



Disseny i implementació d'un robot amb connexió WiFi

Cèsar Guerrero Miralles
Grau Enginyeria Informàtica

Jordi Bécares Ferrés
Consultor

15 de gener de 2013

***A Javier i Cris, companys de camí i mestres.
I a tots els éssers sensibles.***

Copyright © 2013 Cèsar Guerrero Miralles

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

A copy of the license is included in the section entitled "GNU Free Documentation License".

FITXA DEL TREBALL FINAL

Títol del treball:	Disseny i implementació d'un sistema encastat mòbil amb connexió WiFi controlat a través d'Internet
Nom de l'autor:	Cèsar Guerrero Miralles
Nom del consultor:	Jordi Bécares Ferrés
Data de lliurament (mm/aaaa):	01/2013
Àrea del Treball Final:	Sistemes Encastats
Titulació:	Grau Enginyeria Informàtica
Resum del Treball (màxim 250 paraules):	
<p>En el present treball s'ha dissenyat i portat a terme un prototipus de sistema encastat mòbil amb connectivitat de xarxa sense fils i accés a Internet capaç de rebre ordres des d'un navegador web.</p> <p>El nucli del sistema el constitueix una placa de desenvolupament LPCXpresso LPC1769 de NXP Semiconductors basada en el microprocessador ARM Cortex-M3. Per a donar-li connectivitat s'ha fet servir una mòdul de xarxa RN-171-XV de Roving Networks que es comunica amb la placa principal mitjançant la interfície UART.</p> <p>Per a implementar la mobilitat s'han aplicat esquemes típics de robòtica recreativa, utilitzant un xassís amb dues rodes accionades de manera independent per dos servomotors modificats per proporcionar moviment continu i controlats per les sortides PWM de la placa mare. D'aquesta manera, a part del moviment i cerca de cobertura de xarxa en una dimensió que ens demana la especificació mínima del sistema, podem estendre aquestes funcionalitats a dues dimensions.</p> <p>El programari del sistema s'ha dut a terme en l'entorn de desenvolupament proporcionat pel fabricant, que consisteix en una adaptació de l'IDE Eclipse utilitzant llenguatge C. S'ha seguit un model d'aplicatiu productor-consumidor aprofitant la funcionalitat multitasca proporcionada pel kernel FreeRTOS.</p>	

Abstract (in English, 250 words or less):

In the present work the prototype for a self-propelled embedded system with wireless connectivity has been carried out. The system is designed to receive commands from the Internet.

The core of the system is a LPCXpresso LPC1769 development board from NXP Semiconductors, which is based on the ARM Cortex-M3 microprocessor. Wireless networking capability is provided by a RN-171-XV module from Roving Networks, which communicates with the main board via the UART interface.

In order to achieve mobility, recreational robotics schemes have been applied, in the form of a two-wheeled aluminum sheath body. Both wheels are independent and are powered by modified servos that can provide continuous motion. Control signals are sent through the PWM outputs of the main board. With this design, it is possible to go beyond the basic specification of the project and provide two-dimensional mobility and access point searching.

Software for the system has been developed under the LPCXpresso IDE in C language, following a producer-consumer model. The necessary multitasking capability is provided by the FreeRTOS kernel.

Paraules clau (entre 4 i 8):

Sistemes encastats; robòtica; microcontroladors; mota

Índex

1. Introducció	8
1.1 Context i justificació del Treball	8
1.2 Objectius del Treball	8
1.3 Enfocament i mètode seguit	10
1.4 Planificació del projecte	11
1.4.1 Descripció en detall de les tasques.	11
1.4.2 Fites del projecte.	14
1.4.3 Planificació inicial.....	15
1.4.4 Revisió final de la planificació.	17
1.5 Recursos emprats	20
1.5 Breu sumari de productes obtinguts.....	21
1.6 Breu descripció dels altres capítols de la memòria	22
2. Antecedents.....	23
3. Descripció física	25
3.1 Descripció mecànica.....	25
3.1.1 Configuració	25
3.1.2 Motors	26
3.1.3 Xassís.....	27
4. Descripció funcional	34
4.1 Sistema total.....	34
4.2 Disseny de l'aplicació.....	36
5. Descripció detallada	38
5.1 Capa d'aplicació.	38
5.2 Capa d'entrada/sortida.....	40
5.2.1 Llibreria printfLib.	40
5.2.2 Llibreria movementLib/fitxer pwm.c	40
5.3 Capa de subsistemes.	41
5.3.1 Llibreria wiflyLib	41
5.3.2 Llibreria movementLib/fitxer movement.c	41
5.4 Capa de gestió de comandes.	46
5.4.1 Llibreria commandLib.	46
6. Viabilitat tècnica	49
7. Valoració econòmica	50
8. Conclusions	51
9. Glossari.....	54
10. Bibliografia	55

Llista de figures

Figura 1. Calendari de planificació inicial del projecte.	15
Figura 2. Diagrama de Gantt de la planificació inicial del projecte.	16
Figura 3. Calendari de la planificació final del projecte.	17
Figura 4. Diagrama de Gantt de la planificació final del projecte.	19
Figura 5. Resum de les possibilitats de moviment d'un robot amb dues rodes i dos motors: a) moviment unidimensional; b) girar cap a un costat; c) gir ràpid sobre el seu eix.	26
Figura 6. Esquema del xassís del robot MarkIII de la PORTS, utilitzat en aquest treball	29
Figura 7. Esquema del robot Rhyno-1, basat en el xassís Mark III.	30
Figura 8. Connexionat elèctric del sistema	32
Figura 9. Diagrama de blocs de l'aplicació.	34
Figura 10. Estructura de capes de l'aplicació.	37
Figura 11. Diagrama de flux de l'aplicació.	39
Figura 12. Diagrama de flux de la cerca de cobertura en una dimensió.	43
Figura 13. Diagrama de flux de la cerca de cobertura en dues dimensions.	45

1. Introducció

1.1 Context i justificació del Treball

En aquest projecte es proporcionarà al client, Consultors de la UOC S.A., el programari necessari per al control des d'Internet via wifi d'un petit robot basat en la mota LPC1769. Caldrà cobrir totes les etapes del cicle de vida: planificació, anàlisi, disseny i prova.

El context del projecte és la voluntat del client d'ampliar el seu àmbit de negoci centrat actualment en sistemes encastats industrials basats principalment en motes LPCXpresso de NXC. El client ens subcontracta per a que li generem el coneixement per estendre la seva activitat a l'àmbit de la robòtica, amb possibles aplicacions en entorns industrials perillosos, assistència a persones grans, vigilància i d'altres.

Del sistema es requereix per una banda que sigui capaç de rebre ordres de moviment endavant i endarrere des de Internet a través d'una xarxa sense fils wifi, i per altra que disposi d'un mecanisme de cerca automàtica del punt amb millor cobertura de xarxa.

1.2 Objectius del Treball

El treball final de carrera (TFG) té com a objectiu materialitzar els coneixements adquirits durant la carrera, en aquest cas El Grau en Enginyeria Informàtica, en la consecució d'un projecte real en totes les seves fases, des de l'Anàlisi previ i presa de requeriments fins a la prova i posada en marxa. Per raons evidents, la fase de manteniment n'ha de quedar al marge.

Aquest TFG se situa en l'itinerari d'Enginyeria de Computadors i més específicament en l'àmbit dels sistemes encastats. A més dels conceptes propis de l'assignatura del mateix nom, també s'aprofundirà en allò après a Sistemes Operatius, Xarxes i Disseny de Sistemes Operatius: programació en llenguatge C, ús de les funcions del sistema operatiu, sistemes operatius en temps real, ús i gestió de tasques.

Pel que fa als objectius del sistema desenvolupat, es resumeixen en els punts següents:

Objectius bàsics:

(a) Sistema de moviment endavant i enredera:

El sistema ha de poder executar comandes de moviment en una dimensió, és a dir, avançar, retrocedir i aturar-se.

(b) Connexió via Wifi:

El sistema ha de poder enllaçar amb una xarxa sense fils oberta o amb seguretat WEP o WPA.

(c) Control remot des de Internet:

El sistema ha de poder ser controlat des de qualsevol dispositiu connectat a Internet per mitjà d'un simple navegador web.

(d) Recepció de comandes de control:

El sistema ha d'incorporar un servidor web que permanent escolti i interpreti les comandes enviades per mitjà de peticions http.

(e) Sistema automàtic de cerca de cobertura:

El sistema ha de ser capaç de verificar en nivell de cobertura de xarxa de que disposa i moure's a una posició millor mitjançant un algorisme que utilitzi les comandes de posicionament.

Objectius de valor afegit:

(f) Sistema de moviment en dues dimensions:

S'implementaran funcions de gir que permetin al prototipus moure's en el pla.

(g) Sistema automàtica de cerca de cobertura en dues dimensions:

El sistema ha de ser capaç de trobar la direcció de millor cobertura i moure-s'hi.

1.3 Enfocament i mètode seguit

Les principals restriccions que han condicionat la posada en pràctica del projecte han estat dues: Per una banda una limitació en el temps molt ajustada, i per l'altre el volum de feina desenvolupada abans que els requeriments finals del projectes fossin enunciats, i que, en gran mesura es podia reaprofitar per assolir-los. La suma d'aquests dos factors determina a la pràctica la inevitabilitat de descartar tota opció que no impliqui d'utilitzar el maquinari i programari subministrat, sense disposar de la possibilitat real d'entrar a valorar si les seves característiques són les que millor s'adapten a la tasca demanada.

No obstant això, la plataforma utilitzada té un punt molt a favor en aquest context, i és l'ampli suport del fabricant i l'abundant disponibilitat de codi d'exemple per a múltiples aplicacions. Tot plegat desemboca en un enfocament dirigit a reaprofitar codi fins allà on sigui possible, adaptant-lo i optimitzant-lo per a la nostra aplicació només fins al punt on la ineludible limitació temporal ho permetés.

Pel que fa a la vessant purament mecànica del projecte, els plantejaments són anàlegs. Si bé en la web no es troben exemples de sistemes encastats autopropulsats basats en plaques LPCXpresso, sí que hi ha tot un univers de referències disponibles a pàgines, fòrums, blogs i botigues dedicats

a l'àmbit de la robòtica recreativa, entesa com l'afició de muntar i programar petits robots de propòsit divers. [1][2]

Aquests robots es basen habitualment en microcontroladors amb característiques inferiors o similars al que disposem per al projecte, notablement la popular família Arduino[3]. Les solucions de programari emprades en aquests robots no són aplicables al nostre cas, però sí que ho són les configuracions de xassis i les maneres d'utilitzar els motors per propulsar-los. Novament, la possibilitat d'emprar solucions ja comprovades ha permès de reduir notablement el temps de desenvolupament.

1.4 Planificació del projecte

1.4.1 Descripció en detall de les tasques.

T0. Instal·lació del programari. S'instal·la l'entorn LPCXpresso, PuTTY i Microsoft Office Project en les màquines que utilitzarem.

T1. Prova del maquinari. Es connecten un per un al PC cadascun dels elements de maquinari i se'n comprova el funcionament.

T1a. Prova de la mota. Es carreguen i executen projectes d'exemple.

T1b. Prova del mòdul de xarxa sense fils. Es comprova que respon a comandes i es configura per accedir a Internet.

T2. Redacció de l'informe d'instal·lació. Es sintetitzen tots els resultats anteriors per a l'entrega de la PAC1.

T3. Connexió del maquinari. Es connecta la mota per una banda al mòdul de xarxa i per a l'altra al pc per a depuració.

