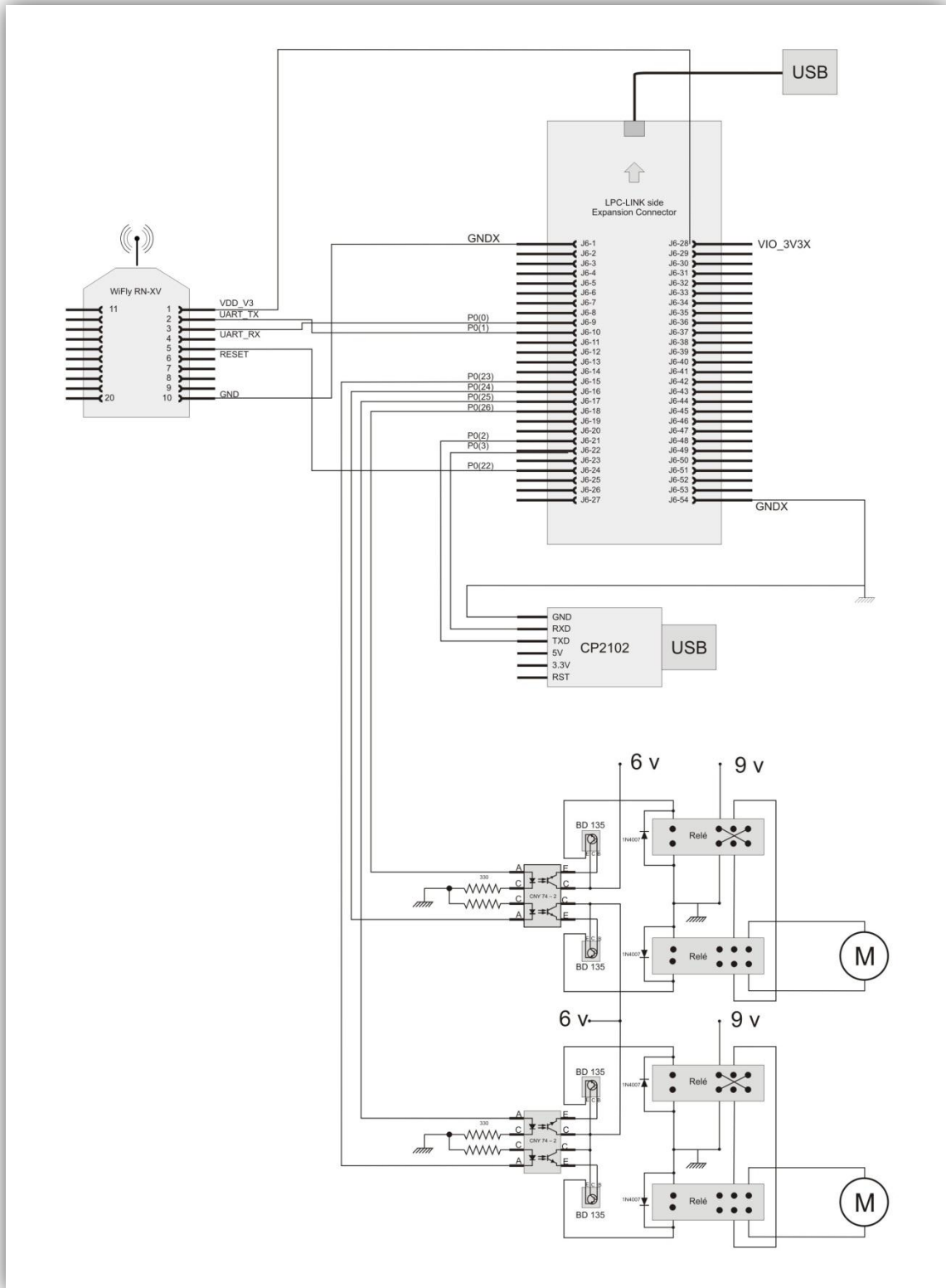


Figura 20. Esquema de connexió per pins

4.11.2. Esquema de connexió per mitja de la controladora de motors

L'opció plantejada inicialment consistia en la utilització d'una controladora de motor. Aquesta rebia una alimentació independent de la LPC per tal de poder activar sense problemes els motors.

S'havia d'instal·lar un divisor de tensió entre la sortida TX de la controlador i la RX de la placa LPC, per protegir la LPC de l'accés de tensió (5v.) que podia rebre de la controladora de motors.

