

ÍNDEX MEMÒRIA

Índex memòria.....	1
Resum.....	3
Resumen	3
Abstract	3
Objectiu	4
Capítol 1: INTRODUCCIÓ	5
1.1. Competicions de robots de sumo	5
1.1.1. <i>Tatami</i>	5
1.1.2. <i>Dimensions del robot</i>	6
Capítol 2: COMPONENTS ELECTRÒNICS	7
2.1. Sensors	7
2.1.1. <i>Detecció de línies-CNY70</i>	7
2.1.2. <i>Sensor de distància – SHARP® GP2Y0A41SK0F</i>	8
2.2. Motors i control	10
2.2.1. <i>Motor C-6061</i>	10
2.2.2. <i>L293D</i>	10
2.3. Microcontrolador ATMEGA16.....	12
Capítol 3: CIRCUITS ELECTRÒNICS.....	13
3.1. Alimentació	13
3.2. Circuits dels sensors.....	15
3.2.1. <i>Sensor de distància</i>	15
3.2.2. <i>Sensor detector de línies</i>	15
3.3. Motors.....	18
3.4. Circuit del microcontrolador	19
3.4.1. <i>Oscil·lador de quars</i>	19
3.4.2. <i>Boto d'inici</i>	20
3.4.3. <i>Leds</i>	21
3.5. Circuit de programació	22
3.6. Disseny de la PCB	24
Capítol 4: MECÀNICA	29
4.1. Estructura del robot	29

4.2.	Fixació i localització de sensors i actuadors	30
4.3.	Plaques reforçants	32
Capítol 5:	PROGRAMACIÓ	35
5.1.	Pins de connexió.....	35
5.2.	Diagrama de flux i pseudocodi.....	37
5.3.	Programació en C	39
5.3.1.	<i>Configuració</i>	<i>40</i>
5.3.2.	<i>Subrutines.....</i>	<i>44</i>
5.3.3.	<i>Programa principal</i>	<i>47</i>
5.4.	CodeVisionAVR	50
Capítol 6:	DIAGRAMA DE GANTT	55
Capítol 7:	CONCLUSIONS	57
Capítol 8:	BIBLIOGRAFIA	59

RESUM

Aquest projecte consisteix en el disseny, construcció i programació d'un robot lluitador de mini sumo. L'objectiu dels robots lluitadors de mini sumo és treure del tatami al robot contrincant. El projecte engloba l'elaboració de la placa electrònica, la estructura, que és la part mecànica, y la programació, que és la part informàtica. Aquí es poden diferenciar les tres parts que formen el projecte.

El robot s'ha construït amb peces de LEGO®, i el cervell del robot està format per un microcontrolador (μC) de la marca Atmel-AVR®, programat amb llenguatge C. Cal comentar que el projecte és de caràcter docent, i no pensat per a competir, ja que en aquest cas seria recomanable utilitzar una altra mecànica. El projecte està bàsicament centrat amb els camps de l'electrònica i la programació, la mecànica ocupa un paper secundari.

RESUMEN

Este proyecto consiste en el diseño, construcción y programación de un robot luchador de mini sumo. El objetivo de los robots luchadores de mini sumo es sacar del tatami al robot contrincante. El proyecto engloba la elaboración de la placa electrónica, la estructura, que es la parte mecánica y la programación, que es la parte informática. Aquí se puede diferenciar las tres partes que forman el proyecto.

El robot se ha construido con piezas de LEGO®, y el cerebro del robot está formado por un microcontrolador (μC) de la marca Atmel-AVR®, programado con lenguaje C. Es necesario comentar que el proyecto es de carácter docente, y no está pensado para la competición, ya que en este caso sería recomendable utilizar otra mecánica. El proyecto está básicamente centrado con los campos de la electrónica i la programación, la mecánica ocupa un papel secundario.

ABSTRACT

This project consists of the design, the construction and the programming of a small fighting robot of sumo. The aim of these robots is to take out the opponent robot of tatami. The project includes the development of the electronic board, the structure, which is the mechanical part, and the programming, which is the informatics part.

The robot has been build with LEGO® parts, and the brain of the robot is formed by an Atmel-AVR® microcontroller (μC), programmed with C language. Is important to comment that the project is educational and the competition is not the aim, because the mechanical is not the best choose to competition. The project is specially focused on the electronic and informatics part, the mechanical part has less importance.

OBJECTIU

L'objectiu principal d'aquest projecte és la realització d'un robot de mínim el qual respecti la normativa de competició i sigui eficient, sobretot electrònic i informàticament.

Un cop definit l'objectiu principal, ens centrarem en objectius més concrets. Els quals són:

- Optimització i integració al màxim de l'espai de la placa electrònica, per aconseguir això s'haurà d'intentar aconseguir una única placa electrònica la qual ho controli tot, i alhora sigui programable, sense ser necessari extreure el microcontrolador d'aquesta.
- Utilitzar uns sensors eficients, els quals no generin problemes, i si ho fan, solucionar-los.
- Utilitzar uns motors força forts i alhora petits, aquesta serà una difícil cerca.
- Elaboració d'una estructura prou forta, tot i no poder disposar de temps i recursos per a construir-la. Ja que és on s'hi dedicarà menys temps. Intentar que entri dins la normativa.
- Obtenció d'un programa eficient, el qual estigui protegit envers a situacions estranyes i que no facin que es pengi. També cal que sigui ràpid de reacció.

CAPÍTOL 1: INTRODUCCIÓ

Un combat de robots de minisumo consisteix en la lluita de dos robots els quals s'intenten expulsar del tatami, que és cercle que delimita el camp de combat. Per tant, abans de centrar-se amb les característiques tècniques i de disseny, s'explicarà les dimensions del robot, del tatami i les normatives a complir per a poder competir.

En aquest cas, s'ha centrat el disseny tant del tatami com el del robot amb la normativa de competició de robots de minisumo de l'UPC de Terrassa.

1.1. Competicions de robots de sumo

És lògic que al tractar-se d'una competició entre dos robots els quals són dissenyats per varies persones, existeixi una mínima normativa a complir alhora del disseny, tant en les dimensions com en el pes. Cal dir que hi ha varies modalitats, des de el robot de sumo, que va ser el primer, ja que era el més gros, degut a la difícil possibilitat de reducció del components electrònics. Fins al actuals microsumo, els quals tal i com diu el seu nom són molt petits i una mica lluny a l'abast del estudiants. Avui en dia la modalitat més adequada de cara la docència és la modalitat minisumo, ja que és la més accessible sobretot pel que fa als recursos tecnològics.

Per tant, el projecte s'ha centrat exclusivament amb el disseny d'un robot de minisumo, excloent les dimensions i normatives de les altres modalitats. A continuació es mostra els conjunt de normes a seguir per a poder-lo realitzar.

1.1.1. Tatami

El tatami és l'àrea de combat del robot, és simplement un cercle negre delimitat per una línia blanca, la qual delimita el camp. Quant un robot surt de la línia perd el combat.

El tatami és una superfície circular de fusta pintada de color negre mat amb un diàmetre de 77cm, situada a 2,5 cm del terra. Les línies que delimiten el camp seran pintades de blanc setinat de 2,5cm de grossor.

En el nostre cas, s'ha construït un tatami amb cartró ploma d'1 cm de grossor, el qual s'ha pintat amb pintura negre mat i la línia divisòria s'ha fet enganxant una cartolina blanca.

1.1.2. Dimensions del robot

El disseny del robot té dues limitacions bàsiques, el pes i les dimensions. El pes total del robot no pot superar els 500g, i les dimensions màximes que pot tenir són de 10x10 cm. Pel que fa a l'altura del robot no hi ha límit d'alçada, però no es recomanable que sigui gaire alt, ja que incrementa la possibilitat de ser bolcat.

Informació a tenir en compte:

- La bateria ha de durar tot el combat, no pot se substituïda.
- L'accionament del robot es farà manualment i reaccionarà al cap de 5s.
- La comunicació amb el robot està prohibida
- Està totalment prohibit danyar al robot contrari

Per a més informació detallada sobre la competició de la UPC de Terrassa consultar l'annex, ja que recull el document complet.

CAPÍTOL 2: COMPONENTS ELECTRÒNICS

2.1. Sensors

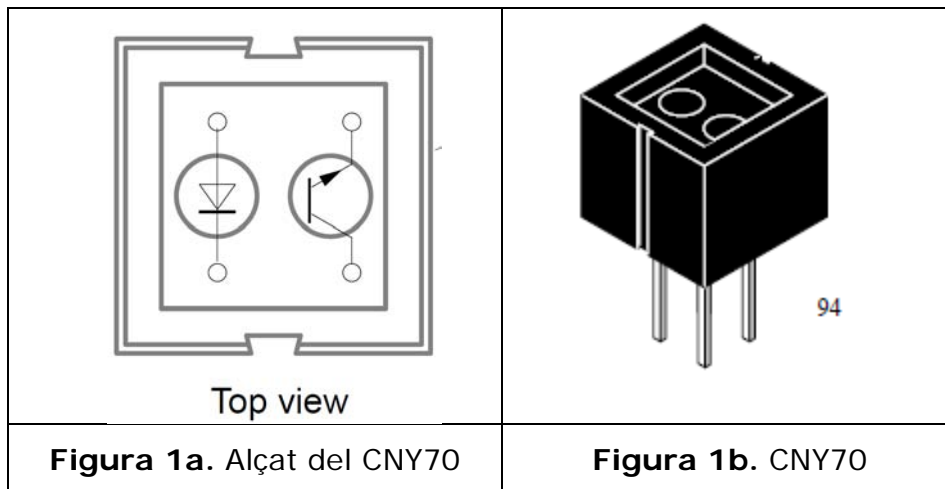
Els sensors són els encarregats d'adquirir informació del medi, transformen informació física en senyals elèctriques. En aquest cas, els sensor usats són bàsicament sensors de distància, els quals utilitzen la tecnologia dels infrarojos.