T4. Driver de comunicacions. Es desenvolupen les capes de llibreria necessàries per a comunicar la mota amb Internet.

T4a. Driver printf. Llibreria per comunicar la mota amb el mòdul de xarxa per port UART..

T4b. Driver wifly. Llibreria que utilitza l'anterior per enviar al mòdul de xarxa comandes de configuració i de connexió a Internet.

T5. Redacció de l'informe del driver de comunicacions. Es sintetitzen tots els resultats anteriors per a l'entrega de la PAC2.

T6. Aplicatiu productor-consumidor. Es desenvolupa un aplicatiu multitasca basat en aquest paradigma com a formació prèvia de cara a fases posteriors del projecte.

T7. Redacció de l'informe de l'aplicatiu productor-consumidor. Es sintetitzen tots els resultats anteriors per a l'entrega de la PAC3.

T8. Identificació de tasques. Es divideix el projecte en una sèrie de tasques.

T9. Determinació de Fites del projecte i Recursos. S'avaluen aquests conceptes, necessaris per a la PAC4.

T10. Planificació temporal. S'estableix el calendari d'execució de cada tasca.

T11. Redacció del Pla de Treball. Es sintetitzen tots els resultats anteriors per a l'entrega de la PAC4.

T12. Anàlisi de requeriments. Detall de les funcions que ha de realitzar el sistema.

T12a. Requeriments funcionals. Es recullen les necessitats d'informació del client.

T12b. Requeriments no funcionals. Es defineixen criteris per avaluar l'operació del sistema.

T13. Disseny físic del sistema.

T13a. Disseny mecànic. Tria de la configuració i components: xassís, motors, rodes, eixos.

T13b. Disseny elèctric. Establir els mecanismes per a controlar els motors utilitzant les sortides PWM i/o GPIO de la mota.

T14. Adquisició del material.

T15. Muntatge del sistema.

T15a. Muntatge mecànic. Encaix de les peces adquirides i fabricació de les peces complementàries que siguin necessàries.

T15b. Connexionat elèctric. Circuits d'alimentació i control dels motors.

T16. Prova de control. Desenvolupament d'un programa de prova que comprovi la resposta dels motors als senyals de control.

T17. Llibreria de moviment del robot. Primera versió de la llibreria amb comandes per anar endavant i endarrere.

T18. Sistema de cerca de màxima cobertura. Primera versió de la funció que utilitzi la llibreria de mòdul de xarxa per mesurar la cobertura, i el moviment en una dimensió per maximitzar-la.

T20. Llibreria de captura d'ordres. Funcions per a capturar d'Internet ordres per al robot.

T21. Sistema generador d'ordres. Desenvolupament d'un sistema que, des d'un PC remot, generi comandes perquè el robot les executi.

T22. Aplicatiu de control. A partir de les llibreries creades, desenvolupar un aplicatiu multitasca que controli el robot:

T22a. Tasca productora: Captura i encua ordres d'Internet.

T22b. Tasca consumidora: Agafa i executa ordres de la cua.

T23. Proves. Crear i executar el joc de proves. Depurar errors que hagin restat inadvertits en les fases de depuració dels components individuals.

T24. Llibreria de moviment del robot. (Opcional, si el temps ho permet). Segona versió de la llibreria amb comandes per girar, permetent el moviment en 2D.

T25. Sistema de cerca de màxima cobertura. (Opcional, si el temps ho permet). Segona versió de la llibreria funció que utilitzi el moviment en dues dimensions per maximitzar la cobertura.

T26. Redacció de la documentació del programari. Es sintetitzen tots els resultats anteriors per a l'entrega de la PAC5.

T27. Redacció de la Memòria. Es completa el redactat de la memòria final per a l'entrega de la PAC6.

T28. Presentació virtual. Es prepara la presentació PowerPoint que resumirà el projecte per a l'entrega de la PAC7.

1.4.2 Fites del projecte.

Les dates clau que ha calgut tenir en compte per al desenvolupament del projecte són les següents:

DATA	NOM DE LA FITA	DESCRIPCIÓ
19/09/12	Inici del curs	Data de començament del semestre acadèmic.
09/10/12	PAC1	Lliurament de l'informe d'instal·lació i prova de l'entorn de treball i el maquinari.
06/11/12	PAC2	Lliurament del driver de comunicacions.
25/11/12	PAC3	Lliurament de l'exemple d'aplicatiu productor-consumidor.
01/12/12	PAC4	Lliurament de la planificació del projecte.
02/01/12	PAC5	Lliurament del programari complet del projecte
15/01/13	PAC6	Lliurament de la memòria del projecte
19/01/13	PAC7	Lliurament de la presentació del projecte

1.4.3 Planificació inicial.

Calendari.







		Nombre de tarea	Duración	Comienzo	Fin
1	✓	Inici semestre	0 días	mié 19/09/12	mié 19/09/12
2	✓	Instal·lació del programari.	1 día	jue 27/09/12	jue 27/09/12
3	✓	Prova del maquinari	24,5 días	vie 28/09/12	mar 09/10/12
4	✓	Lliurament PAC1	0 días	mar 09/10/12	mar 09/10/12
5	✓	Driver de comunicacions	27 días	jue 11/10/12	mar 06/11/12
6	✓	Lliurament PAC2	0 días	mar 06/11/12	mar 06/11/12
7	✓	Aplicatiu productor-consumidor	13 días	mar 13/11/12	dom 25/11/12
8	✓	Lliurament PAC3	0 días	dom 25/11/12	dom 25/11/12
9	✓	Pla de treball	6 días	lun 26/11/12	sáb 01/12/12
10	✓	Lliurament PAC4	0 días	sáb 01/12/12	sáb 01/12/12
11		Adquisició del material	6 días	lun 03/12/12	sáb 08/12/12
12		Muntatge del sistema	5 días	dom 09/12/12	jue 13/12/12
13		Libreria de moviment	3 días	vie 14/12/12	dom 16/12/12
14		Sistema de cerca de cobertura	2 días	lun 17/12/12	mar 18/12/12
15		Sistema de comandes per Internet	6 días	jue 06/12/12	mar 11/12/12
16		Aplicatiu de control	3 días	mié 19/12/12	vie 21/12/12
17		Proves	5 días	sáb 22/12/12	vie 28/12/12
18		Ampliació del sistema a moviment 2D	3 días	sáb 29/12/12	lun 31/12/12
19		Lliurament PAC5	0 días	mié 02/01/13	mié 02/01/13
20		Memòria	12,5 días	jue 03/01/13	mar 15/01/13
21		Lliurament PAC6	0 días	mar 15/01/13	mar 15/01/13
22		Presentació	4 días	mar 15/01/13	sáb 19/01/13
23		Lliurament PAC7	0 días	sáb 19/01/13	sáb 19/01/13

Figura 1. Calendari de planificació inicial del projecte.

Diagrama de Gantt.

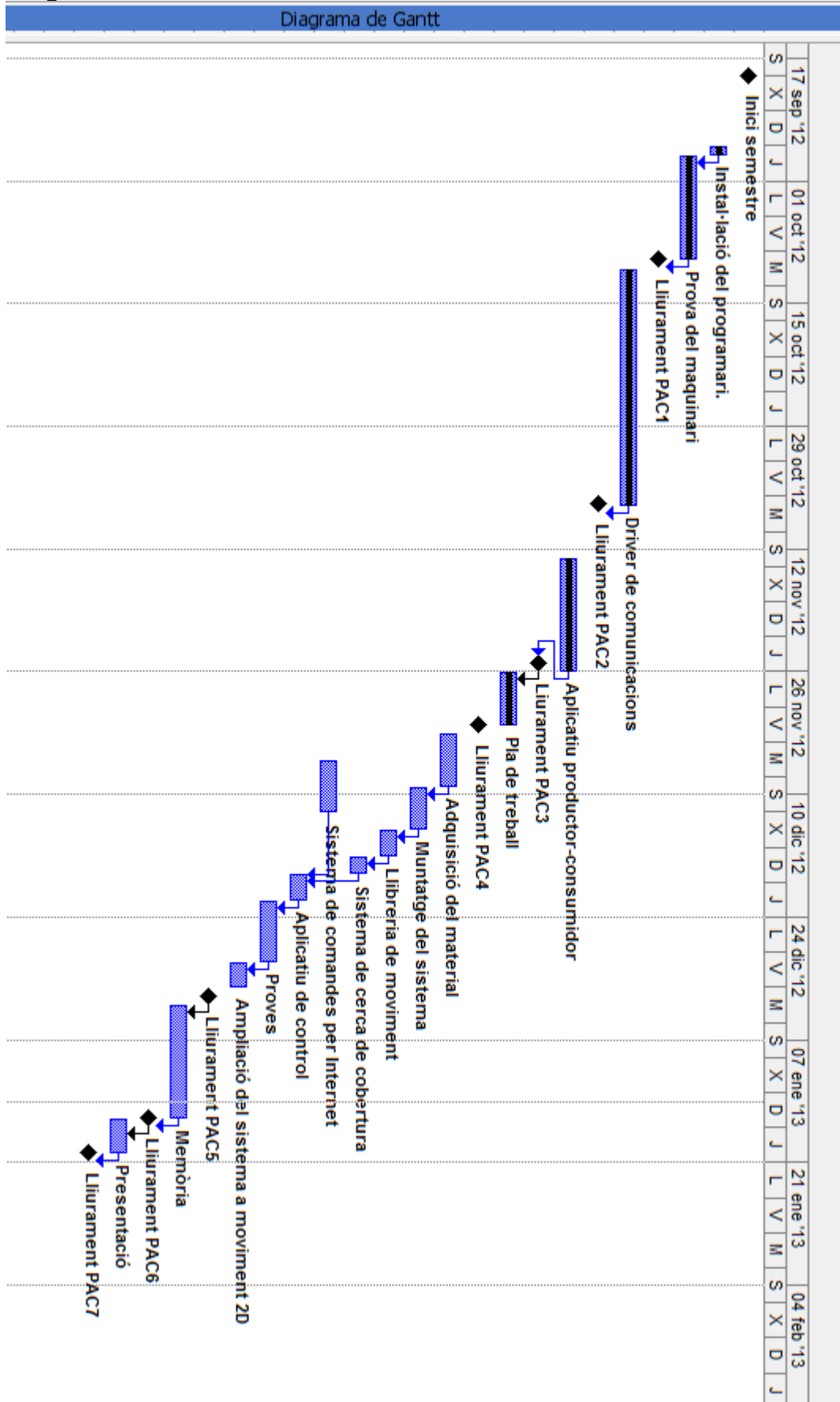


Figura 2. Diagrama de Gantt de la planificació inicial del projecte.

1.4.4 Revisió final de la planificació.

Calendari.