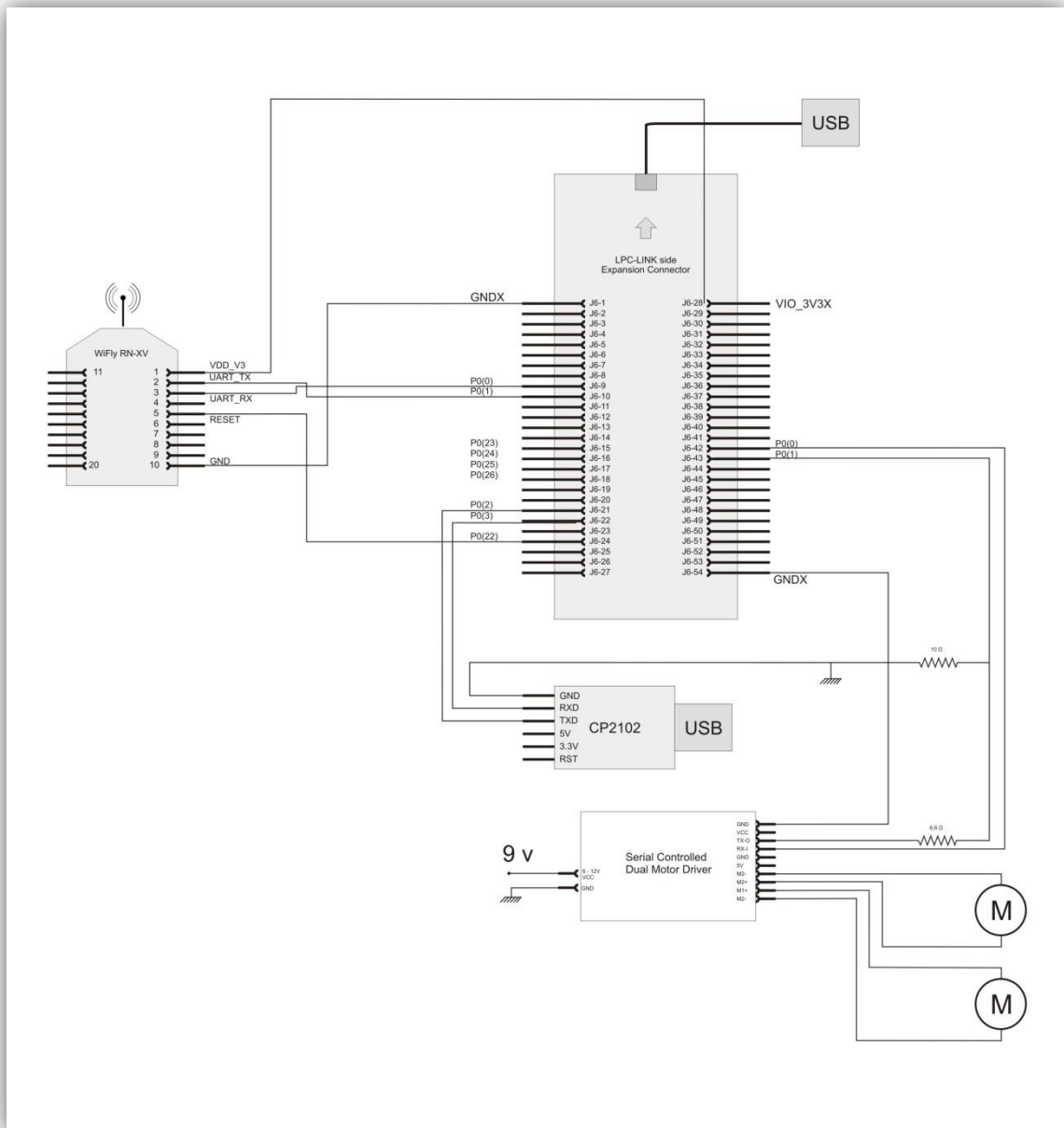


Figura 21. Esquema de connexió per controladora

4.12. Mostra les dades pel cutecom o PuTTY

Tal com s'ha comentat anteriorment, un dels recursos que el vehicle disposa és la CP2102, que la seva TX i RX esta connectada a la TX i RX del port 0 de la LPC.