2.1.1. Detecció de línies-CNY70

Per a detectar el final del tatami s'utilitzarà un component anomenat CNY70, el qual detecta si el color es blanc o negre. Aquest component està compost per un encapsulat el qual inclou un fototransistor i un led infraroig. De manera que el funcionament d'aquest consisteix en emetre un raig de 950nm de longitud d'ona amb el led i a continuació llegir la reflexió amb el fototransistor, de tal manera que depenent de la superfície en que incideixi, el raig serà reflectit o no, i això dependrà del color de la superfície.

Utilitza aquesta longitud d'ona perquè quan es troba amb un material de color negre la ona no és reflectida i per tant el fototransistor no condueix, i quan hi ha color blanc si que condueix. De manera que quan detecti color blanc s'obté un 0 lògic i quan detecti negre s'obté un 1. L'obtenció d'aquests valors són si es fa la lectura des de el col·lector del fototransistor.

Per tal de aconseguir un correcte funcionament d'aquest dispositiu, cal que estigui casi tocant el terra, ja que detecta a pocs mil·límetres de distància.



2.1.2. Sensor de distancia – SHARP® GP2Y0A41SK0F

Per a la detecció del robot contrari s'utilitzarà un sensor de distancia que utilitza la tecnologia dels infrarojos. El sensor utilitzat és el GP2Y0A41SK0F de la marca Sharp®. Aquest sensor genera una tensió proporcional a la distancia obtinguda. El marge de mesura va des de els 4 cm fins als 50 cm, amb aquesta distancia ja és suficient per a detectar el robot contrari, ja que si es troba a més de 50cm, està massa lluny per a atacar-lo.

A la figura 2 es pot observar la relació entre la distancia i el voltatge generat per el sensor, es pot apreciar que la relació no és lineal, per tant, seria recomanable linealitzar-la per software en cas que fos necessari, però en aquesta aplicació no ho serà, a l'apartat de software s'explica el perquè.

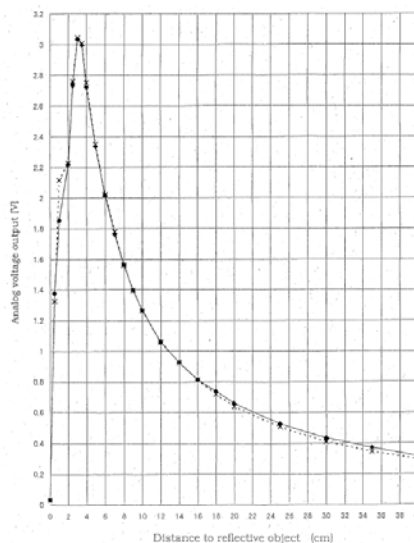


Figura 2. Relació del voltatge de sortida amb distancia-

L'utilització d'aquest sensor es força senzilla, només té tres pins de connexió:

1. Vo. Senyal de sortida, senyal a mesurar
2. GND
3. Vcc. Alimentació

Per tant, l'únic que s'ha de fer és convertir la senyal analògica de sortida del sensor en digital per tal que el μC el pugi llegir.

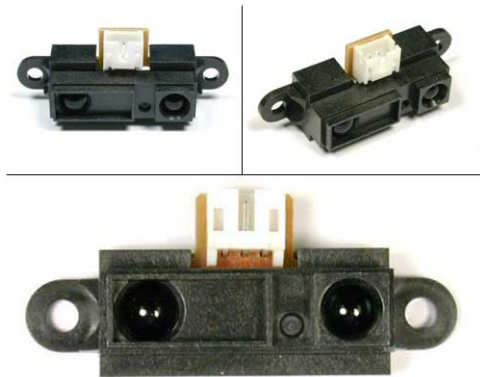


Figura 3a. Vistes GP2Y0A41SK0F

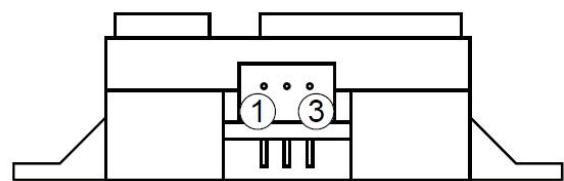


Figura 3b. Connexions GP2Y0A41SK0F

Tot i que el *datasheet* diu la distància màxima de detecció és de 10cm, s'ha comprovat que es sensor es capaç de detectar fins a 1 cm, a continuació s'adjunta la taula que ho demostra.

Taula 1. Comprovació de detecció de 20cm a 1cm. Valors reals obtinguts.

DISTANCIA	VOLTATGE, Vo (V)
20cm	0,61
15cm	0,8
10cm	1,2
8cm	1,5
6cm	2
5cm	2,3
4cm	2,7
3cm	2,9
2cm	2
1cm	1,8

Per a més detall sobre aquest component, consultar el *datasheet*, el qual està al annex del projecte.

2.2. Motors i control

Els motors són els encarregats de generar el moviment del robot, es necessari la utilització de dos motors ja que el moviment de les rodes és independent l'un de l'altre, ja que sinó no seria possible fer girar el robot.

2.2.1. Motor C-6061

Els motors utilitzats són de corrent contínua, de la marca Cebek i el model és el C-6061. Són un motors bastant petits els quals incorporen un reductor de velocitat, ja que és necessari reduir-la, per a poder tenir més parell i a la vegada una velocitat més controlada.

El reductor de velocitat està format per un seguit d'engranatges, els quals s'han de muntar manualment per tal d'obtenir la reducció requerida. En aquests cas s'ha fer el muntatge de la reducció de 9:1. També es pot realitzar el muntatge de reducció de 27:1, el qual tindrà un parell molt més elevat, però per a realitzar les proves de programació s'ha utilitzat el de 9:1 perquè el robot és més ràpid i podem veure si el programa és ràpid en respondre.

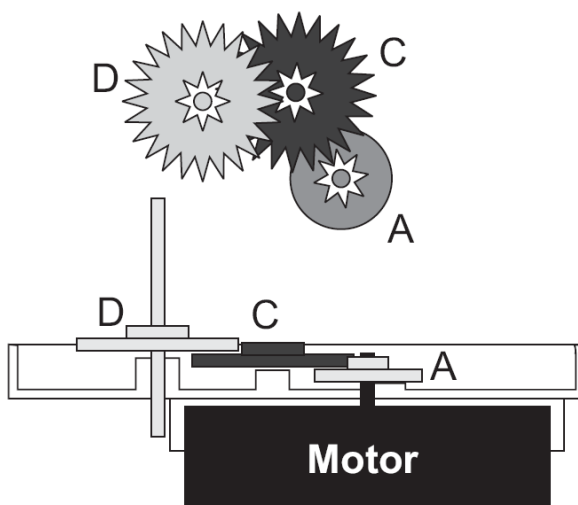


Figura 4. Motor + reductor 9:1

Però per a poder controlar aquests motors es necessari disposar d'una electrònica de control i a la vegada de potencia, ja que és la part del circuit que consumeix més intensitat. Per a poder realitzar aquesta funció existeix un xip integrat, el L293D.

2.2.2. L293D

L'293D és un driver quàdruple que permet alimentar 4 motors o 2 motors amb un pont en H. El pont en H permet controlar la polaritat d'alimentació del motor, de tal manera que permet controlar el sentit de gir del motor.

L'L293 està bàsicament dissenyat per al control de motors CC, ja que pot arribar a proporcionar un corrent molt més elevat que el que pot proporcionar el microcontrolador, ja que si el motor s'alimentés directament del μC , es podria arribar a cremar.

Aquest dispositiu disposa a cada sortida el següent circuit:

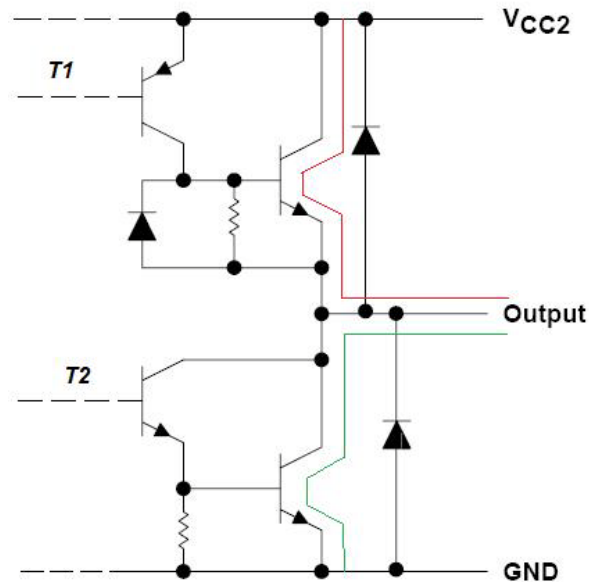


Figura 5. Circuit intern del L293D

De manera que cada sortida pot tenir el valor de V_{cc2} o de GND, solament cal governar els 2 transistors T1 i T2. El diode que es pot veure entre en joc quan es connecten dues sortides al mateix dispositiu, que és el que es vol fer, d'aquesta manera s'obté un pont en H.