	1	Nombre de tarea	Duración	Comienzo	Fin	
Diagrama de Gantt	1	✓	Inici semestre	0 días	mié 19/09/12	mié 19/09/12
	2	✓	Instal·lació del programari.	1 día	jue 27/09/12	jue 27/09/12
	3	✓	Prova del maquinari	24,5 días	vie 28/09/12	mar 09/10/12
	4	✓	Lliurament PAC1	0 días	mar 09/10/12	mar 09/10/12
	5	✓	Driver de comunicacions	27 días	jue 11/10/12	mar 06/11/12
	6	✓	Lliurament PAC2	0 días	mar 06/11/12	mar 06/11/12
	7	✓	Aplicatiu productor-consumidor	13 días	mar 13/11/12	dom 25/11/12
	8	✓	Lliurament PAC3	0 días	dom 25/11/12	dom 25/11/12
	9	✓	Pla de treball	6 días	lun 26/11/12	sáb 01/12/12
	10	✓	Lliurament PAC4	0 días	sáb 01/12/12	sáb 01/12/12
	11	✓	Adquisició del material	9 días	lun 03/12/12	mar 11/12/12
	12	✓	Muntatge del sistema	6 días	vie 07/12/12	mié 12/12/12
	13	✓	Llibreria de moviment	5 días	vie 14/12/12	mar 18/12/12
	14	✓	Sistema de cerca de cobertura	2 días	mié 19/12/12	jue 20/12/12
	15	✓	Sistema de comandes per Internet	6 días	jue 06/12/12	mar 11/12/12
	16	✓	Aplicatiu de control	2 días	vie 21/12/12	sáb 22/12/12
	17	✓	Proves	6 días	mié 26/12/12	lun 31/12/12
	18	✓	Ampliació del sistema a moviment 2D	3 días	lun 31/12/12	mié 02/01/13
	19	✓	Lliurament PAC5	0 días	vie 04/01/13	vie 04/01/13
	20	✓	Memòria	10,5 días	sáb 05/01/13	mar 15/01/13
	21	✓	Lliurament PAC6	0 días	mar 15/01/13	mar 15/01/13
	22		Presentació	4 días	mar 15/01/13	sáb 19/01/13
	23		Lliurament PAC7	0 días	sáb 19/01/13	sáb 19/01/13

Figura 3. Calendari de la planificació final del projecte.

L'execució del calendari planificat ha estat en general altament satisfactori, amb desviacions que en cap cas han anat més enllà raonables d'un o dos dies. Entre aquests ajustaments menors podem mencionar:

- En la pràctica la fase d'adquisició de material s'ha solapat amb el muntatge, quan s'ha hagut de buscar alternatives per peces que no acabaven de fer les funcions que els hi estaven destinades.

- El desenvolupament de la llibreria de moviment es va allargar en dos dies pels problemes amb la utilització de les sortides PWM que es detallen en un capítol posterior.

- Per contra, la preparació de l'aplicatiu de control, un cop disponibles els subsistemes, va ser més breu de planificat. Cal explicar, però, que en la fase de proves es va fer una profunda reestructuració de l'aplicatiu coincidint amb les millores que es van mostrar necessàries per millorar la resposta el sistema.

Diagrama de Gantt.

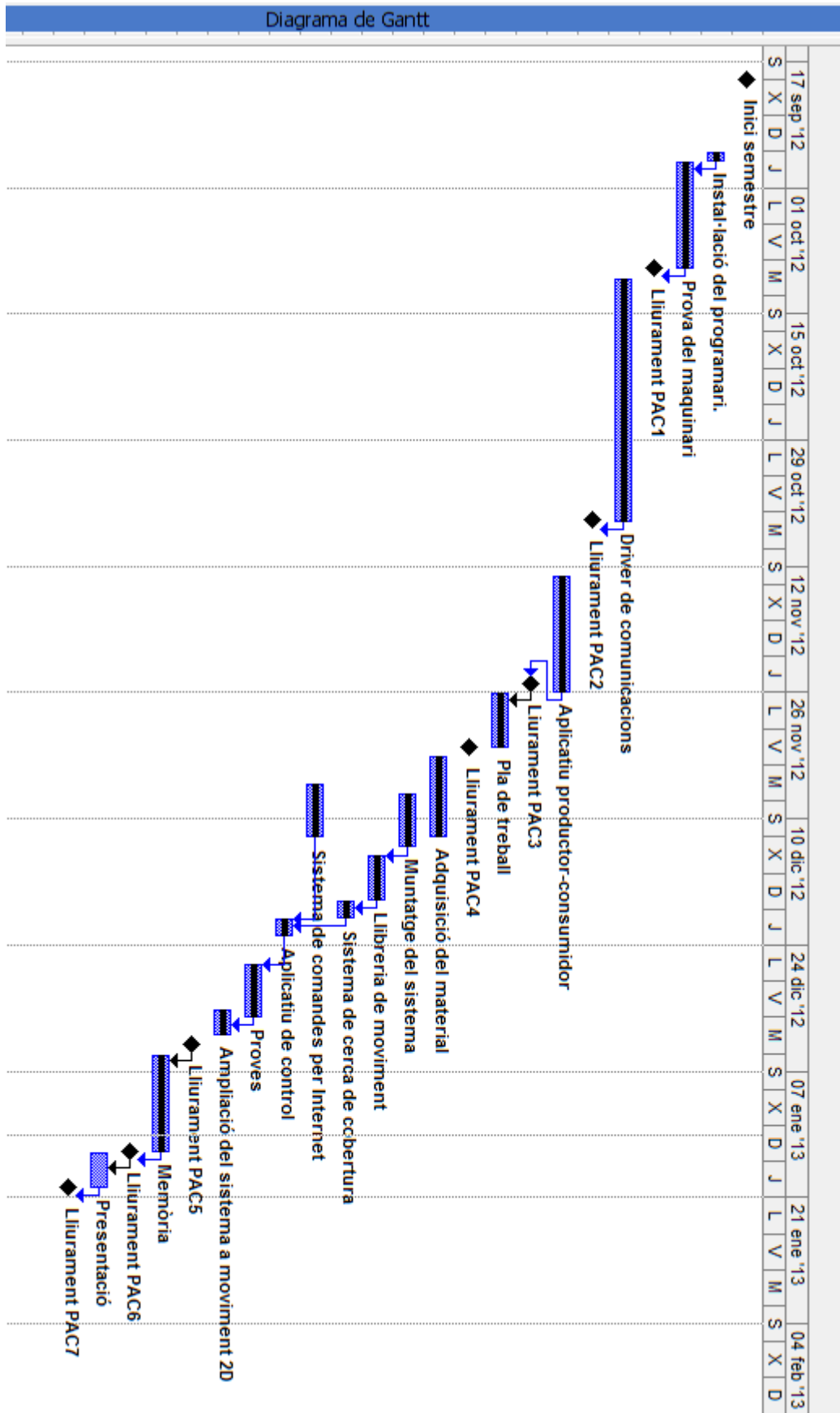


Figura 4. Diagrama de Gantt de la planificació final del projecte.

1.5 Recursos emprats

Recursos de maquinari.

MÀQUINA	TIPUS	CPU	RAM	SISTEMA OPERATIU
Servidor Principal	Toshiba Satellite A660	Intel Core i7 720QM	4 GB	Windows7 Professional 64bit SP1
Servidor Secundari	PC	Intel Q6600	4 GB	Windows7 Professional 32bit SP1
Mota	LPC1769	ARM Cortex-M3	64KB	FreeRTOS 6.1.1
Mòdul de xarxa	RN-171-XV	RN-171	128KB	Wifly-GSX 2.32

Recursos de programari.

PROGRAMARI	SUPORT
LPCXpresso 4.30.0_1023	Web
MS Office 2003, Visio i Project	DVD-Rom
PuTTY 0.62	Web

Recursos bibliogràfics.

RECURS	SUPORT
Material de l'assignatura	Web
Enllaços del pla docent	Web
Fòrums de LPCXpresso	Web

Recursos humans.

RECURS	CATEGORIA	TIPUS	DISPONIBILITAT (hores/setmana)	COS T (euros/hora)
Cèsar Guerrero Miralles	Analista/Programador	Ordinari	10	35
Cèsar Guerrero Miralles	Analista/Programador	Extraordinari	45	70

1.5 Breu sumari de productes obtinguts

La consecució dels objectius s'ha materialitzat en l'entrega del programari del producte, juntament amb la present memòria i una presentació que resumirà el treball realitzat, i que inclourà una demostració en vídeo de les funcionalitats del robot desenvolupat. Previ a l'entrega final s'han realitzat cinc entregues parcials:

-PAC1: Informe de la instal·lació i prova del programari de desenvolupament, així com de la part de maquinari referida a la mota LPC1769 i al mòdul de xarxa sense fils WiFly.

-PAC2: driver per gestionar les comunicacions entre la mota i el mòdul de xarxa per permetre l'accés a Internet del sistema.

-PAC3: prova de sistema productor-consumidor com a prototipus d'aplicació multitasca sobre el maquinari LPC1769+WiFly.

-PAC4: pla de treball.

-PAC5: programari del robot controlat per Internet i amb funció de cerca de cobertura de xarxa.

-PAC6: memòria del projecte.

-PAC7: presentació del projecte.

1.6 Breu descripció dels altres capítols de la memòria

El capítol 2 descriu els antecedents del projecte en el doble context dels sistemes encastats i la robòtica recreativa.

A continuació, en el capítol 3 es comença a descriure el sistema desenvolupat començant pels seus elements físics, per a continuar en els capítols 4 i 5 amb els aspectes de programari, d'una manera més general en el primer i més detallada en el segon. Aquest tres capítols constitueixen el gruix dels continguts del treball.

En el capítol 6 es fa una petita avaluació de la viabilitat del projecte, per després passar, en el capítol 7, a presentar una succinta valoració econòmica.

Les conclusions del projecte es presenten en el capítol 8, per tancar amb el glossari (9) i la bibliografia.

2. Antecedents

Aquest projecte s'ha d'entendre com d'interès principalment didàctic en el camí de la formació de l'enginyer, amb la missió que aprengui a afrontar la consecució d'un projecte de principi a final i a cercar solucions als problemes que inevitablement sorgeixen, dintre de les restriccions existents. No és un objectiu primordial, doncs, obtenir com a resultat solucions tècniques revolucionàries o amb un potencial de negoci únic, o un producte que estigui remotament proper de convertir-se en una solució comercialitzable.

La consideració anterior és rellevant perquè pel tipus de problemes que s'han resolt en el projecte ja hi ha solucions comercialitzades, eficients i a molt baix preu. Parlem, per exemple, dels helicòpters teledirigits des del telèfon mòbil que es poden adquirir per poques desenes de euros.[4] Afegir-hi les funcionalitats de cerca de cobertura seria purament una qüestió de programari no gaire complexa que podria requerir, com a molt, un microcontrolador més potent.

En el context dels sistemes encastats, el projecte té un posicionament relativament atípic, en el sentit que no aprofundeix en el concepte tan actual de les xarxes de sensors sense fils. Tot i que el mateix mòdul de comunicacions WiFly ja té per si sol les característiques necessàries per esdevenir una mota en aquest tipus d'aplicacions -per potència de processament, memòria i capacitats d'entrada i sortida digitals i analògiques-, en el present projecte serveix principalment com a interfície de comunicacions per al mòdul LPC1769 que esdevé el cervell del sistema.

Es tracta, doncs, més d'un sistema encastat en el sentit tradicional d'un sistema autònom amb una finalitat concreta gestionat per un petit ordinador. Però, quan aquests sistemes incorporen el concepte d'automobilitat entren ja en el camp d'estudi de la robòtica, que a tots els nivells, des de l'acadèmic fins al comercial és l'interès d'uns altres professionals. En aquest sentit, cal no

menystenir la contribució provinent del creixent àmbit de la robòtica recreativa com a font d'idees i solucions provades en un entorn competitiu en forma dels populars tornejos de sumo, partits de futbol, curses de velocitat.

Potser el punt més interessant d'aquest treball és prendre consciència de les possibilitats que aquest solapament d'interessos ofereix.