Si es vol fer un seguiment de l'evolució del programa, es pot connectar un cable USB endollat al PC. (comentem que el cable ha de ser el suficientment llarg com per seguir els moviments que realitzarà el vehicle). Aleshores mitjançant un programa com el PuTTY o el cutecom es podrà captar les dades que envia la LPC. La majoria de processos que realitza, envia el resultat de la seva execució.

En la Figura 22. Captura de PuTTY, podem veure una captura d'una prova realitzada.

Al principi podem veure un scan de les xarxes disponibles, mostrant les dues xarxes disponibles en aquell moment. A continuació el programa internament es connecta a la xarxa corresponent, mostrant a la pantalla "connexió correcta". Seguidament ens indica que ha enviat el nom de la xarxa al servidor.

A continuació demana dades al servidor però aquest respon que la llista està buida. Posteriorment, l'usuari posa una ordre a la cua del servidor, que es llogida per la LPC, mostrant a la pantalla els valor que ha llegit del servidor.

Mostra a continuació els valors que ha llegit de la cua del programa, executa la instrucció, on en aquest exemple és "motor endavant esquerra".

Com que les tasques s'executen simultàniament, la tasca de demanar la següent instrucció s'avança respecte la d'enviar els valors de controls, tal com es pot veure en la captura de pantalla. Primer li indica que la llista d'instruccions del servidor està buida i a continuació demana el RSSI i el temps a la WiFly i l'envia al servidor.

```

COM3 - PuTTY
SCAN:Found 2
Num      SSID      Ch  RSSI   Sec   MAC Address      Suites
1         ONO3E1F  01 -37 WPA2PSK 5c:35:3b:34:4f:ea AES/TKIPM-TKIP  WPSPB 1104
2         ONO800420 11 -80 WPA2PSK e0:69:95:54:a4:c1 AES/TKIPM-TKIP   3104

connexio correcte
Send Karxa ONO3E1F
RSSI -37
Karxa ONO3E1F
resposta server: Buida
Esperem un temps:
server: Direccio: F, sentit:L, temps: 2
Valor queue: Direccio: F, sentit:L, temps: 2, velocitat: 7
Motor endavant esquerra
resposta server: Buida
Esperem un temps:
show rssi -38
Send RSSI -38
Temps ences UpTime=42 s
Send WiTime 42
server: Direccio: F, sentit:S, temps: 2
Valor queue: Direccio: F, sentit:S, temps: 2, velocitat: 7
Motor endavant
server: Direccio: F, sentit:R, temps: 2
show rssi -37
Send RSSI -37
Temps ences UpTime=71 s
Send WiTime 71
Valor queue: Direccio: F, sentit:R, temps: 2, velocitat: 7
Motor endavant dreta
server: Direccio: F, sentit:L, temps: 2
show rssi -38
  
```

Figura 22. Captura de PuTTY

5. Nocions del sistema operatiu en temps real FreeRTOS

En aquest capítol es donarà una noció del sistema operatiu en temps real utilitzat en aquest projecte.

El FreeRTOS és un sistema operatiu en temps real adreçat als petits sistemes encastats en temps real. Utilitza el kernel per a gestionar les diferents tasques. Aquestes tasques independents són el model de com s'estructura aquest sistema operatiu, i a més a més de la seva independència, només es pot executar una tasca en cada moment.

Una figura important en els sistemes operatius en temps real és el planificador o *scheduler*, que és el repartidor de temps de processament entre les diferents tasques. Ha de començar i finalitzar repetidament cada tasca tal com indica l'aplicació, podent assignar diferents prioritats per cada una d'elles. A més a més el planificador té la responsabilitat de guardar l'estat actual d'una tasca al moment de rellevar-la per una altre, i de retornar-li el mateix valor quan es torni a fer servir.

Les tasques es poden executar, bloquejar o posar en suspensió, ja que es disposa de diferents funcions específiques per implementar-les. Es poden també crear tasques dins de tasques, permeten moltes possibilitats de programació.

Un altre punt important de que disposa és la configuració dels ports sèrie a través del port RS232 UART. Ens dona la possibilitat de rebre i enviar informació bit a bit, connectant-lo a ordinadors, altres microprocessadors i a tot tipus de perifèrics.

Les cues són la forma més bàsica de comunicació entre tasques, on una tasca serà l'emissora de dades i altre tasca serà la receptora, on normalment l'emmagatzema en un registre FIFO (primer en entrar, primer en sortir).