A continuació es mostra la connexió d'un motor en pont en H, però la imatge fa referència al component L293, el qual no incorpora díodes, però en el nostre cas, s'ha utilitzat el L293D, on no és necessari posar els díodes, ja que estan incorporant en l'encapsulat. Però s'ha optat per mostrar aquesta imatge per tal de poder entendre el funcionament del component.

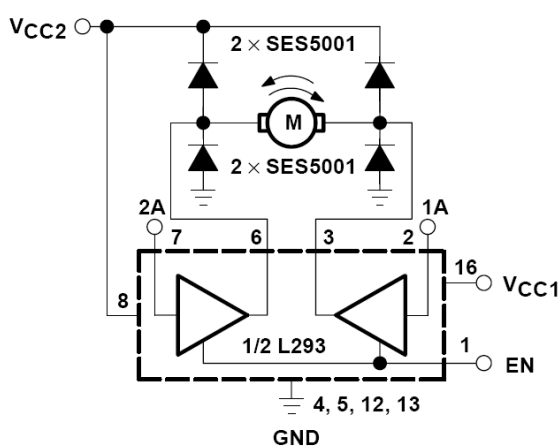


Figura 6. Connexionat L293

EN	1A	2A	FUNCIÓ
H	L	H	Gira a la dreta
H	H	L	Gira a l'esquerra
H	L	L	Parada de motor
H	H	H	Parada de motor
L	X	X	Parada de motor

Taula 2. Control del L293D

Aquest dispositiu és més que suficient per al motor utilitzar, ja que segons el *datasheet*, el motor a màxima eficiència consumeix 55mA i el component L293D per arribar a proporcionar fins a 600mA, així que no hi haurà cap problema.

2.3. Microcontrolador ATMEGA16

El microcontrolador(μC) és el cervell del robot, és l'encarregat de interconnectar tots els components que formen el robot, es pot dir que és la part més important del robot, ja que sense ell no funcionaria res.

El microcontrolador utilitzat és l'ATMEGA16 de la marca Atmel® i de la família dels AVR. L'elecció d'aquest microcontrolador i no un altre ha estat bàsicament perquè ja hi havia treballat anteriorment i perquè ja tenia tota l'electrònica de programació d'aquest, que és un kit docent anomenat STK500. El qual serà utilitzar per a connectar el robot amb l'ordinador i d'aquesta manera poder programar el microcontrolador amb el programa desitjat.

Cal reconèixer que aquest microcontrolador és massa potent per a l'ús que li volem fer, ja que amb qualsevol μC més senzill seria útil i suficient, però com ja s'ha dit anteriorment, s'ha utilitzat aquest degut a la seva disponibilitat.

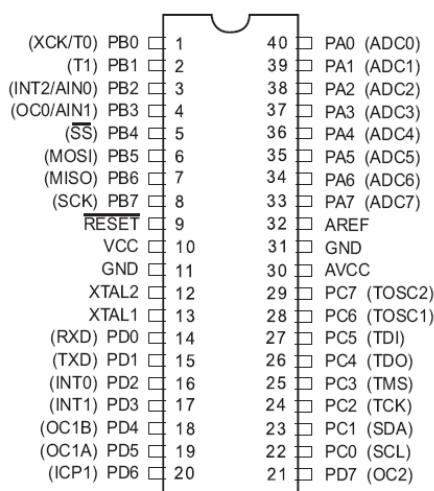


Figura 7. Pinout Atmega16

Com es pot observar en la figura 6, el port A del microcontrolador hi ha un convertidor analògic-digital, el qual ens serà molt útil alhora d'analitzar les senyals dels sensors de distància, d'aquesta manera ens estalviem la utilització d'un convertidor extern i a la vegada es guanya espai en la placa base.

Una característica important a comentar és que tots els ports incorporen resistència de pull-up interna, de manera que no es necessita la utilització de driver al connectar LEDs o altres components que requereixin força intensitat.

CAPÍTOL 3: CIRCUITS ELECTRÒNICS

En aquest capítol s'explicaran tot el conjunt de circuits electrònics els quals formen la placa base del robot, i tota la connexió del diferents sensors i motors.

3.1. Alimentació

L'alimentació del circuit ha de ser de 5Vcc, ja que tant el microcontrolador, com els motors i com els circuits TTL utilitzats, s'alimenten amb 5Vcc, per tant l'opció més adequada és utilitzar el regulador de tensió 7805, el qual proporciona 5Vcc a tot el circuit i de manera força estabilitzada.

El regulador LM7805 per a poder funcionar correctament necessita un valor de tensió bastant més alt de 5V, el *datasheet* recomana que sigui a partir del 7-8V fins als 25V. En aquest cas s'ha optat alimentar-lo a 9V degut a la existència de piles de 9V.

Però al cap de realitzar algunes proves, veiem que els robots no tenen massa potencia i que la durada de les bateries es molt curta, per això s'ha optat per connectar en paral·lel 3 bateries de 9V, d'aquesta manera augmenta la durada de la bateria i s'obté una intensitat superior.

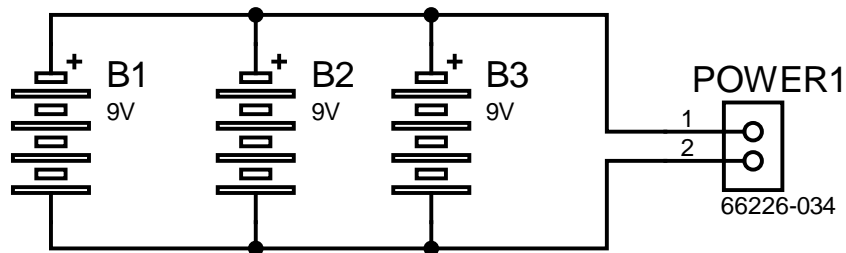


Figura 8. Connexió de bateries en sèrie

Un cop solucionat el tema de les bateries, cal explicar la connexió del LM7805, el qual no pot anar sol, es recomanable i més en aquest cas, la utilització de filtres de senyal, ja que si hi hagués algun pic de corrent podria fer mal al microcontrolador o penjar el programa, i faria perdre el combat. També cal dir que és molt important, ja que en els motors hi ha una constant commutació de la polaritat d'alimentació i això pot genera pics importants, per tant cal estar ben protegits.

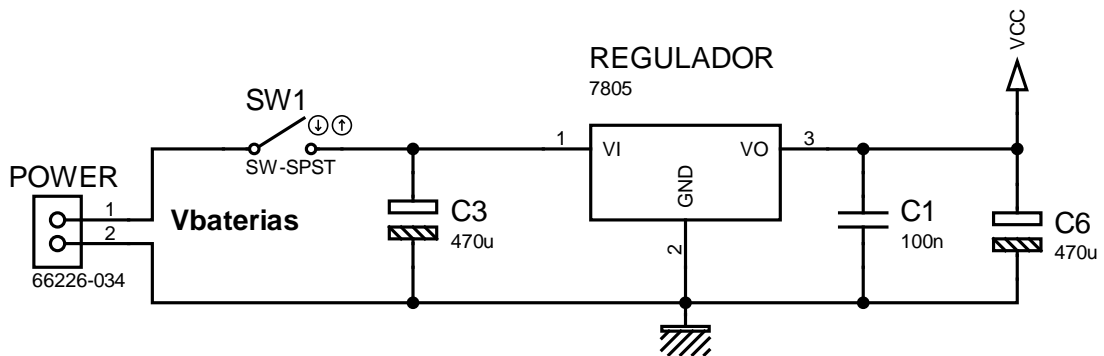


Figura 9. Circuit d'alimentació amb regulador LM7805

Tal i com es pot apreciar a la figura 8, hi ha condensadors tant a l'entrada com a la sortida del regulador. Es pot observar que a la sortida hi ha dos condensadors els quals tenen valors força diferents, això es per a poder filtrar un ventall més ampli de freqüències de pics, els valors d'aquests condensadors són els valors típics, ja que no hi ha un valor en concret que sigui el correcte.

A l'esquema es pot observar una sortida amb l'etiqueta Vcc, això vol representar que tot el circuit s'alimentarà des de aquest punt.

Taula 3. Components del circuit d'alimentació

SW1	Interrupctor general del circuit
C3	Condensador electrolític de 470µF de 35V
C1	Condensador de plàstic de 100nF
C6	Condensador electrolític de 470µF de 16V
REGULADOR	Regulador de tensió LM7805

3.2. Circuits dels sensors

Els circuits dels sensors són bàsicament per connectar el sensor amb el microcontrolador, de manera que depenent del sensor, serà de manera directa com és el cas del sensor de distància o s'haurà d'ajuntar un petit circuit, com és el cas del sensor de línies.

3.2.1. Sensor de distància

El sensor de distància SHARP® GP2Y0A41SK0F té una electrònica interna, gràcies a la qual, podem connectar el sensor directament al microcontrolador, de manera que és força simple d'utilitzar.

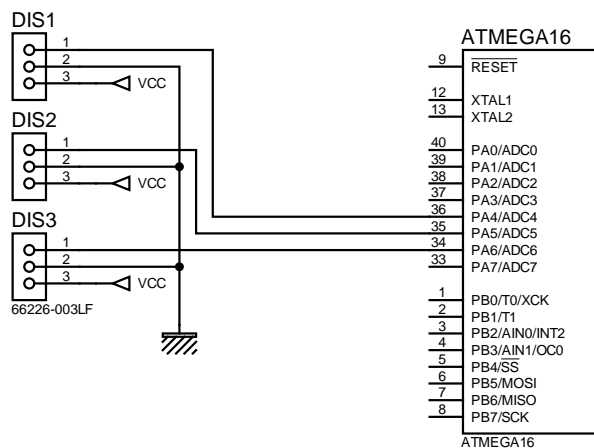


Figura 10. Circuit sensors distància

Com es pot veure a la figura, l'únic que cal fer és alimentar el sensor i connectar la senyal de sortida al convertidor analògic-digital del microcontrolador, on posteriorment serà convertida en digital i analitzada. A l'apartat de programació s'explicarà com es realitza aquesta conversió i anàlisi de la senyal obtinguda pel sensor.