Com es dedueix del comentat més amunt, un sistema amb només les capacitats incorporades en aquests treball sembla insuficient per merèixer un interès comercial immediat. Des de cert punt de vista, es tractaria, al cap i a la fi, d'un 'robot cec' capaç de moure's, però no de conèixer la seva posició o la dels obstacles que l'envolten. Però, un cop superada aquesta limitació, cosa no molt complicada mitjançant sensors de proximitat per ultrasons, o inclús brúixoles [5], les capacitats de la plataforma utilitzada com a base de sensors podrien ser útils, per exemple, per a la producció de robots de monitorització de les condicions, en entorns perillosos.

3. Descripció física

3.1 Descripció mecànica

3.1.1 Configuració

Tal i com s'ha explicat en apartats anteriors, el problema de donar mobilitat en el pla a petits sistemes de baix cost basats en un microcontrolador ha estat perfeccionat durant anys en l'entorn de la robòtica recreativa. Enfrontats a una problemàtica similar, sembla per tant del tot raonable remetre's a les idees que en aquest camp s'han demostrat efectives.

Una configuració simple, i a la vegada molt popular per al disseny de petits robots de propòsit general, és la conformada per una plataforma amb dos motors i dues rodes, més un tercer element de suport que pot ser una pota, acabada o no en un rodament de bola. Aquesta configuració permet implementar la capacitat de gir d'una manera molt més senzilla que un disseny amb quatre rodes, en el que caldria introduir un mecanisme de direcció en almenys un dels eixos. Pel contrari, en la configuració amb dues rodes i dos motors, fer avançar o retrocedir el robot implica accionar els dos motors en el mateix sentit. Per girar cap a un costat, només cal detenir la roda d'aquell costat i fer avançar la roda del costat contrari. També es pot fer que el robot roti sobre el seu eix accionant els dos motors simultàniament en sentits oposats. La figura 5 esquematitza aquests conceptes pel cas d'un robot amb una pota posterior, en visió zenital. Les fletxes laterals indiquen el sentit de gir de cada roda, i la fletxa gruixuda mostra el moviment resultant del sistema.

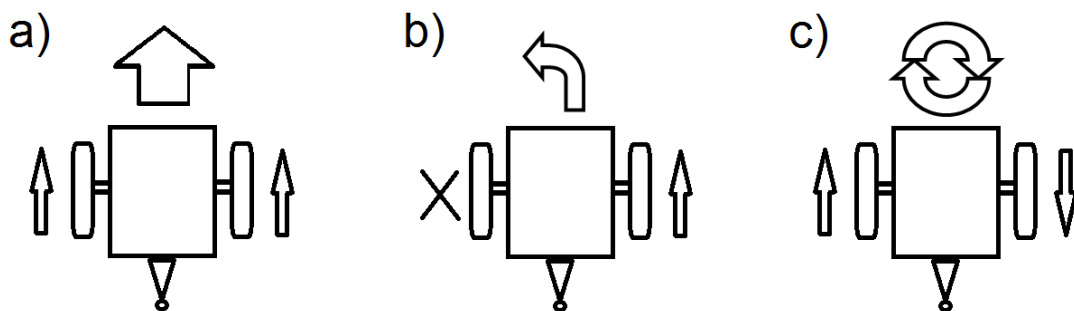


Figura 5. Resum de les possibilitats de moviment d'un robot amb dues rodes i dos motors: a) moviment unidimensional; b) girar cap a un costat; c) gir ràpid sobre el seu eix.

La tria d'aquesta configuració s'ha fet, en definitiva, perquè permet donar-li un valor afegit al projecte, el moviment en dues dimensions, amb un cost mínim, sense dificultar la consecució en un primer termini del requeriment bàsic de moviment en una dimensió.

3.1.2 Motors

Tractant-se d'un sistema autònom alimentat amb bateries, l'element propulsor ha de ser forçosament algun tipus de motor DC de baix voltatge. Aquests dispositius tenen unes necessitats de corrent elevades que els fan inadequats per ésser accionats directament per les sortides GPIO del controlador. Relés o una placa controladora de motors específica com per exemple [6] es fan necessaris.

Una alternativa a això, novament sorgida del món de la robòtica recreativa, és la utilització de servomotors modificats. Un servomotor o, en forma breu, servo, és originalment un motor DC al que mitjançant un senyal de control se li pot demanar de moure un eix a una posició angular donada dintre d'un rang delimitat.

El motor DC va connectat a una caixa d'engranatges. L'eix de sortida (al que connectem la roda) és solidari a un potenciòmetre, de manera que llegint la

resistència del potenciòmetre un circuit intern de control sap si ha de fer girar el motor en un o altre sentit fins a deixar l'eix en la posició demanada. Com que el servo conté el seu propi circuit de control, no necessita que el senyal de control li proporcioni potència, sinó que té tres entrades: positiu d'alimentació, control, i terra. Aquesta característica el fa ideal per al sistema que ens ocupa.

Però, per a poder-lo utilitzar per moure un vehicle rodat, cal modificar el servo perquè proporcioni un gir continu de 360°, i això normalment requereix dues modificacions: eliminar els bloquejos mecànics que limiten el rang de gir, i substituir el potenciòmetre amb una resistència fixa.

El primer punt es pot fer amb una fulla de modelisme o un bisturí, donat que el que cal es rebaixar un topall en algun dels engranatges que normalment estan fabricats en niló.

Pel que fa al potenciòmetre, la resistència que el substitueix ha de ser d'un valor tal que el circuit de control, en llegir-la, interpreti que l'eix està en una posició determinada, com per exemple, 90 graus. Si, per exemple, enviem el senyal de control que en condicions normals faria moure l'eix a la posició de 0 graus, el controlador decidirà que ha de fer girar l'eix cap a l'esquerra fins que la resistència prengui el valor corresponent, cosa que mai succeirà, i per tant el resultat és un accionament continu del motor cap a l'esquerra. Si, en canvi, enviem el senyal de control corresponent a 180 graus, el controlador creurà que cal moure l'eix en l'altre sentit, cap a la dreta, i anàlogament obtindrem un moviment continu en aquell sentit.

3.1.3 Xassís

Pel que fa al disseny del xassís, hem emprat un de propòsit general, popular, ben provat i disponible en el domini públic: el del robot MarkIII de la Portland Area Robotics Society [7]. procedir d'aquesta manera suposa un important estalvi de materials, i sobre tot de temps de desenvolupament en

evitar haver de fer prototipus i afrontar problemes potencials com manca d'estabilitat, posició inadequada del centre de gravetat, etcètera.

El procediment complet el podem trobar a [8]

Els servos utilitzats normalment en robòtica i vehicles de radiocontrol acostumen a tenir les següents característiques que suposen un estàndard de facto [9]:

- Alimentació: 4.8-6V DC
- Control: senyal PWM de 50 Hz amb amplada de pols en el rang 1-2 ms. La posició neutra correspon sempre a un pols de 1.5ms.

El fet que molts microcontroladors, incloent el popular Arduino, disposin de sortides configurades per proporcionar senyals de tipus PWM (pulse width modulation), és la última característica que fa que l'ús de servos en petits projectes de robòtica. Afortunadament, el LPC1769 disposa de sis sortides d'aquest tipus.

El model de servomotor utilitzat en aquest projecte és el Futaba s3003. Es tracta d'un model molt comú que es troba fàcilment en botigues d'electrònica i les seves característiques, que coincideixen exactament amb les que acabem de descriure, es poden trobar a [10]

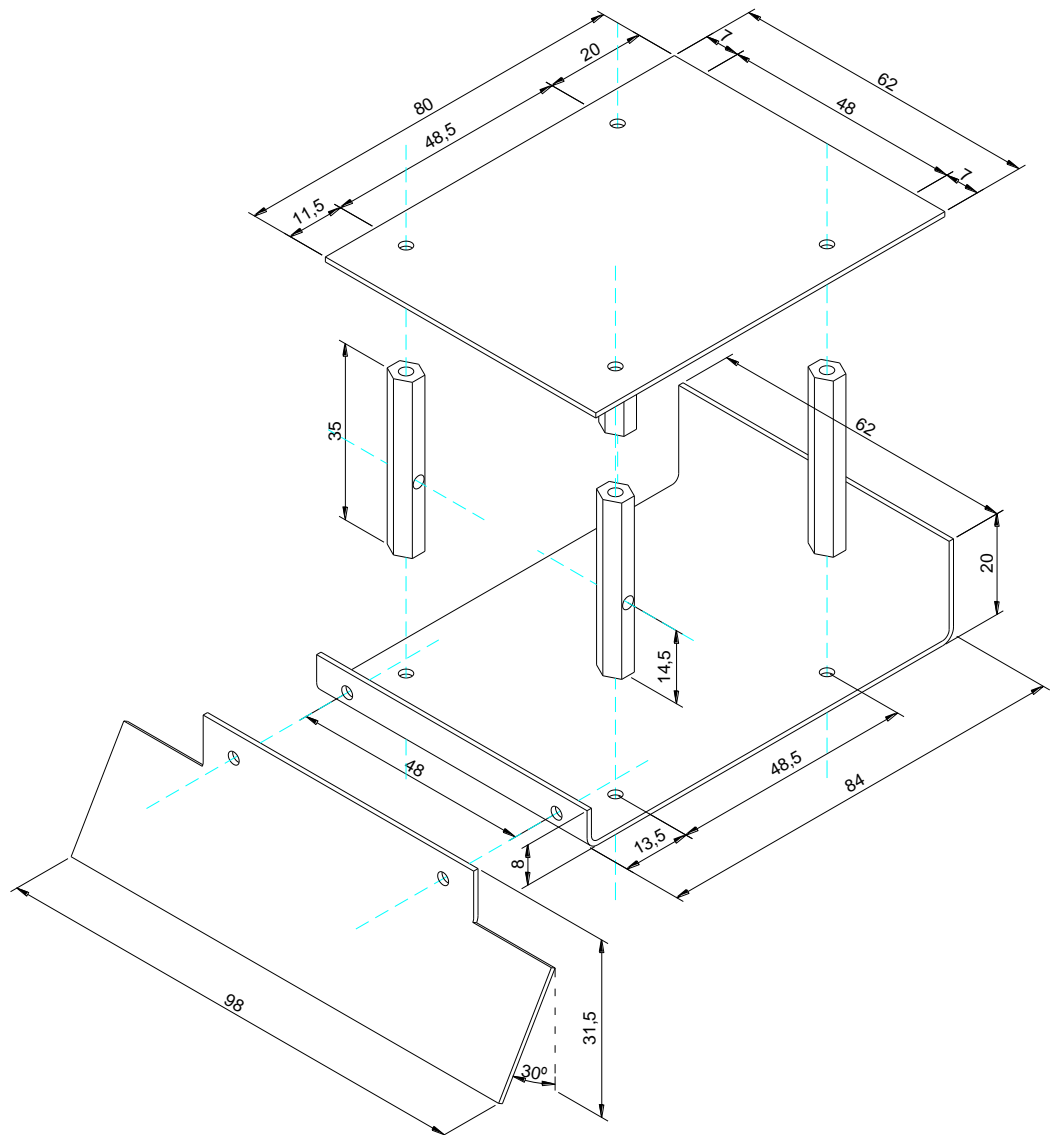


Figura 6. Esquema del xassís del robot MarkIII de la PORTS, utilitzat en aquest treball

El xassís és senzill de fabricar, doncs el material utilitzat, planxa d'acer d'un mil·límetre de gruix es pot tallar, foradar, llimar i doblegar sense eines sofisticades. La figura 6 mostra un esquema del xassís. Es pot apreciar que l'element principal són dues plataformes unides per pilons d'alumini hexagonals. Les mides estan triades per allotjar a sobre de la plataforma inferior dos servomotors estàndard, fixats amb cargols als pilons d'alumini, i que mouen dues rodes també estàndard amb banda de goma i 67 mil·límetres

de diàmetre. La plataforma superior està destinada al microcontrolador i la seva circuiteria auxiliar.