Disposa també del semàfor, que ens permeten tenir el control sobre determinats recursos. Si una tasca necessita un recurs a d'agafar el semàfor, d'aquesta manera bloqueja l'accés a altres tasques mentre no l'alliberi.

6. Viabilitat tècnica

Encara que el prototip desenvolupat en aquest projecte funciona correctament, i la seva viabilitat tècnica sigui possible, la implantació al mercat d'aquest producte no seria possible sense algunes modificacions importants que també es comentarà en l'apartat de proposta de millores.

Un cop desenvolupada l'aplicació, la part de fabricació del producte per a esser posat al mercat és simple. Amb una plataforma amb rodes motoritzades i les plaques del microcontrolador, la WiFly i el driver del motor, es pot fabricar el prototip.

El problema està en que un vehicle que està dirigit via internet sense una visió directe de l'usuari, i sense sensor de cap tipus, el seu control pot ser nul. Com a conseqüència la seva comercialització seria inviable.

Per tant s'hauria de plantejar modificar els requisits inicials i dotar al vehicle de sensor o d'una videocàmera per tal que el vehicle, a més a més de ser dirigit, sigui controlat.

7. Valoració econòmica

Mostrarem a continuació una valoració aproximada del desenvolupament realitzat en aquest projecte. S'ha de tenir en compte, que a més a més de poder ser un projecte que podria ser demanat per un client en concret, és un projecte acadèmic. Per tant la inversió de temps en recerca, pot desvirtuar una valoració econòmica d'un altre projecte similar desenvolupat per professionals del sector.

Els materials emprats per la consecució del projecte es poden veure en la següent taula:

Component	Quantitat	Preu unitat	Preu total
LPCXpresso LPC1769: microcontrolador amb Cortex-M3 ARM de NXP. També inclou Eclipse IDE de NXP	1	20.00 €	20.00 €
WiFly Rn-XV: és un mòdul Wi-Fi que incorpora un processador de 32 bits, 802.11b/g	1	26.00 €	26.00 €
CP2102: conversor USB a sèrie	1	4.00 €	4.00 €
Serial Controlled Motor Driver ROB-09571	1	25.80 €	25.80 €

Reles: RLG2R-2C-220	4	8.63 €	34.52 €
optoacobladors CNY 74 – 2	2	0.70 €	1.4 €
transistors BD 135	4	0.18 €	0.72 €
Resistències de 330 ohms	4	0.31 €	1.24 €
díodes 1N4007	4	0.19 €	0.76 €
Vehicle teledirigit on sols s'aprofita la carcassa i els motors	1	23.00 €	23.00 €
Cablejat	2	2.00 €	4.00 €
Piles 1.5 v	6	1.70 €	10,20 €
		TOTAL	151.64 €

Taula 2. Despeses de material

A continuació es detallen les despeses de mà d'obra de l'aplicació:

Mà d'obra	Quantitat	Preu unitat	Preu total
Estudis preliminars	20	30.00 €	600.00 €
Programació aplicació	250	30.00 €	7500.00 €
Muntatge i posta en marxa	25	30.00 €	750.00 €
Documentació	40	15.00 €	600.00 €
		TOTAL	9450.00 €

Taula 3. Despeses de mà d'obra

Les despeses totals de desenvolupament de l'aplicació:

Mà d'obra	Preu total
Despeses de material	151.64 €
Despeses de mà d'obra	9450.00 €
TOTAL	9601.64 €

Taula 4. Despesa total

8. Conclusions

Com a cloenda d'aquesta memòria, exposarem a continuació les conclusions que s'han extret de la realització d'aquest projecte.

8.1. Conclusions

Tal com s'ha comentat a l'inici de la memòria, es dividia el objectes en dos tipus. Per un costat els objectius més generals i educatius s'ha assolit d'una manera satisfactòria. Recordem que eren els següents:

- Introducció als sistemes encastats
- Entendre l'entorn de programació LPCXPRESSO
- Conèixer el funcionament dels sistemes interns de FreeRTOS
- Funcionament de la LPC 1769 amb un sistema operatiu en temps real
- Entendre el funcionament del mòdul WiFly

En quant als objectius específics d'aquest projecte també s'han assolit, no absents de certes dificultats. Cal destacar la importància de la fase preparatòria d'aquest projecte, on s'ha dedicat molt temps a la comprensió, preparació i desenvolupament de l'entorn i dotar d'eines útils per la consecució amb èxit del projecte.