En aquest cas només s'ha utilitzat un connector per tal de connectar el sensor amb la placa base mitjançant un cable tripolar extern

3.2.2. Sensor detector de línies

El sensor CNY70 tot i estar integrat dins d'un encapsulat i semblar un sensor intel·ligent com el de distància, aquest no ho és. L'única electrònica que té incorporada és un led infraroig i un fototransistor sensible als infrarojos, de tal manera que serà necessari connectar l'electrònica necessària per obtenir el correcte funcionament.

Segons el *datasheet* el led s'ha d'alimentar com a màxim a 50mA, per tant si es connecta amb una resistència de 220Ω s'obté 22mA i és més que suficient per a funcionar correctament. Pel que fa a la connexió del fototransistor, s'ha optat per posar una gran resistència, per tal de tenir una gran diferència de voltatge.

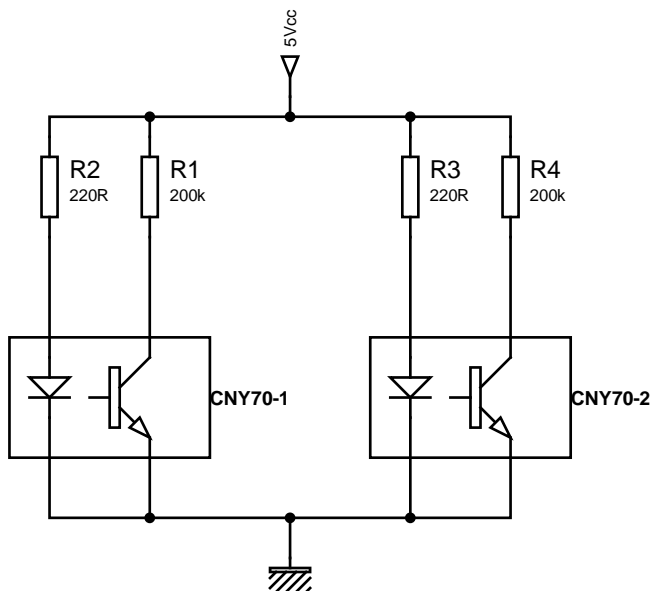


Figura 11. Connexió CNY70

A la figura 10 es pot observar tal i com quedaria la connexió del sensor si es pogués posar dins la placa base, però com que això no es possible, ja que el sensor ha de ser externs, s'han d'extreure 3 punts de tal manera que les resistències pugin estar igualment a la placa base. Amb 3 punts es necessari tot i que el sensor te 4 pins, però com que dos d'ells són les masses, es poden ajuntar.

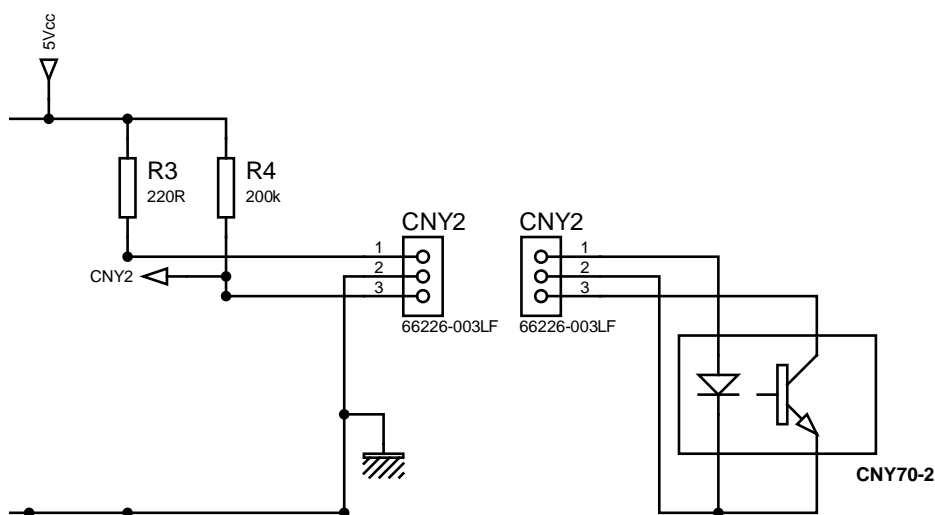


Figura 12. Figura entenedora

En la figura 12 es pot observar la relació entre els punts de connexió del sensor i del que s'extreu. El circuit de la part dreta del la figura 12 està al costat del

sensor, és a dir, és un circuit extern a la placa base. En la següent figura es pot apreciar el resultat final.

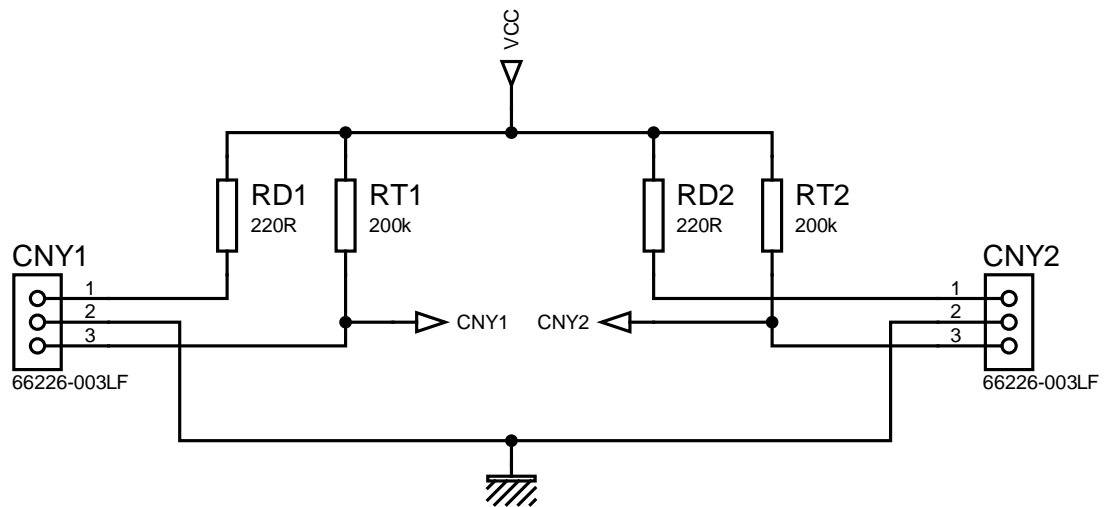


Figura 13. Connexió real CNY70

A la figura 13 s'observa el resultat final de la connexió dels sensors, i la senyal CNY1 i CNY2, és on s'obté el senyal la qual varia de la següent forma:

- Si detecta color blanc: 0,2V
- Si detecta color negre: 3,5V

Amb aquests valors s'ha decidit connectar la senyal directament al pin de les interrupcions externes, de tal manera que quan detecti blanc generarà una interrupció el programa.

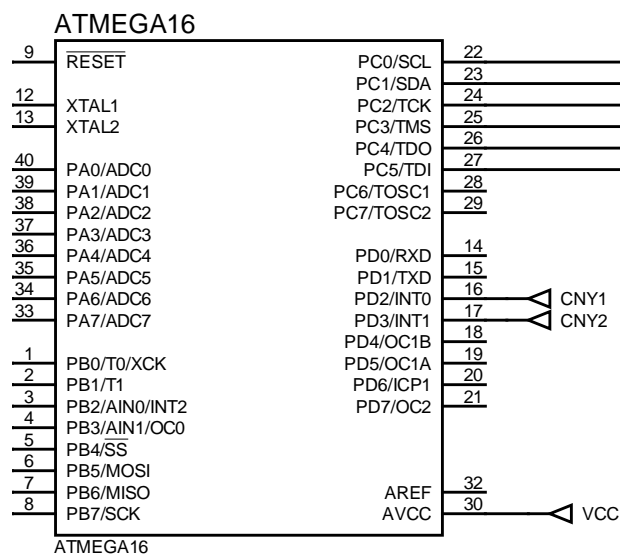


Figura 14. Connexió del CNY70 al µC

3.3. Motors

El circuit dels motors com s'ha comentat anteriorment està bàsicament governat pel circuit integrat L293D, des de el qual es pot controlar ambdós motors a la vegada.

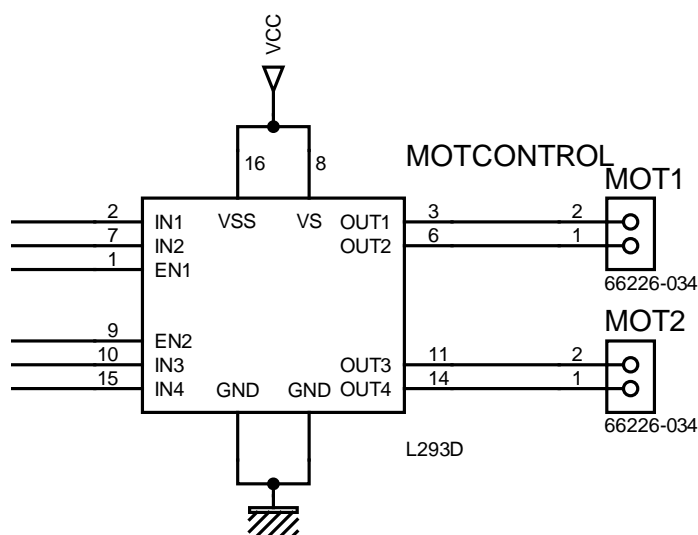


Figura 15. Circuit dels motors

D'aquesta manera es poden controlar els dos motors des de el mateix component. Només cal governar des de el microcontrolador els pins IN1, IN2, EN1, EN2, IN3, IN4. A la següent taula es mostra les possibilitats de control.