L'últim element del xassís és una planxa frontal o escut, sobre el que descansa l'equilibri del robot i que està, doncs, en tot moment en contacte amb el terra. Aquest escut frontal és també el lloc pensat per col·locar-hi els sensors que pugui necessitar el robot per fer la seva tasca, com per exemple sensors de reflectivitat en el cas d'un seguidor de línies.

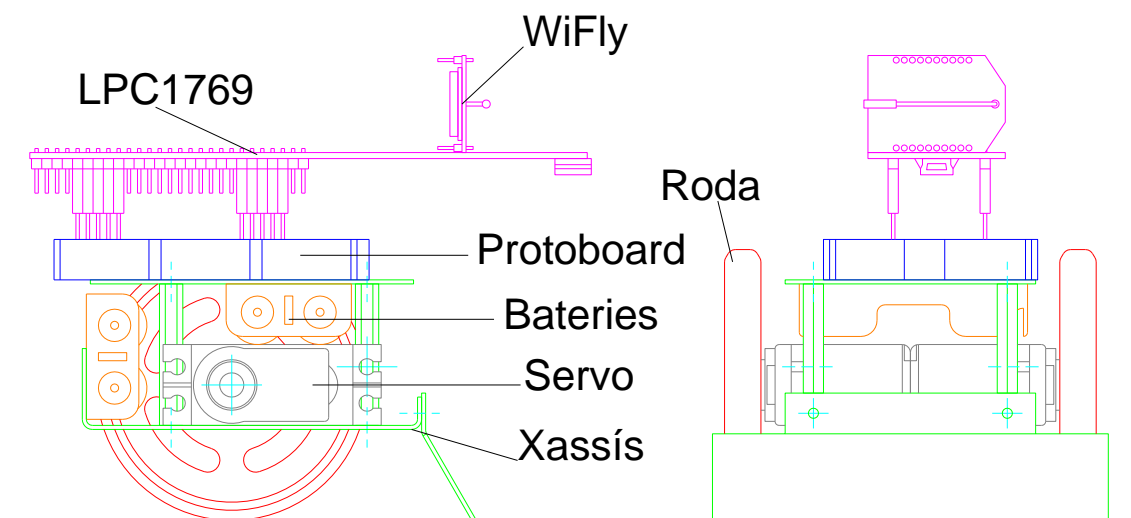


Figura 7. Esquema del robot Rhino-1, basat en el xassís Mark III.

El robot objecte d'aquest treball ha estat construït a voltant d'aquest xassís com es mostra a la figura 7. En l'espai entre ambdues plataformes s'hi allotgen dos portapiles dobles de mida AA per alimentar el sistema. La posició de les bateries juga un paper important en l'equilibri del robot, doncs es tracta d'un dels elements més pesants i cal situar-lo el més baix possible i intentant carregar el pes en la part davantera, on l'escut frontal fa contacte amb el terra.

A sobre de la plataforma superior hem fixat una placa de prototipatge electrònic (proto-board) de 5 x 7,5 cm. L'espaiat entre els contactes d'aquesta

placa és l'estàndard de 2,54mm, igual que el de la placa LPCXpresso, que hem fixat mecànicament a la primera, de cap per avall i amb l'ajuda de contactes allargadors. El mòdul d'interfície de depuració de la placa LPCXpresso queda situat a la part davantera, la qual cosa com hem dit contribueix a l'equilibri general del robot. A sobre, a prop de l'extrem, s'hi ha situat el mòdul de xarxa WiFly, en posició vertical i orientat cap a un costat. El propòsit d'aquesta disposició és que l'antena del dispositiu es trobi en pla horitzontal i apuntant cap a un costat, tal i com explicarem en el moment de descriure la implementació de la funcionalitat de cerca de cobertura de xarxa. La forma final del robot, amb aquesta protuberància en la part davantera que recorda la banya d'un rinoceront, ha conduït a batejar-lo amb el nom Rhino-1.

3.2 Descripció elèctrica

L'interconnexió dels diferents elements del sistema a nivell elèctric, és a dir, placa principal, mòdul de xarxa, interfície uart-usb per a depuració, motors i alimentació, es mostra esquematitzada a la figura 8.

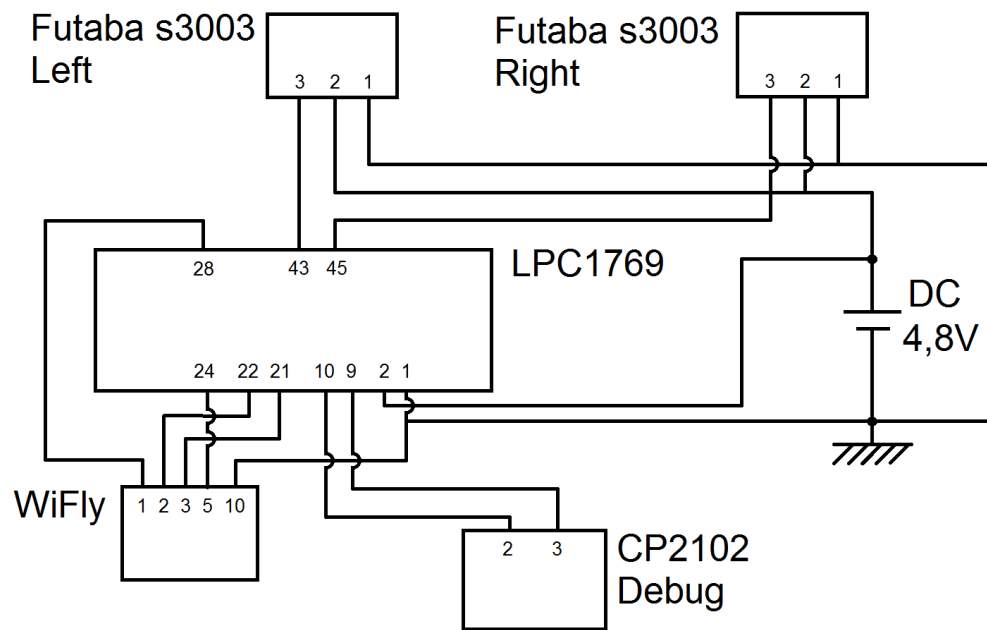


Figura 8. Connexionat elèctric del sistema

	P	Function	Connected to	Color
in				
L				
PC769				
	1	GND	Batt -	Black
	2	VIN(4.5-5.5V)	Batt +4.8V	Red
	9	P0.0(TX3)	CPE2102 Pin 3	Gray
)	(TX)	
0	1	P0.1(RX3)	CPE2102 Pin 2	White
)		
1	2	P0.2(TX0)	WiFly Pin 3 (RX)	Yellow
)		
2	2	P0.3(RX0)	WiFly Pin 2 (TX)	Green
)		
4	2	P0.22	WiFly Pin 5 (RST)	Orange

8	2	VOUT +3.3V	WiFly Pin 1 (POW)	Red
3	4	PWM1.2	Left motor pin3 (CTRL)	Yellow
5	4	PWM1.4	Right motor pin3 (CTRL)	Yellow
4	5	GND	WiFly Pin 10 (GND)	Brown

Per a l'alimentació del sistema, hem decidit utilitzar quatre piles recarregables de NiMH que proporcionen una tensió de 1,2v cadascuna, repartides en dos portapiles dobles que com ja s'ha explicat en l'apartat referent al xassís, s'allotgen entre les dues plataformes. Aquesta tria permet una única font d'alimentació per a tot el sistema, donat que:

- Els servomotors requereixen una entrada en el rang 4,8-6V.
- La placa LPC1769 té un circuit regulador propi a l'entrada que es pot alimentar amb una tensió entre 4,5 i 5,5V pel pin 2.
- La placa LPC1769 disposa d'una sortida regulada a 3,3V (pin 28), que utilitzem per alimentar el mòdul WiFly pel pin 1.

La resta de connexions serveixen per a la utilització dels ports UART0 per a la comunicació entre la placa LPC1769 i el mòdul Wifly (pins 9/10 de la placa i 3/2 del WiFly), i UART3 per a la sortida de depuració al PC a través del convertidor USB/UART CP2102. També s'utilitza el pin 24 del LPC1769 per reiniciar el WiFly (pin 5). Finalment, els motors van connectats a les bateries pels seus pins 1 i 2, mentre que el pin 3 que rep el senyal de control va connectat a les sortides PWM 2 i 4, configurades tal i com s'explicarà en el capítol dedicat a la llibreria de moviment.

4. Descripció funcional

En aquest capítol s'explicaran les principals decisions de disseny que han canalitzat el desenvolupament del projecte a nivell d'aplicació, quins son els components del sistema, i com interactuen entre ells.

4.1 Sistema total

Des d'un punt de vista funcional, el sistema té dos components principals: el robot i el PC remot que el controla. Al seu torn, dins del robot es reconeixen tres subsistemes: el microcontrolador LPC1769, el mòdul de xarxa WiFly i els motors. La figura 9 mostra el diagrama de blocs dels sistema.

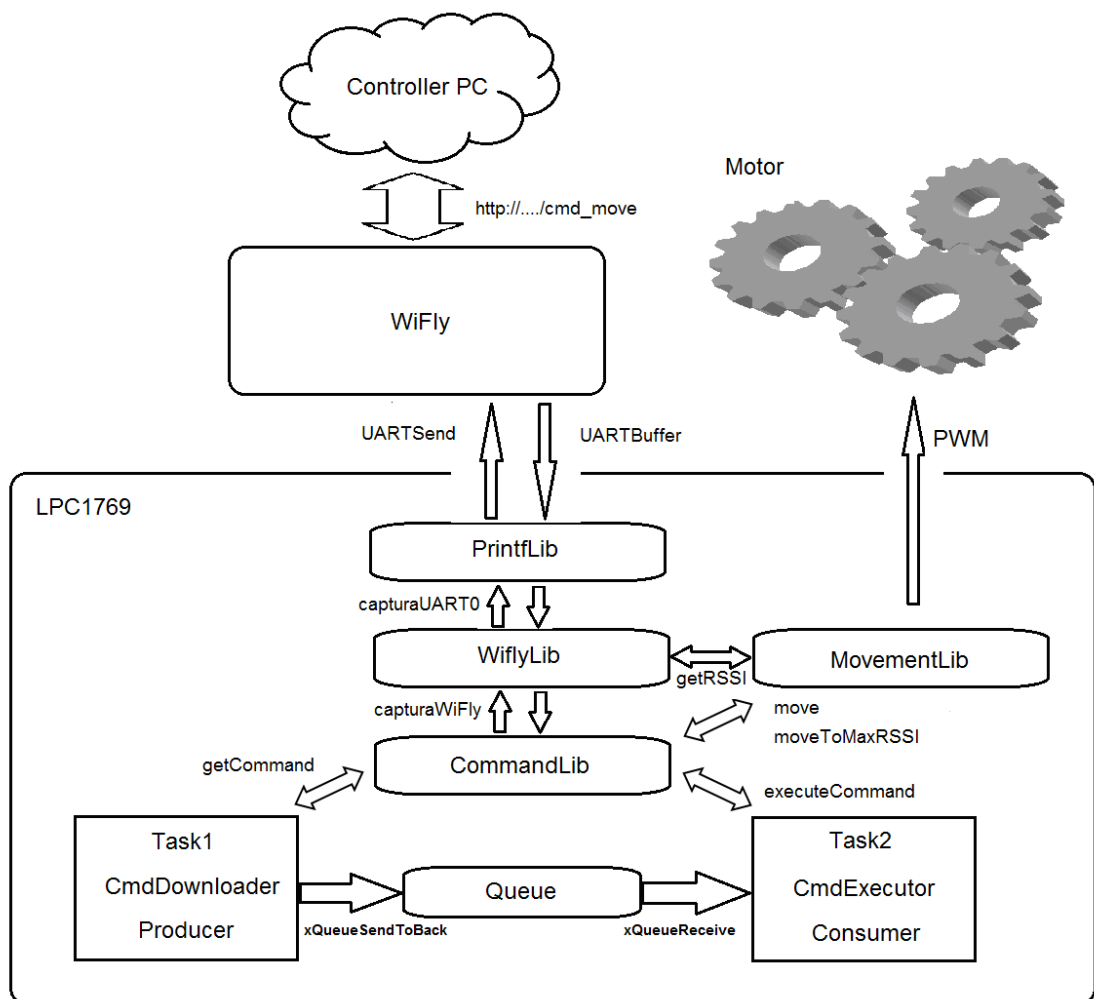


Figura 9. Diagrama de blocs de l'aplicació.