Aquest objectius assolits específics del projecte han estat:

- Oferir una interfície via web on es permeti controlar el sistema
- Dotar al sistema d'una comunicació sense fils WiFi
- Dotar al sistema de la capacitat autònoma de connexió a la xarxa amb més cobertura
- Lectura periòdica de dades del servidor
- Creació de la plataforma del vehicle motoritzat
- Control del moviment del vehicle
- Dotar al sistema de la capacitat d'enviar dades al servidor

Cal recordar, que una de les principals dificultats ha estat l'averia del microcontrolador LPC en la fase final del desenvolupament del projecte. Això va obligar a adoptar mesures no previstes en el control del moviment del vehicle. Però, també gràcies a aquets esdeveniment, s'ha pogut desenvolupar i comparar dos sistemes bastant diferents entre si de control.

8.2. Proposta de millores

S'ha desenvolupat un prototip on els requisits mínims del projecte és que el sistema ha de ser capaç de rebre dades via del WiFi per anar endavant i endarrere. Però aquest mínim topant en una problemàtica important. Si es controla des d'un punt llunyà, on amb connexió a internet

podria ser qualsevol punt, l'usuari no podrà veure els obstacles en que es pot trobar aquest vehicle.

Per tant una millora obligada seria la instal·lació de d'una càmera de vídeo que pugui transmetre les imatges per on es mou el vehicle.

Encara que no tant efectiu com la proposta de la càmera, instal·lar sensors de proximitat ajudaria en el control del vehicle

Un altre punt que es podria millora en aquest projecte, és el control de l'estat de la font o fonts d'alimentació del sistema. D'aquesta manera, si l'usuari està informat, podria prendre mesures abans de perdre el control del vehicle.

Un punt fàcilment ampliable seria introduir més dades de control, com per exemple la potencia que utilitza la Wifly, el canal que utilitza etc. I al mateix temps que l'usuari pugui modificar-los.

8.3. Autoavaluació

Per començar voldria comentar la valoració molt positiva del fòrum de l'aula del TFC que, encara que érem sols 3 alumnes, la participació ha estat molt important en certes etapes del projecte, arribant a prop de 450 missatges al final del semestre.

Entrar en el mon dels sistemes encastats ha estat un descobriment de les immenses possibilitats que poden oferir, però no absentes de dificultats alhora de la seva programació, que en alguns moments ha arribat al límit de la desesperació, sobretot quant no saps els motius per la qual una part del codi no funciona.

Per programació del temps per desenvolupar tota l'assignatura del TFC, s'han dedicat molt temps en conèixer i comprendre l'entorn programació LPCXPresso i el funcionament dels sistemes interns de FreeRTOS. Aquest temps, molt necessari, ha limitat el temps de desenvolupament en si del projecte, cosa que crec que a provocat treballar a contrarellotge.

Finalment comentar que la valoració global ha estat molt enriquidora, tant a nivell de coneixement com a nivell humà.

8.4. Dificultats trobades

Varies han estat les dificultats trobades en tot el temps de realització del projecte. Primerament la poca pràctica en programació, ha obligat a moltes consultes per internet per solucionar temes que amb experiència es poden solucionar ràpidament.

Una de les dificultats trobades és el control de la memòria utilitzada. El FreeTOS requereix molta memòria RAM per crear tasques, cues i fer una crida de l'scheduler. La memòria restant s'ha d'administrar amb molta cura per no sobrepassar la memòria total disponible.

Una altre dificultat de ha requerit bastanta dedicació ha estat el control de motors. Inicialment es va provar amb un driver de motors. La dificultat era alimentar la placa amb prou potencia com per moure els motos. A continuació per problemes d'averia, es va haver de modificar el control passant a un sistema més manual de control, amb optoacobladors, transistors, relés, etc amb una inversió en temps dels càlculs dels diferents components.

9. Glossari

Driver : Un controlador de dispositiu és un programa que permet al sistema operatiu interactuar amb un perifèric.

Optoacobladors: és un dispositiu d'emissió i recepció que funciona com a interruptor activat mitjançant la llum emesa per un díode LED. S'utilitza per aïllar elèctricament dispositius molt sensibles.

RAM: la memòria d'accés aleatori, s'utilitza com a memòria de treball per els sistemes operatius, els programes i la majoria de software. És on es carreguen les instruccions que executarà el processador.

Relé: és un dispositiu electromecànic que funciona com un interruptor. Té l'avantatge de separar el circuit de la bobina del electroimant dels circuits que controla els contactes que activa.