Abans cal comentar que els pins EN1 i EN2 són simplement per habilitar o inhabilitar l'alimentació dels motors, per tant si es posen a "0", desactiva qualsevol combinació que pugin tenir els altres pins.

Taula 4. Taula de control del Mot1

EN1	IN1	IN2	ACCIÓ
0	X	X	Motor parat
1	0	1	Gir cap a la dreta
1	1	0	Gir cap a l'esquerra
X	1	1	Motor parat
X	0	0	Motor parat

La taula de control del Mot1 és exactament igual que la Taula 4 però amb els bits EN2, IN3 i IN4.

Un cop ja s'ha definit totes les possibilitats de moviment dels motors, només fa falta connectar els pins de control al microcontrolador, es farà al port C, ja que fins ara estava lliure.

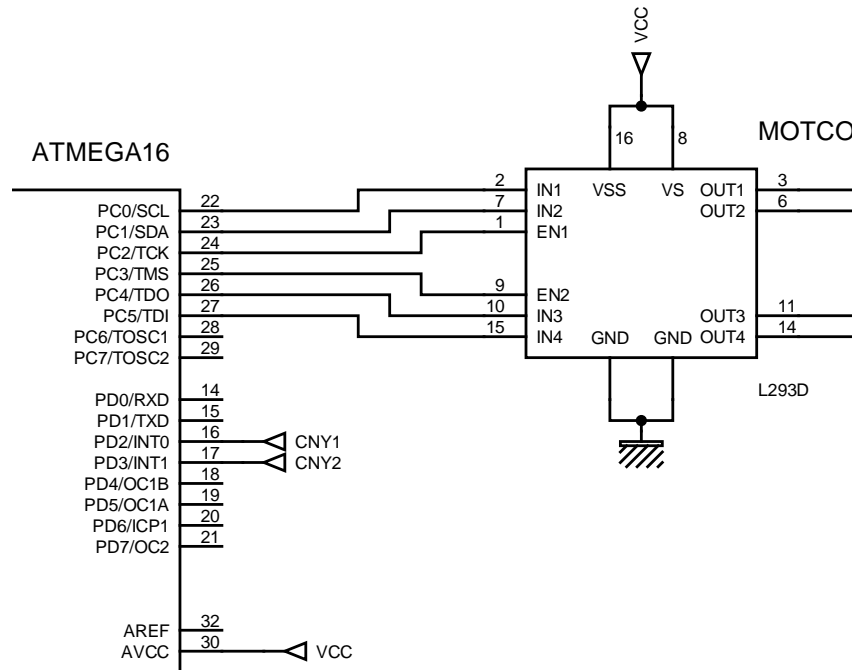


Figura 16. Connexió dels motors al µC

3.4. Circuit del microcontrolador

Per tal de que el microcontrolador funcioni només necessita l'alimentació de 5Vcc, i el oscil·lador.

3.4.1. Oscil·lador de quars

L'oscil·lador és un cristall de quars el qual oscil·la a una freqüència constant, de manera que el microcontrolador utilitza aquesta freqüència constant per generar els seus cicles interns. De manera que com més alta sigui la freqüència del cristall més ràpid serà el microcontrolador, però com és normal, cada microcontrolador te els seus límits.

El nostre microcontrolador pot arribar a treballar sobre els 30MHz, però s'ha utilitzat el valor del cristall de 3,8684 MHz, per dues raons, una és per que aquest nombre és fàcil d'utilitzar com a freqüència a l'hora d'utilitzar els *timers* del microncotroladors, a l'apartat de programació s'entendrà perquè, i l'altre raó és degut a la possessió d'aquest.

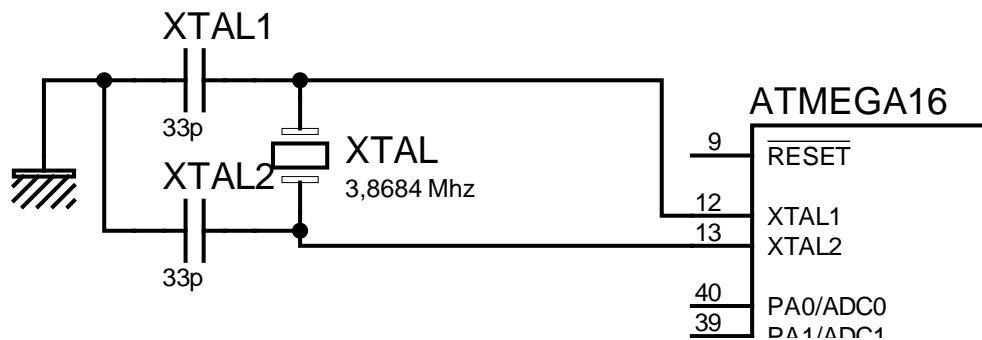


Figura 17. Circuit d'oscil·lació del μ C

El cristall de quars, tal i com es pot veure a la figura 17, es connecta entre els pins del microcontrolador XTAL1 i XTAL2, però sempre es recomana connectar 2 condensadors de baixa capacitat entre el cristall i massa. La funció d'aquests condensadors és eliminar possible soroll de tal manera que estableixen l'oscil·lació del cristall. En el nostre cas s'ha utilitzat dos condensadors ceràmics de 33pF.

3.4.2. Boto d'inici

El boto d'inici és els que s'haurà de pulsar per tal que el robot comenci el combat, la senyal d'aquest es pot portar a qualsevol port que quedi lliure del microcontrolador, ja que simplement s'haurà de llegir si hi ha VCC o GND i actuar en conseqüència.

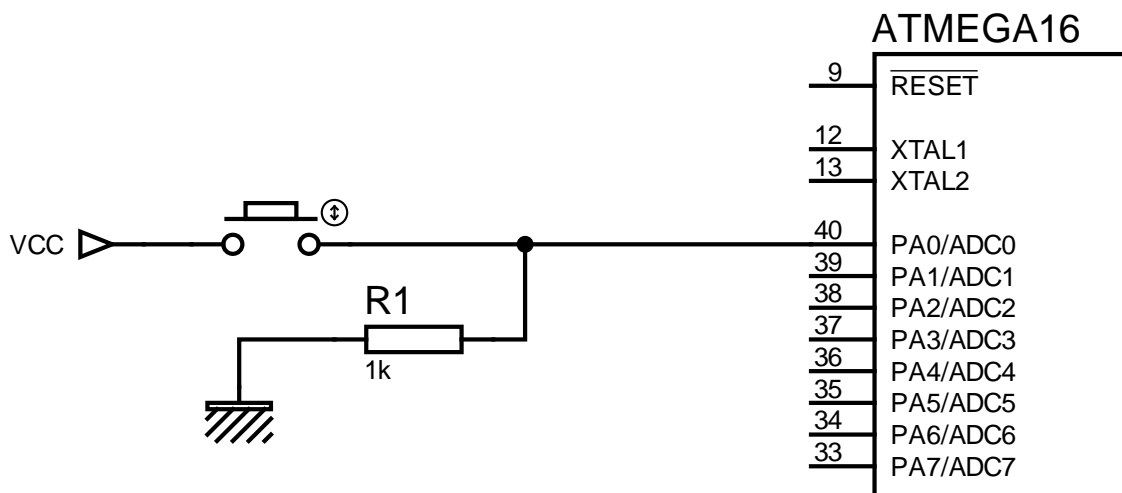


Figura 18. Polsador d'inici

Gracies a aquesta disposició al PA.0 sempre hi haurà GND menys quan es polsi el polsador, que hi haurà VCC. La resistència es posa per tal de que quan polsem no hi hagi una connexió directe entre VCC i Massa, ja que això seria terrible per el nostre circuit.

3.4.3. Leds

Per tal de saber que està passant al programa, s'han instal·lat 4 leds. La funció d'aquests leds és sobretot alhora de programar el robot, ja que indiquen en quin estat del programa està. També indiquen si el circuit està connectat o no, ja que el primer led sempre s'encén només donar alimentació al circuit.