La primera decisió que s'ha pres és que les comunicacions des de la banda del client, és a dir, des del PC controlador siguin tan simples com sigui possible. Així, en comptes de dependre d'un aplicatiu client que calgui instal·lar, hem preferit un sistema que fos accessible des de qualsevol navegador web, i per tant, des de qualsevol dispositiu connectat a Internet en qualsevol moment. Les comandes s'envien en l'adreça url de la crida, tal i com s'explicarà en el proper capítol.

Dins del robot, el mòdul WiFly exerceix una funció peculiar. En sí mateix es tracta d'un sistema programable i amb certa capacitat de procés, capaç d'acceptar comandes través d'una interfície UART, o fins i tot des d'Internet via Telnet. Dels dos canals, el primer és el que fem servir per comunicar-hi des de la placa LPC1769. Del paràgraf anterior es dedueix que necessitem establir algun tipus de servei web per atendre les comandes, però aquesta és una funcionalitat que el mòdul WiFly, tot i estar capacitat per maquinari, no inclou en el seu firmware. Sí que pot, en canvi, gestionar peticions de connexió tcp entrants, establir sessions i reenviar la informació que circula en aquestes sessions a través de la UART. Això possibilita implementar el servei web a nivell http en l'aplicació del LPC1769, de manera que podem dir que el mòdul WiFly exerceix com a una mena interfície de xarxa d'alt nivell, doncs s'encarrega de tots els nivells fins al de sessió en el model OSI.

El mòdul WiFly té una altra funció, perquè també és la font de l'única altra entrada d'informació que té el sistema (a part de les comandes), que és la lectura del paràmetre RSSI de la connexió wifi que ens dona una mesura de la cobertura.

El sistema té dues sortides cap a l'exterior. Una és la resposta http que es fa arribar al PC client; l'altra és el moviment que el robot fa com a resultat de la comanda. En aquest sentit, els motors són un perifèric de sortida del sistema.

4.2 Disseny de l'aplicació

Una anàlisi elemental de la missió a realitzar per l'aplicatiu a executar al mòdul LPC1769 és que ha de fer dues tasques diferents: La primera, comunicar-se amb l'exterior, per mitjà del mòdul WiFly, per rebre comandes; la segona seria executar aquestes comandes transformant-les en moviment del sistema. No voldrem que les comandes que arribin mentre s'està executant una altra es perdin, i per tant caldrà que ambdues tasques s'executin en paral·lel.

El que aquests requeriments determinen és la utilització d'una aplicació multitasca, i entre aquestes, d'un model productor-consumidor. Efectivament, tenim la funció de la primera tasca, `cmdDownloader`, és obtenir un producte, la comanda, que serà executat i per tant consumit per l'altra tasca. Ambdues tasques treballen primordialment amb subsistemes diferents; la primera amb les comunicacions amb l'exterior, i la segona amb els motors. Però, les tasques s'han de comunicar doncs el resultat de la primera alimenta la segona, a través d'una cua.

Tot això es pot observar en la figura 9 que tot just hem presentat. També s'hi il·lustra el fet que l'aplicació, per fer la seva feina, ha d'assolir objectius a nivells molt diferents. Per exemple, per a rebre una comanda hem d'acabar llegint al port UART el que ens comunica el client a través del mòdul WiFly. En aquest camí acabem passant d'un objectiu de nivell aplicació, obtenir una comanda, a un de tasca a baix nivell, gestionar una entrada/sortida. Clarament, per qüestions d'intel·ligibilitat, eficiència i reusabilitat no seria assenyat d'incloure tot el codi que efectua totes aquestes funcions en una sola peça de codi, i és per això que es planteja la necessitat de descompondre l'aplicació en capes segons el nivell dels objectius a satisfer en cada cas.

Així, els objectius a nivell d'aplicació són els que tenen les tasques, obtenir comandes i executar comandes. D'això s'encarrega la capa de gestió de comandes: la llibreria `commandLib`.

Per produir comandes, la capa gestió de comandes necessita el cos d'una petició rebuda pel mòdul WiFly, mentre que per executar una comanda, s'ha de donar una ordre concreta als motors. Aquest nivell el resol la capa de sistemes, formada per les llibreries wiflyLib i movementLib.

Per tractar amb el mòdul WiFly cal gestionar les comunicacions per la UART, mentre que per enviar comandes als motors cal gestionar les sortides PWM. Ens trobem, doncs en la capa d'entrada/sortida. Per fer la primera tasca tenim la llibreria printfLib. Per a la segona, tenim el fitxer PWM.c dintre de la llibreria movementLib. Com que cap altra part de l'aplicació utilitza els ports PWM, descomposar la llibreria movementLib en dos projectes només per posar de manifest la descomposició en capes no sembla prou justificat.

La figura 10 resumeix la estructura en capes que acabem de descriure.

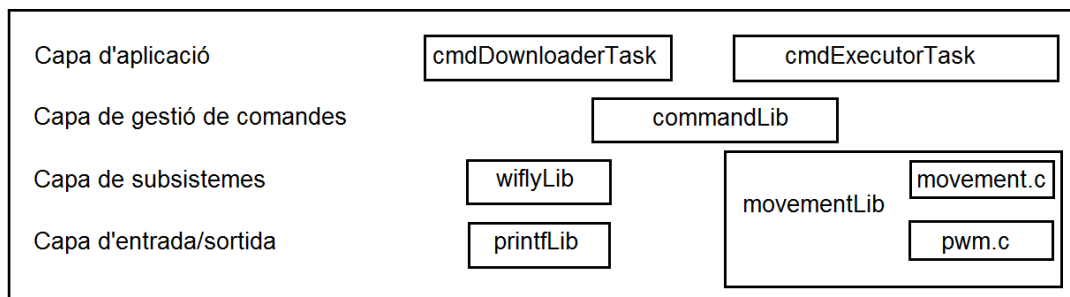


Figura 10. Estructura de capes de l'aplicació

5. Descripció detallada

En aquest apartat detallarem, capa per capa, com s'ha implementat les decisions de disseny exposades en el capítol anterior, i quines funcions s'ofereixen en cada nivell. Explicarem primer la capa més externa, la de l'aplicació per tenir una idea general, i després les altres de més interna a més externa. Ho fem així perquè cal haver explicat les funcions de baix nivell abans d'intentar descriure les d'alt nivell que les fan servir.

5.1 Capa d'aplicació.

Hem explicat que tractem amb un aplicatiu multitasca, i específicament amb un que segueix un model productor-consumidor. Per a fer-ho, hem inclòs en el projecte el kernel de FreeRTOS 6.1.1, que ens aporta les funcionalitats de planificació i gestió de tasques d'un autèntic sistema operatiu en temps real, incloent la gestió de cues, que necessitem per a passar les comandes d'una tasca a l'altra. Dintre d'aquesta capa l'únic que es fa es crear les tasques i la cua, engegar el planificador i, un cop a dins de les tasques, executar en un bucle infinit les operacions del gestor de comandes `getCommand` (a la tasca productora) i `processCommand` (a la tasca consumidora). En el primer cas, la comanda obtinguda en cada cicle es passa a la cua, mentre que en el segon, abans d'executar una comanda, cal extreure-la de la mateixa cua. El diagrama de flux es pot veure a la figura 11.

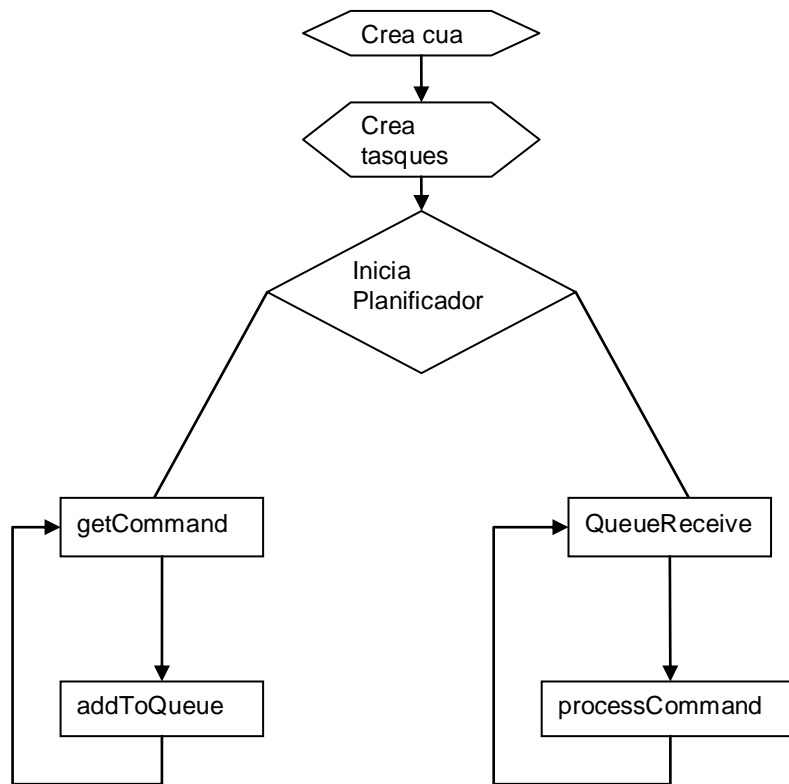


Figura 11. Diagrama de flux de l'aplicació

5.2 Capa d'entrada/sortida.

5.2.1 Llibreria printfLib.

Aquesta llibreria els ports UART per enviar i rebre missatges, cosa que en l'aplicatiu fem servir per a comunicar-nos amb el mòdul WiFly (UART0), i per a treure missatges per pantalla en un PC amb l'objectiu de depurar l'aplicació (UART3). Les funcions més rellevants són:

`printfInit`: inicialitza un port UART.

`printfString`: envia un missatge per la UART, sota la protecció d'un semàfor, i opcionalment espera una resposta a la mateixa UART. Això serveix, per exemple, per saber què respon el mòdul Wifly a una comanda que li enviem.

`capturaUART0`: espera indefinidament, en estat d'espera passiva, que apareguin missatges al port UART per recuperar-los.