RSSI: *Receive Signal Strength Indication*. És el terme que s'utilitza per mesurar el nivell de potencia del senyal rebut en les xarxes sense fil.

SAR: *Synthetic Aperture Radar*.

Scan: s'utilitza per a designar l'acció d'analitzar l'estat de les xarxes.

Scheduler: el planificador és un component funcional molt important dels sistemes operatius multitasca i multiprocés i és essencial en els sistemes operatius de temps real.

Servidor: és un ordinador que forma part d'una xarxa, que proveeix serveis a altres ordinadors denominats clients.

UART: *Universal Asynchronous Receiver Transmitter*. Controla els ports i dispositius sèrie. El UART agafa els bytes de dades i els transmet individualment de forma seqüencial. Un segon UART els re-ensambla. Es realitza mitjançant un únic cable, reduint el cost comparat en la transmissió en paral·lel.

WECA: *Wireless Ethernet Compatibility Alliance*, és una empresa creada en 1999 amb la finalitat de fomentar la compatibilitat entre tecnologies Ethernet sense fil.

Wi-Fi: és un mecanisme de connexió de dispositius electrònics d'una manera sense fil. Pot tenir un abast de 20 metres.

10. Bibliografia

<http://cv.uoc.edu/app/mediawiki14/wiki/IniciCortexM3>

<http://c.conclase.net/librerias/>

<http://es.wikipedia.org/wiki>

<http://www.arduino.cc/es/>

<http://www.nxp.com/>

http://es.wikipedia.org/wiki/Transistor_Darlington

Software:

<http://www.java.com/es/download/help/path.xml>

http://www.java.com/es/download/manual_v6.jsp

<http://www.eclipse.org/downloads/>

<http://lpcxpresso.code-red-tech.com/LPCXpresso/>

<http://www.putty.org/>

Motor:

<https://www.sparkfun.com/products/9571>

<http://www.clubse.com.ar/diego/notas/2/nota23.htm>

http://www.datasheetcatalog.org/datasheets2/79/79769_1.pdf

<http://www.fairchildsemi.com/ds/BD/BD135.pdf>

http://www.nosso.com.ar/spanish/tech_topics/relays.php

Datasheet

http://www.datasheetcatalog.org/datasheets2/79/79769_1.pdf (CNY74-2)

<http://www.datasheetcatalog.org/datasheet/philips/BD139-16.pdf> (BD135)

<http://descargas.cetronic.es/4052S.pdf> (Relé)

11. Annexos

11.1. Manual d'usuari

11.1.1. Instal·lació, compilació i execució del codi de la LPC

La instal·lació de tot el programari necessari per treballar amb el microcontrolador és senzilla, l'únic que cal fer és registrar-se i instal·lar el programa que es pot descarregar en la següent pàgina:

<http://lpcxpresso.code-red-tech.com/LPCXpresso/>

per qualsevol dubte es pot consulta la guia d'instal·lació proporcionada per LPCXpresso:

http://www.nxp.com/documents/other/LPCXpresso_Getting_Started_Guide.pdf

En el compte de correu utilitzat en el registre, LPCXpresso ens envia el *username* i el *password* per activar el producte, el qual s'activarà en *Product activation* de la pestanya *HELP*.

Un cop el programa està instal·lat descarreguem i descomprimint el fitxer del codi font. A continuació s'ha de canviar l'espai de treball actual pel del projecte que és LPCpresso. Es pot fer mitjançant *Switch Workspace* de la pestanya *File*.

Si el resultat de la instal·lació és correcte, s'hauria de visualitzar tal com es veu en la següent figura:

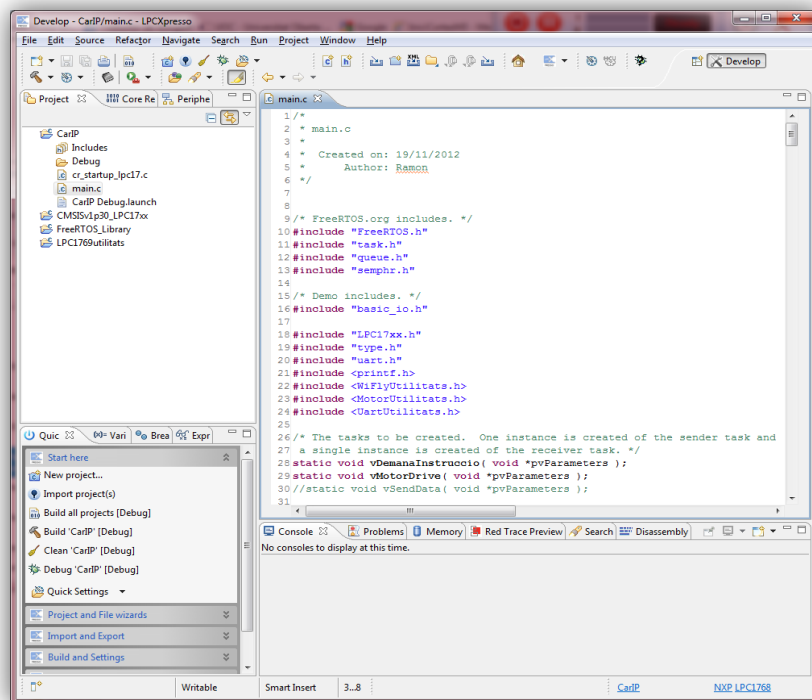



Figura 23. Pantalla de treball de LPCXpresso

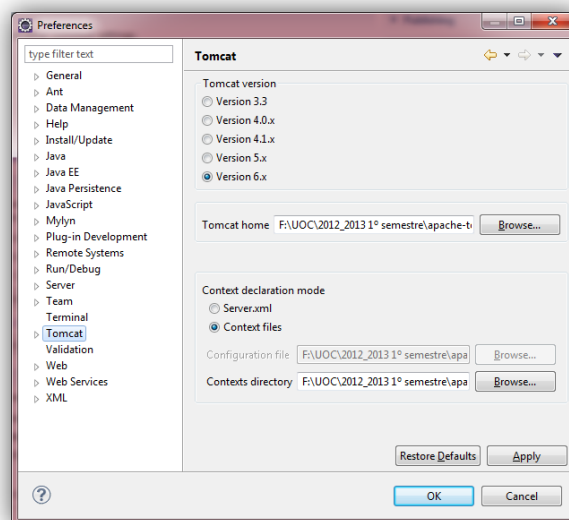
Un cop connectada la LPC1769 podrem s'ha de debugar tots els projectes i instal·lar el programa, mitjançant *Program Flash* representat per l'Icona  , que es pot trobar en dins LPCXpresso\CarIP\Debug\CarIP.axf

11.1.2. Instal·lació, compilació i execució del codi del Servidor

Per instal·lar el servidor, s'han de descarregar i instal·lar per aquest ordre els següents programes:

- Descarregar i instal·lar Java Runtime Environment (JRE) 6 : http://www.java.com/es/download/manual_v6.jsp
- Descarregar i instal·lar Eclipse IDE for java EE Developers: <http://www.eclipse.org/downloads/>
- Descarregar i instal·lar Apache-tomcat 6.0.36: <http://tomcat.apache.org/download-60.cgi>

configurar l'Apache-Tomcat per que es pugui executar des de Eclipse,



descarregar projecte i obrir el codi font i importar les lliberies a l'entorn Eclipse.

11.2. Imatges del vehicle

A continuació mostrem detalls del vehicle. Primerament tal com es pot veure es mostre el vehicle amb només el xassís, els dos motors i les sortides de 6 i 9 volts que es poden obtenir del porta bateries de 6 pès que te incorporat el vehicle.

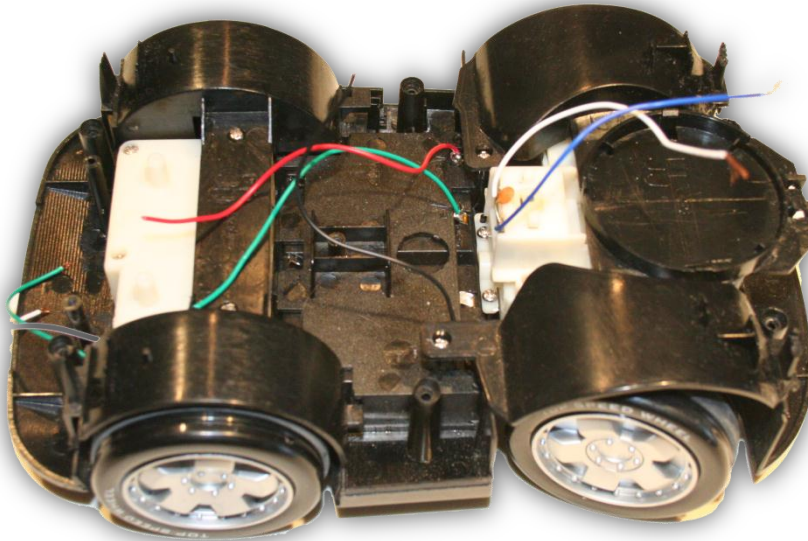


Figura 24. Xassís vehicle

A continuació podem veure el muntatge de control de motors format per els relés, optoacoblador, transistors, díode i resistència. En el vehicle, hi haurà un muntatge per cada motor.