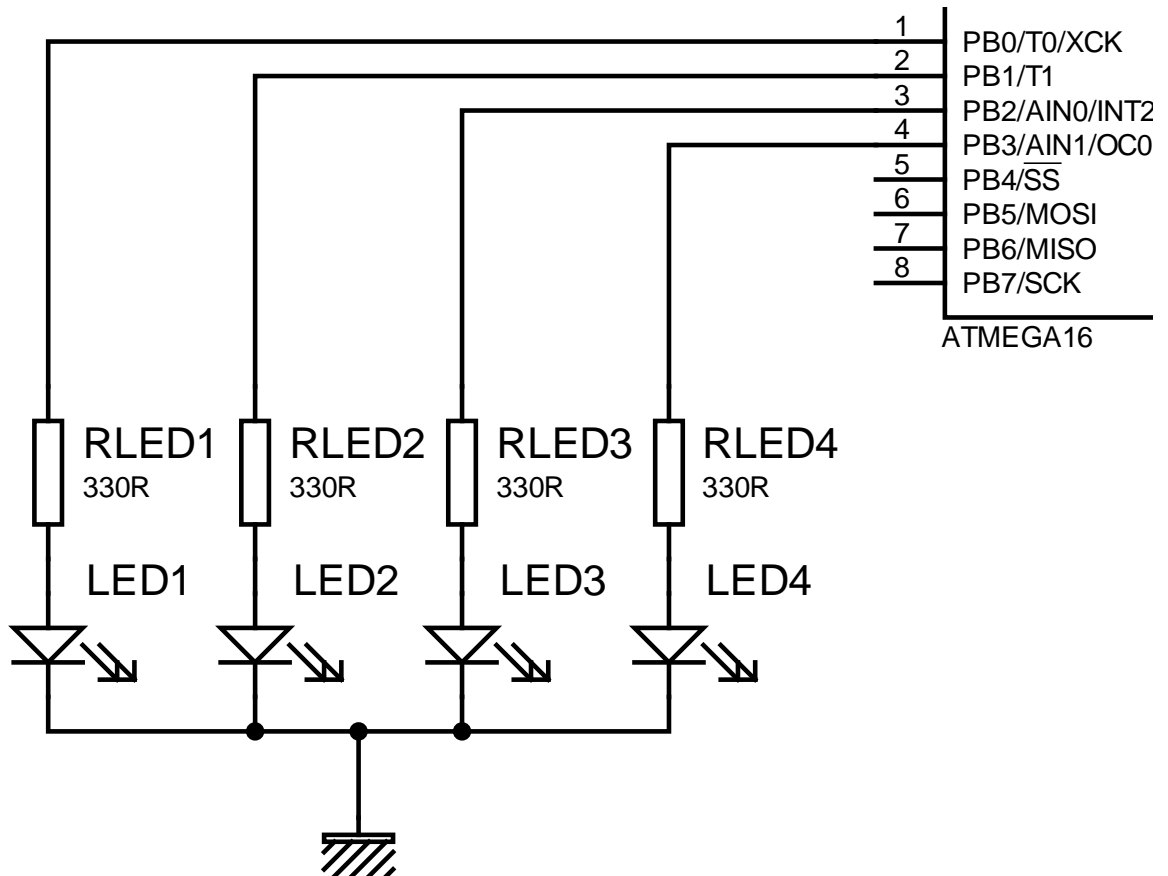


Figura 19. Circuit de leds

La resistència de cada Led és de 330 perquè els leds s'alimenten sobre els 20mA.

$$I = \frac{V}{R} = \frac{5V}{330\Omega} = 15mA$$

Cal recordar que el microcontrolador ATmega16 incorpora resistències de *pull-up* internes, això significa que la corrent no s'extreu del microcontrolador, sinó del circuit d'alimentació, al disposar de *pull-up* no fa falta connectar un buffer per connectar els leds.

3.5. Circuit de programació

El circuit de programació és el més complex, tot i que la part que es veu en aquest projecte és l'última part, ja que la placa que s'ha dissenyat no es pot connectar directament a l'ordinador, requereix una electrònica externa força avançada. En la placa base simplement es fa al connexió dels pins de programació al microcontrolador, per tant, abans de centrar-nos amb el disseny del circuit de la nostra placa, s'explicarà el funcionament o els trams per on ha de passar la senyal fins arribar al microcontrolador.

El primer problema de programació que presenten aquesta família de microcontroladors és que estan dissenyats per a ser programats mitjançant el port paral·lel del ordinador, el qual ja no l'incorporen els últims ordinadors, i molt menys els portàtils. Per tant, el que antigament es solucionava amb un parell de resistències ara requereix d'una electrònica més complexa.

Per a poder solucionar aquest problema s'ha utilitzar un kit d'estudiant que ofereix Atmel, anomenat STK500, el qual permet programar tot tipus de microcontroladors de 8bits AVR.



Figura 20. STK500

Abans de continuar cal dir que s'ha utilitzar aquest programador degut a l'antiga possessió d'aquest alhora de realitzar el projecte, si no hagués estat així, existeixen programadors més compactes i més econòmics.

Aquest kit d'auto aprenentatge és molt útil però té un petit problema, que la connexió amb l'ordinador és a mitjançant el port sèrie, el qual els ordinadors d'avui dia i sobretot els portàtils ja no l'incorporen. Per tant caldrà utilitzar un convertidor de senyal USB a la senyal RS-232 del port sèrie.

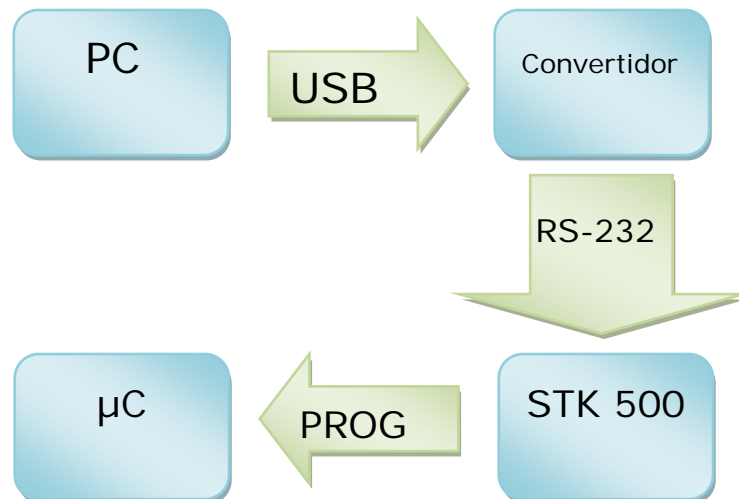


Figura 21. Transformació de la senyal de programació

La placa STK500 permet programar fora de la placa, ja que deixa accessible per l'usuari els pins de programació i facilita un cable per a poder-ho connectar exteriorment. Els pins que proporciona són els que es mostren a la figura 19.

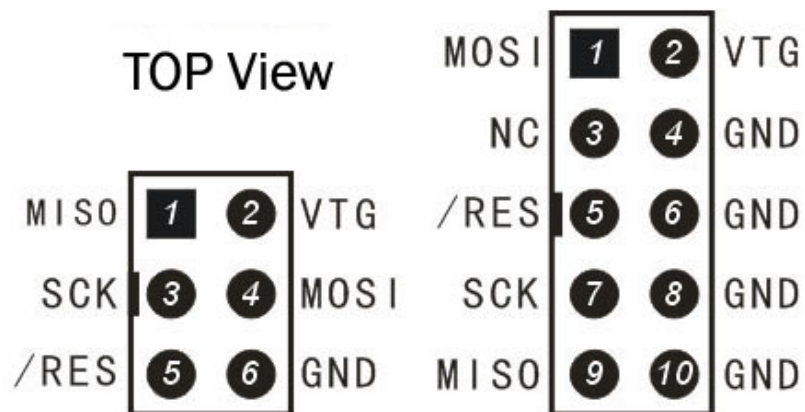


Figura 22. Pins de programació

Les dues opcions són iguals, simplement ho ofereixen en mode 6 pins o 10 pins, en el nostre cas s'ha utilitzat el mode de 6 pins, per tal de minimitzar l'espai de la nostra placa.

Un punt molt positiu d'aquesta solució adoptada, és que es podrà programar el circuit sense haver de extreure el microcontrolador de la placa, això estalviarà molt de temps i farà augmentar l'eficiència del circuit dissenyat.

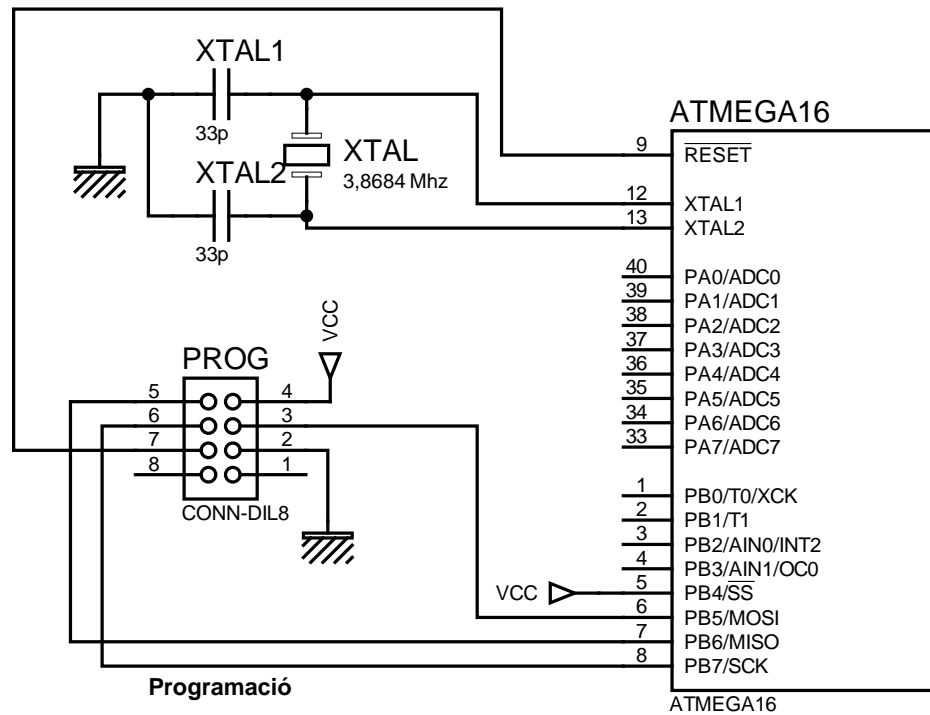


Figura 23. Circuit de programació

És molt important que estigui connectat el cristall al microcontrolador alhora de programar-lo, ja que sense oscil·lador no funciona. Per això s'ha mostrat el cristall justament amb el circuit de programació.