5.2.2 Libreria movementLib/fitxer pwm.c

Aquí tenim el codi que gestiona de manera més directa les sortides PWM que controlen els motors. Les funcions més rellevants són:

`PWM_Ini`: Fa la inicialització de les sortides PWM2 (pin 43) i PWM4 (pin 45) amb els paràmetres correctes per al nostre sistema, que inclouen la freqüència (50Hz) i l'ample de pols inicial (1500µS).

`PWM_Start/PWM_Stop`: Activa/desactiva una sortida PWM.

`PWM_Set`: canvia l'amplada de pols en una de les sortides PWM. En el nostre sistema, valors més grans de 1000 µs però menors de 1500 µs serveixen per ordenar que un motor vagi endavant, mentre que si són més grans de 1500 µs però menors de 2000 µs, fan que vagi endarrere. Un valor d'exactament 1500 deixa el motor aturat.

5.3 Capa de subsistemes.

Com hem vist al capítol 3, aquesta capa serveix per proveir les crides necessàries quan les capes superiors necessiten serveis dels subsistemes externs a la placa LPC1769, és a dir, el mòdul Wifly i els motors. Per comunicar-s'hi utilitzen les crides del nivell d'entrada/sortida.

5.3.1 Llibreria wiflyLib

Atén totes les funcions necessàries del mòdul wifly. Les funcions que són més rellevants per a l'aplicatiu són:

`wifly_reset`: reinicia el dispositiu mitjançant un senyal al seu pin 5.

`wifly_setTcpServerConfig`: configura el mòdul per atendre connexions tcp. El paràmetres necessaris són: protocol, port, mida de paquet i missatge de resposta a la connexió (cal deixar-lo buit). Ha de guardar la configuració i fer una crida a `wifly_reset` per tal que els valors prenguin efecte

`wifly_closeTcpConnection`: tanca una connexió TCP des del costat del servidor. Cal notar que, per una errada en el firmware del mòdul WiFly, no s'envia correctament el missatge FIN al client, de manera que aquest desconeix que la sessió ha finalitzat.

`Wifly_getRSSIInt`: consulta el paràmetre RSSI de la connexió wifi actual, que dona una mesura del nivell de cobertura. Opcionalment retorna un valor promig de diverses mesures per millorar-ne la precisió.

5.3.2 Llibreria movementLib/fitxer movement.c

5.3.2.1 Funcions de moviment simples.

Fan crides a les funcions de pwm.c per configurar les amplades de pols de les sortides amb valors adequats per produir els moviments, segons l'esquema mostrat al capítol 2:

- movement_forward : canal esquerre i dret, 1100 μ s
- movement_back: canal esquerre i dret, 1800 μ s
- movement_stop: canal esquerre i dret, 1500 μ s
- movement_turnRight: canal esquerre 1200 μ s, dret, 1800 μ s
- movement_turnLeft: canal esquerre 1800 μ s, dret, 1200 μ s

5.3.2.2 Funcions de cerca de cobertura.

moveToMaxRSSI_1D. Fa que el sistema vagi endavant o endarrere entre lectures del paràmetre RSSI cercant-ne el valor òptim, i ho pot fer de dues maneres segons el valor del paràmetre step_by_step. Si l'utilitzem, farem que el dispositiu vagi fent passes curtes i aturant-se a fer cada mesura, mentre que si aquest paràmetre val 0 les mesures es faran amb l'equip en marxa, un sistema més ràpid però també més imprecís.

L'execució comença anant endavant fins que es detecti que la cobertura empitjora, la qual cosa es pot deure a que o bé ens hem passat de màxim, bé el sentit inicial ens allunya d'ell. Aleshores, fem que el robot es mogui endarrere novament fins que la cobertura empitjori. Si detectem que la cobertura actual està dintre d'un marge raonable del màxim detectat, donem la rutina per acabada; si no, tornem a provar-ho. El diagrama de flux d'aquesta funció és el que es mostra a la figura 12.

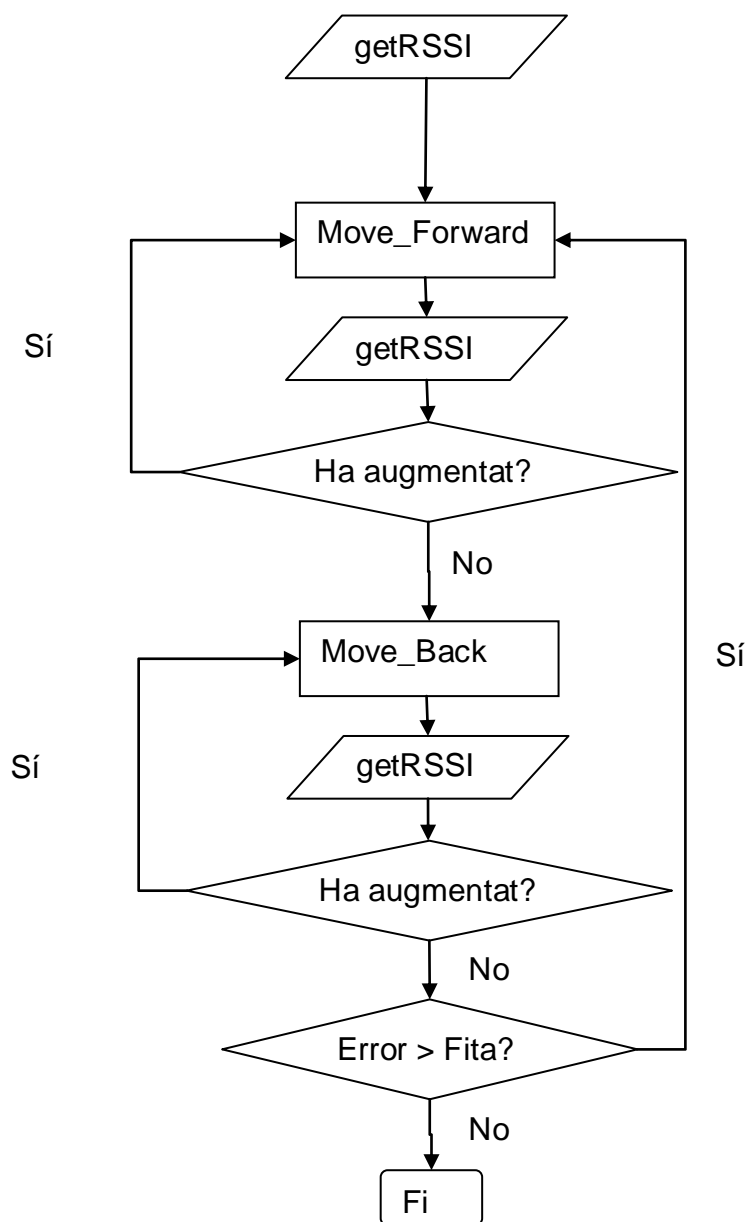


Figura 12. Diagrama de flux de la cerca de cobertura en una dimensió.

moveToMaxRSSI_2D: intenta estendre la capacitat de cerca a dues dimensions amb els mitjans disponibles. La idea que hem intentat aprofitar és la següent: el tipus d'antena, de fil, que porta el mòdul wifly té una sensibilitat igual en les dues dimensions perpendiculars al recorregut de l'antena, i més baixa conforme ens allunyem d'aquest pla. Aleshores, si col·loquem l'antena en el pla paral·lel al terra i apuntant directament cap a un costat (90° a dreta o esquerra), la màxima lectura de cobertura s'hauria de produir quan el punt d'accés es troba directament davant o darrera del nostre dispositiu. Segons aquest principi, el que fa la funció és fer girar el dispositiu sobre el seu propi eix per trobar la direcció on hi ha màxima cobertura, per a continuació aplicar l'altra funció, moveToMaxRSSI_1D, per arribar a l'objectiu. El diagrama de flux corresponent es mostra a la figura 13.

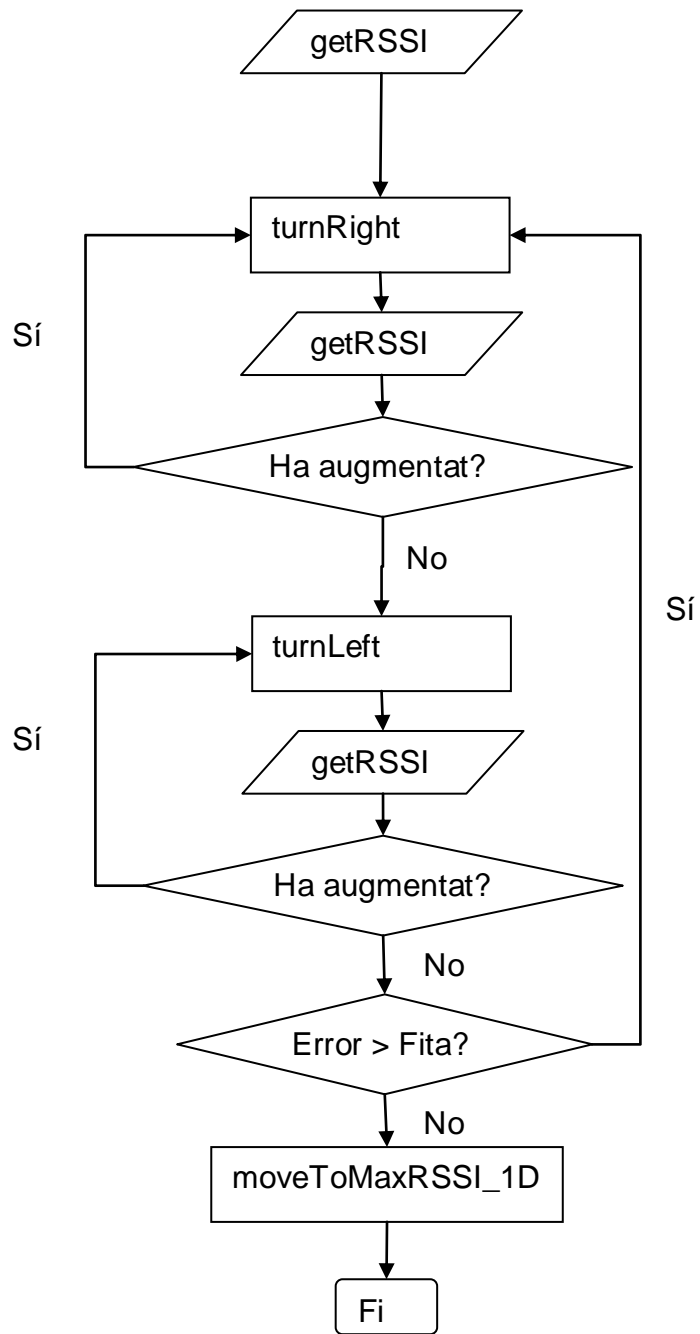


Figura 13. Diagrama de flux de la cerca de cobertura en dues dimensions.

5.4 Capa de gestió de comandes.

5.4.1 Llibreria commandLib.

En aquesta capa hi ha tres comandes, `commandInit`, `getCommand` i `processCommand`. La primera d'elles, `commandInit`, s'encarrega de deixar el sistema en un estat capaç de rebre comandes. Per a fer-ho, inicialitza el mòdul `WiFly`, el configura per atendre a connexions `tcp`, el connecta a la xarxa i inicialitza un semàfor.

Aquest semàfor és necessari perquè la interfície `UART` que comunica amb el mòdul `WiFly` s'utilitza per una banda per esperar comandes, però també és la via per la qual el mòdul respondrà a la funció `getRSSI` que retorna una mesura de la cobertura de xarxa, i totes dues coses no es poden fer alhora.