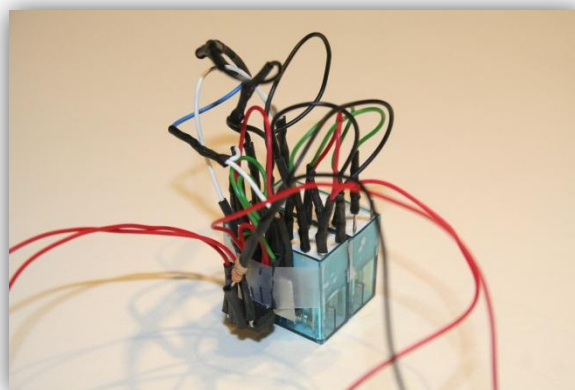


Figura 25. Muntatge control motors

Finalment, en la següent figura podem observar el vehicle amb tots els components incorporats

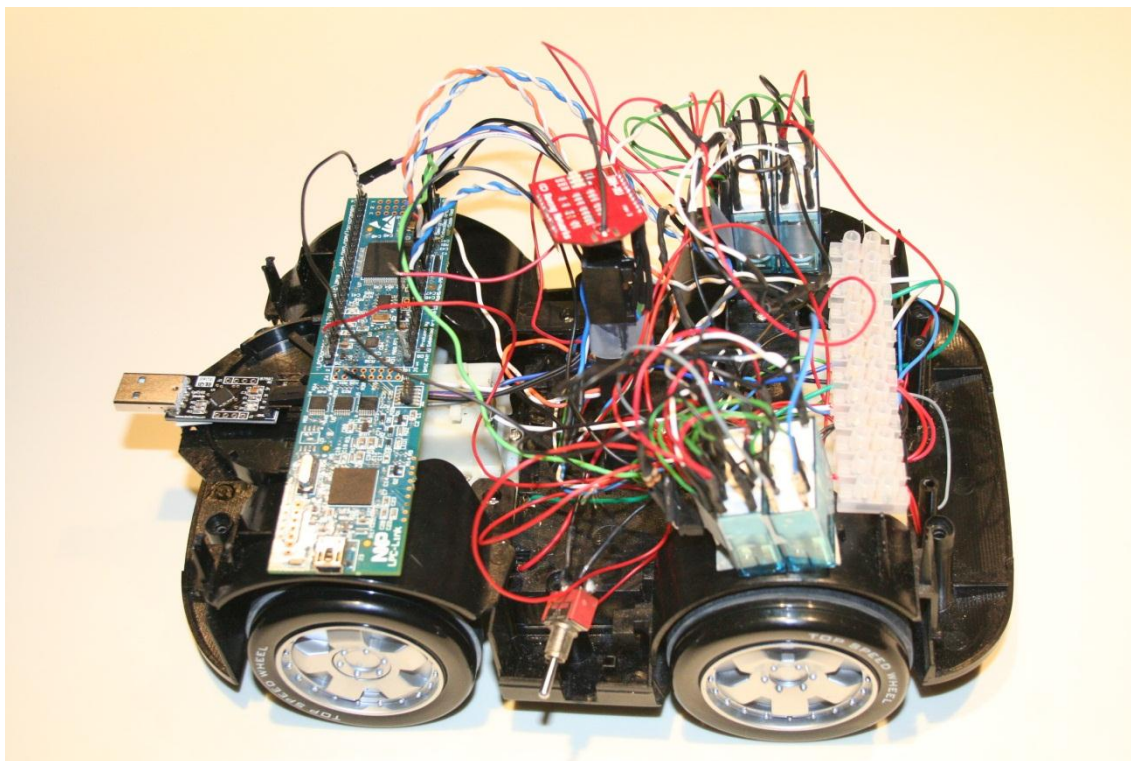


Figura 26. Vehicle IP