3.6. Disseny de la PCB

La PCB és el la placa de circuit imprès, la qual recull totes les connexions dissenyades amb el simulador. La PCB que s'utilitzarà serà de una cara, es a dir, que només hi haurà connexions per una cara, per tant les pistes no es podran encreuar, i si fos el cas, s'haurà de crear un pont.

En la figura 24 es pot observar la les pistes i a la vegada els components, però a l'hora de crear el fotolit s'ha de imprimir només les pistes. Aquesta vista es superior, és per això que les lletres es llegeixen al revés. A continuació es mostra la PCB real, la que s'ha implantat.

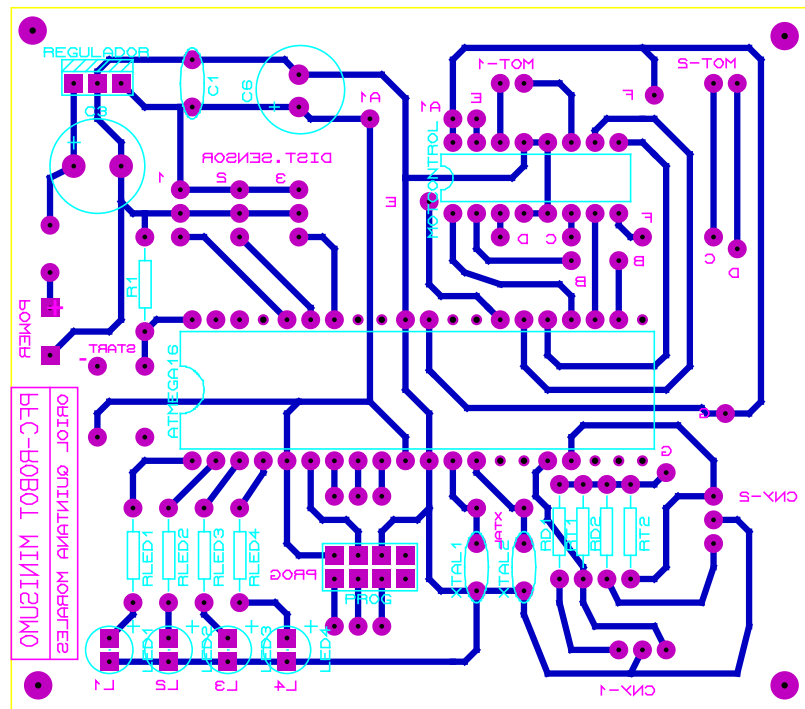


Figura 24. PCB

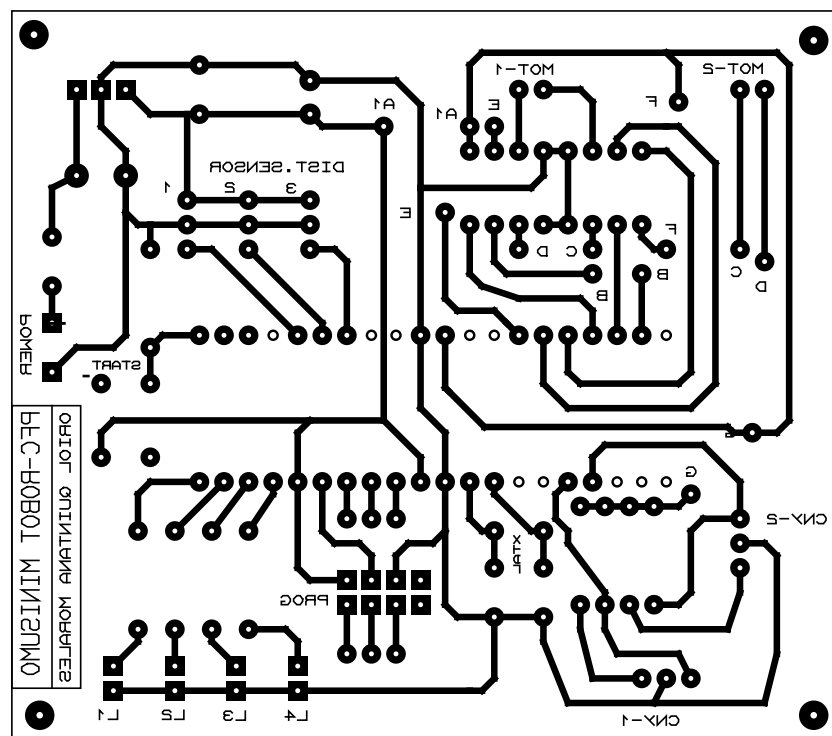


Figura 25. PCB real

Les lletres que apareixen soles com ara, F,G,B fan referència a punts que s'han de realitzar externament, a través de petits cables, ja que les pistes no es poden encreuar.

A continuació es mostra una previsualització en 3D de com queda el circuit.

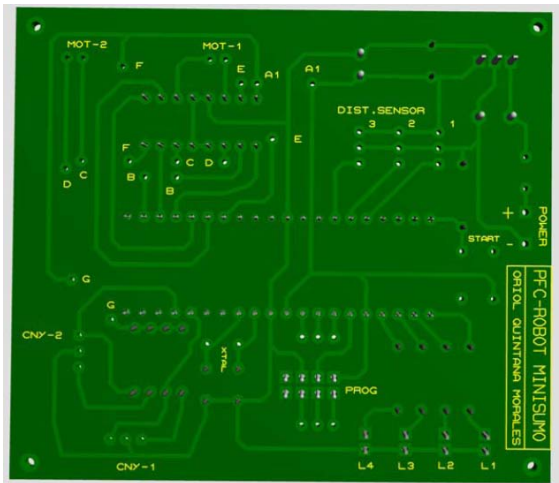


Figura 26a. PCB vista inferior

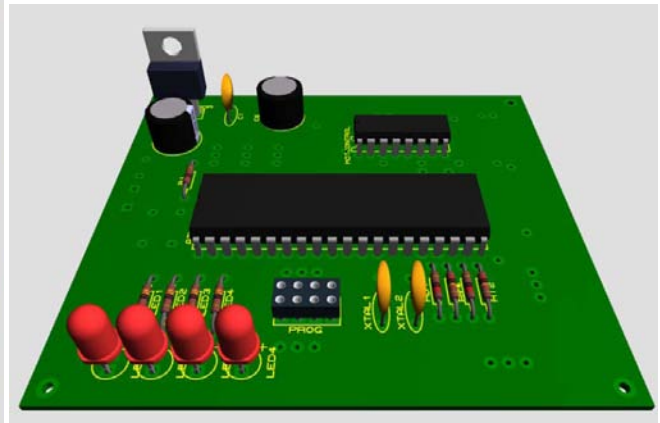


Figura 26b. PCB vista superior

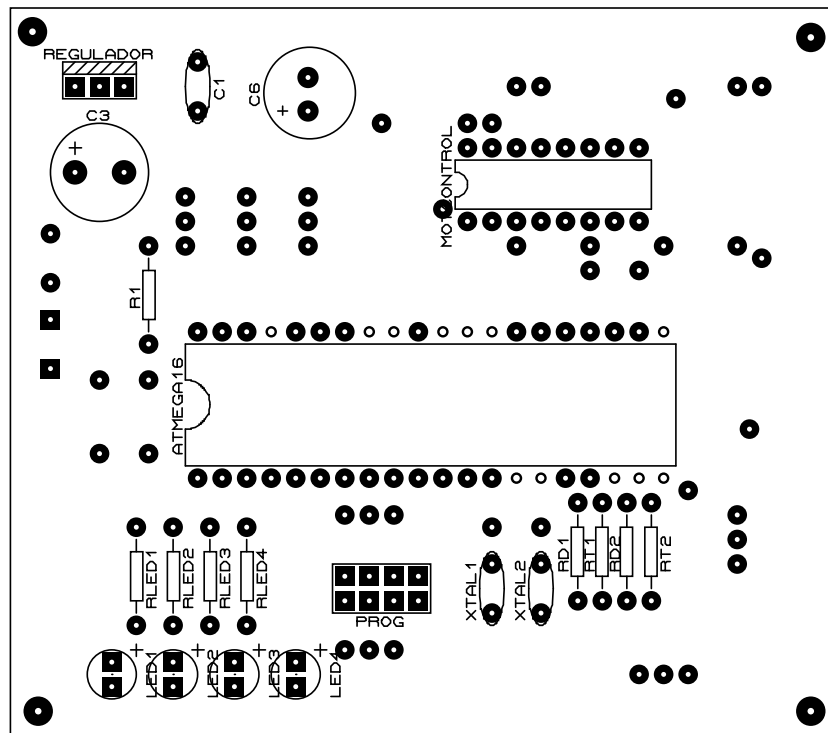


Figura 27. Material i forats PCB

A continuació es mostrarà el llistat de material necessari per a elaborar la PCB completa, la numeració del material es segons el plànol de la figura 27. Abans cal comentar que falten tot el conjunt de connectors, els quals connecten la placa amb els sensors i motors. Els connectors no s'han dibuixat ja que el simulador no els tenia, i apart no són peces fonamentals per al correcte funcionament, es pot utilitzar qualsevol tipus mentre respecti el pas de pista convencional de 2,54mm.