`getCommand`, per la seva banda i sota la protecció del semàfor que acabem de mencionar, fa les següents tasques:

- Esperar missatges del `Wifly (UART0)`, que correspondran al codi `http` encapsulat en una sessió `tcp` oberta pel navegador del client.
- Extreure la comanda de la url de la crida `http`.
- Composar una resposta `html`

`getCommand`, crida la funció `printf_captura_UART0` descrita abans per obtenir de dades bolcades pel client en la sessió `TCP`. El mètode de comunicació que hem triat és, ja que el mòdul `wifly` no pot treballar com a servidor web directament, simular-ho, de manera que les comunicacions des del costat del client siguin extremadament simples i requereixin només un navegador d'Internet.

Així, des del client intentarem accedir a una url consistent en l'adreça ip del mòdul `wifly` (si volem obtenir-hi accés des de fora de la nostra wlan ho

podrem fer redireccionant ports al router), més el path relatiu del recurs al que estariem intentant accedir. Si per exemple al navegador entrem l'adreça `http://192.168.43.69/cmd_move_forward`, el que farà el navegador és obrir una sessió TCP amb el host 192.168.43.69 (el mòdul wifly) i enviar-li la següent comanda GET d'HTTP:

```
OPEN*GET /cmd_move_forward HTTP/1.1
Host: 192.168.43.69
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64;
rv:13.0) Gecko/20100101 Firefox/13.0.1
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: ca,en-us;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
DNT: 1
Connection: keep-alive
```

El que fem, doncs és definir una nomenclatura en que les comandes que volem executar s'especifiquen en el path de l'adreça que li entrem al navegador després d'un prefixe `cmd_` que usem per detectar i extreure la comanda del flux TCP, cosa que fa la funció `getCommand`.

La segona part de la seva funció es compondrà en format html que, dintre de la mateixa sessió TCP, enviarem de tornada al navegador perquè el client sàpiga que la seva comanda ha arribat amb èxit o, per contra, s'ha perdut. Per enviar aquest missatge de resposta utilitzem el chunked transfer protocol perquè, segons hem pogut trobar, la versió actual del firmware del mòdul wifly no tanca correctament les sessions TCP, o més concretament, no envia correctament al client el senyal de finalització. Com que el chunked protocol té la seva manera pròpia d'indicar el final del missatge, el navegador és capaç de detectar-ho i operar correctament.

L'última funció que ens queda per comentar és `processCommand`, que dona accés a les funcions de moviment segons el contingut de la crida. Les funcions de cerca de cobertura estan protegides per un semàfor que també controla l'entrada a `getCommand`. Això és així perquè `moveToMaxRSSI_1D` i

moveToMaxRSSI_2D criden a getRSSI i han d'esperar el resultat per via de la mateixa UART que getCommand ha d'escoltar en espera de noves comandes.

6. Viabilitat tècnica

Per entrar en una valoració de la viabilitat de l projecte caldria especificar clarament com es defineix el concepte. El projecte és plenament viable en el sentit que ha quedat demostrat que és possible aconseguir els requisits demanats amb els mitjans disponibles. Si per altra banda, la pregunta és si el producte aconseguit té una viabilitat comercial, la resposta l'hem començat a elaborar en el capítol d'antecedents.

Com s'ha comentat en aquell capítol, si ens referim a un sistema que compleixi solament els requisits del projecte, és a dir, un petit vehicle governable des d'Internet que sigui capaç de buscar els punts de cobertura òptima, hi ha innumerables solucions molt més properes a entrar en fase de producció. Si, al contrari, ho entenem en un context més ampli plantejant el projecte com una prova de concepte per establir si és raonable pensar en afegir capacitats com les tractades als típics sistemes que utilitzen un maquinari similar a l'aquí emprat, la valoració tornaria a ser favorable.

Pel que fa als punts dèbils del sistema, el que més destaca és que el maquinari no acaba de ser l'òptim per a la tasca endegada. La utilització del mòdul WiFly com a interfície de xarxa a través d'un port UART presenta problemes, alguns dels quals ja hem vist anteriorment: no pot actuar com a servidor web, no tanca correctament les sessions tcp, ens trobem amb competició pel recurs quan intentem escoltar noves peticions mentre s'està executant alguna que requereix l'ús de la UART, certes configuracions requereixen reiniciar-lo en un procés que pot trigar diversos segons, etc. Seria desitjable trobar una solució de connectivitat millor, i donat que les especificacions del LPC1769 indiquen que té capacitat ethernet, no sembla fora de lloc pensar que es podria arribar algun tipus de punt d'accés (AP), que si bé òbviament mai no tindrà el baixíssim consum del WiFly, podria ser prou bo per a fer un robot operatiu.

7. Valoració econòmica

Concepte	Quantitat	Preu unitari	Total
Hores del recurs Cèsar Guerrero /analista-programador - Ordinari	160	35	5600
Hores del recurs Cèsar Guerrero /analista-programador – Extraordinari	40	70	2800
LPC1769	1	20	20
RN-171-XV	1	30	30
Servo futaba s3003	2	15	30
Planxa d'acer 1 mm, m ²	0,1	6,46	0,65
Rodes	2	3,70	7,40
Portabateries	2	1,23	2,46
TOTAL			8490,51

8. Conclusions

A la conclusió d'aquest treball s'ha aconseguit l'objectiu principal, el de portar a termini un projecte en totes les seves fases materialitzant i aprofundint els coneixements adquirits durant la carrera, sobre tot en l'àmbit primordial del projecte, els sistemes encastats.

També, i això no entrava en els meus plantejaments al començament de l'assignatura, m'ha permès acostar-me a una temàtica fascinant i que desconeixia per complet, la de la robòtica recreativa.

Ara és el moment de fer el repàs final a la consecució dels objectius del sistema desenvolupat:

Objectius bàsics:

(a) Sistema de moviment endavant i enredera:

El sistema ha de poder executar comandes de moviment en una dimensió, és a dir, avançar, retrocedir i aturar-se.

Aquest objectiu s'ha superat sense problemes.

(b) Connexió via Wifi:

El sistema ha de poder enllaçar amb una xarxa sense fils oberta o amb seguretat WEP o WPA.

Aquest objectiu també s'ha superat sense problemes.

(c) Control remot des de Internet:

El sistema ha de poder ser controlat des de qualsevol dispositiu connectat a Internet per mitjà d'un simple navegador web.

També s'ha aconseguit, depenent només de l'adequada configuració del router perquè faci el corresponent redireccionament de ports (NAT).

(d) Recepció de comandes de control:

El sistema ha d'incorporar un servidor web que permanent escolti i interpreti les comandes enviades per mitjà de peticions http.

També s'ha aconseguit. L'escolta només s'interromp, momentàniament, quan cal bloquejar l'accés al mòdul wifly per consultar la cobertura.

(e) Sistema automàtic de cerca de cobertura:

El sistema ha de ser capaç de verificar en nivell de cobertura de xarxa de que disposa i moure's a una posició millor mitjançant un algorisme que utilitzi les comandes de posicionament.

La versió en 1 dimensió funciona sense problemes.

Objectius de valor afegit:

(f) Sistema de moviment en dues dimensions:

S'implementaran funcions de gir que permetin al prototipus moure's en el pla.

Objectiu aconseguit sense dificultats.

(g) Sistema automàtica de cerca de cobertura en dues dimensions:

El sistema ha de ser capaç de trobar la direcció de millor cobertura i moure-s'hi.

Objectiu raonablement satisfet. De tots els objectius, aquest, que tenia la consideració “de valor afegit” ja que no formava part dels exigits a les especificacions, és el que dona peu a més comentari. Tractant-se d’un robot “cec”, que no té consciència de la seva posició, no sembla vàlid cap algorisme de cerca que no impliqui intentar encarar directament la direcció de màxima cobertura. Diem això perquè a priori es podria pensar en fer escombrats de moviment en dues dimensions calculant la posició segons el temps i la velocitat per a conformar un mapa de cobertura. Veiem, però, que a poc que el robot rellisqui o derrapi, i això succeeix amb freqüència, la validesa del càlcul s’esfumaria.

Creient, doncs, en la necessitat de ser capaç de detectar diferències de cobertura fent rotar el robot sobre el seu eix, ens hem trobat malauradament que, tot i que l’efecte es produeix, és massa feble com per resultar plenament efectiu. Pensem que caldria una configuració d’antena molt més direccional que la que tenim. Veient que el model RN-171XV s’oferta també en versió amb connector d’antena estàndard, pensem que valdria la pena comprovar aquesta possibilitat.

Així mateix, i tot i que ni tan sols es van arribar a plantejar com a objectius en avaluar correctament que no serien abastables en el temps disponible, seria molt interessant d’evolucionar el robot dotant-lo de sensors d’ultrasons avisadors de proximitat i una brúixola electrònica que permetés determinar la direcció amb precisió.

Una altra millora que afegiria valor al producte, tot i que a canvi d’un volum de feina considerable, seria una millora de la interfície del client dissenyant un aplicatiu web adient. Cal considerar, però, sense sensors el robot té molt poca informació que retornar a l’usuari, i que l’única mesura que podem obtenir en temps real, el paràmetre RSSI que ens indica la cobertura, ens obliga a terminar la connexió tcp actual per a poder-la obtenir.

9. Glossari

RSSI. Received Signal Strength Indicator. Mesura de la potència d'un senyal rebut.

WiFiFly. Mòdul de xarxa RN-171XV de Roving Networks

Mota. Plataforma de sensors sense fils miniaturitzada.

Microcontrolador. Petit ordinador, sovint en un sol circuit integrat, que conté el nucli del processador, memòria i dispositius d'entrada/sortida integrats.

NAT. Network Address Translation. Redireccionament de ports.

Robot. Entitat virtual o mecànica artificial. En general sistema electròmecànic que per la seva aparença o el seu moviment ofereix la sensació de tenir propòsit propi.

Sistema encastrat. Sistema multiagent basat en un petit ordinador que dissenyat per desenvolupar tasques específiques en el sí d'un sistema més gran.

10. Bibliografia

[1]. <http://www.bricogeek.com>

Última visita: 15 gener 2013.

[2]. <https://forum.sparkfun.com>

Última visita: 15 gener 2013.

[3]. <http://ro-botica.com>

Última visita: 15 gener 2013.

[4]. [http://hlbtoys.en.alibaba.com/product/603282038-213240040/WiFi helicopter with video live capability Latest design Helicopter Wifi Camera 604 w.html](http://hlbtoys.en.alibaba.com/product/603282038-213240040/WiFi_helicopter_with_video_live_capability_Latest_design_Helicopter_Wifi_Camera_604_w.html)

Última visita: 15 gener 2013.

[5]. <https://www.sparkfun.com/products/7915>

Última visita: 15 gener 2013.

[6]. <http://www.canakit.com/dual-motor-l298-h-bridge-control-ck1122-uk1122.html>

Última visita: 15 gener 2013.

[7]. <http://www.junun.org/MarkIII/Manual/index.jsp>

Última visita: 15 gener 2013.

The Mark III hardware design, including schematics and PCB layout are Copyright (C) 2002 Portland Area Robotics Society. All rights reserved. The Mark III hardware design and associated software are available for any non-commercial or non-profit use provided that the appropriate credit is given to PARTS and that this copyright notice is attached.

[8] <http://www.seattlerobotics.org/guide/servohack.html>

Última visita: 15 gener 2013.

[9] http://en.wikipedia.org/wiki/Servo_%28radio_control%29

Última visita: 15 gener 2013.

[10]. http://www.servocity.com/html/s3003_servo_standard.html

Última visita: 15 gener 2013.