Taula 5. Material PCB

Ref.	Component	Valor
Regulador	Regulador de tensió	LM7805
C3	Condensador	470 μ F de 35V, electrolític
C6	Condensador	470 μ F de 16V, electrolític
C1	Condensador	100nF de plàstic
MOT CONTROL	Pont en H	L293D
R1	Resistència	1k Ω \pm 5%
ATMEGA16	Microcontrolador	ATMega16
RLED1-2-3-4	Resistència	330 Ω \pm 5%
LED1-2-3-4	Led	Led vermell
Xtall1-2	Condensador	33pF ceràmic
RD1-2	Resistència	220 Ω \pm 5%
RT1-2	Resistència	200k Ω \pm 5%

CAPÍTOL 4: MECÀNICA

La part mecànica del robot és la més senzilla i alhora la menys treballada en aquest cas, ja que el projecte s'ha centrat en la part d'electrònica i programació, perquè és la part estudiada. Per tant s'intentarà solucionar aquest problema de la manera més ràpida i econòmica possible.

4.1. Estructura del robot

La mecànica del robot és bàsicament l'estructura del robot. Per tal d'obtenir una estructura fàcil d'elaborar i fàcil de modificar, s'ha utilitzat peces de LEGO®, la qual es força lleugera i per tant no hi haurà problemes d'excés de pes. El gran problema que té el LEGO® és que és molt fàcil de desmuntar, per tant és una característica no gaire bona alhora de crear un robot lluitador de sumo, per tant caldrà reforçar-la.

Les rodes utilitzades també són de la família LEGO®, però s'han utilitzat unes de força amples per tal d'obtenir un major fregament amb el terra i per tant ser tenir més resistència alhora de defensar-nos del robot contrari, ja que com més enganxat s'està al terra més difícil ho té l'altre robot per fer-te fora del tatami. A la figura 26c es pot observar l'amplada notable de les rodes utilitzades.

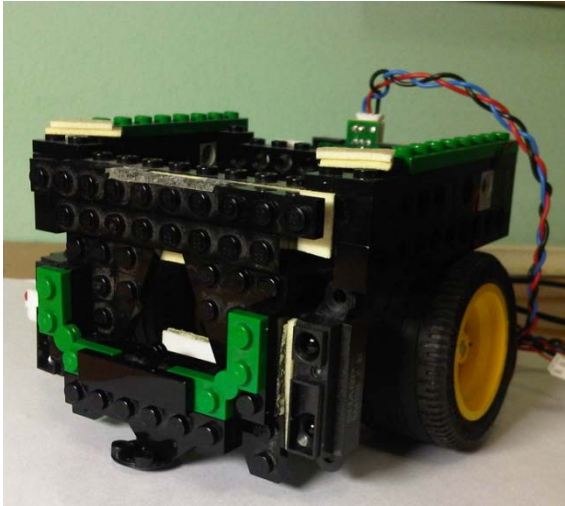


Figura 28a. Estructura de LEGO®

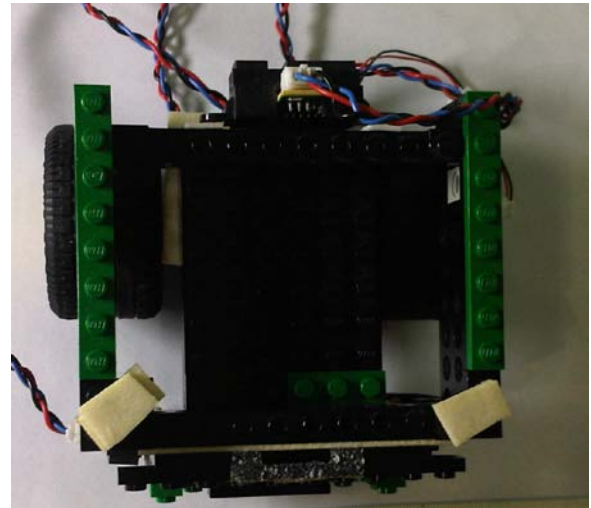


Figura 28b. Vista alçat

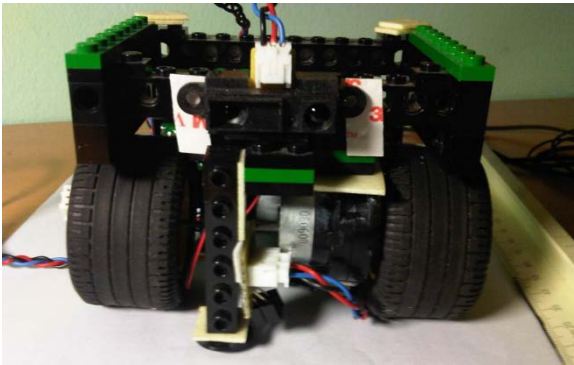


Figura 28c. Visió del darrera del robot

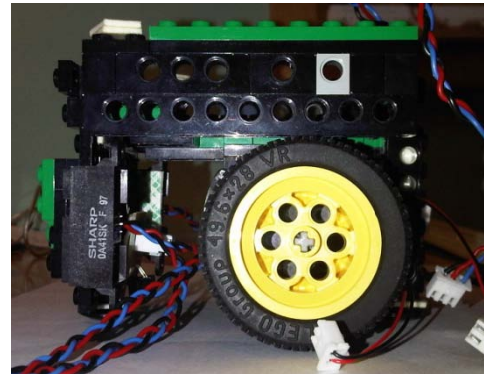


Figura 28d. Vista lateral

4.2. Fixació i localització de sensors i actuadors

Els sensors i els actuadors, motor, al no ser del mon de LEGO®, no encaixen fàcilment a l'estructura, per tant s'hauran de fixar algun que altre mètode de fixació. El mètode utilitzat és el loctite i cinta adhesiva ultra-forta de doble cara.

A continuació es mostrarà la disposició dels sensors i actuadors:

- Sensor de distància

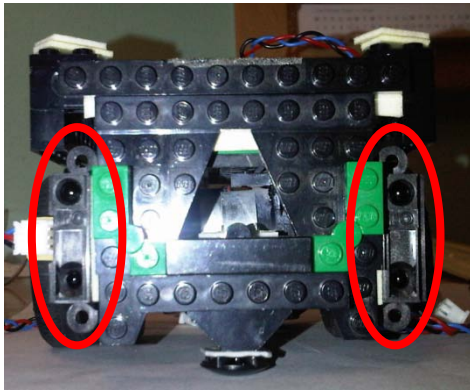


Figura 29a. Sensors frontals

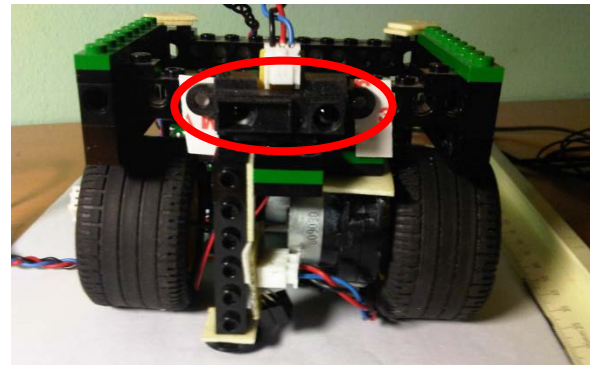


Figura 29b. Sensor darrera

Es posen dos sensors de distancia davant per tal de no perdre de vista el robot contrari, ja que d'aquesta manera es pot saber si l'hem deixat de detectar per la dreta o per l'esquerra. Al darrera se'n posa un, simplement per no perdre tanta visió.

- Detector de línies

Els detectors de línies han de estar casi tocant el terra, per tant, estaran a la part inferior del robot. Per tal de que el robot no surti del tatami de cap de les maneres, es col·locarà un detector a la part frontal i un altre al darrera. D'aquesta manera es totes les possibilitats de sortir del tatami controlat, ja que de costat no es pot moure el robot.

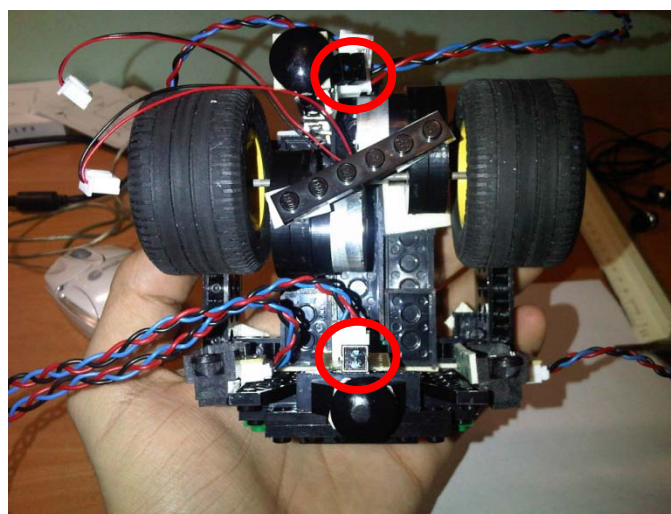


Figura 30. Sensors de línies

- Motors

Els motors són els més complicats de fixar a l'estructura, ja que són els que suporten tot el pes de l'estructura i estan sotmesos constantment a diversos tipus de forces, les quals fan que es desenganxin. Per això, cal fixar-los bé. En el nostre cas, s'han fixat per la part superior, però tal i com es veu a la figura 29, hi ha un reforç per sota, el qual fa que els dos motors no es belluguin ni s'inclinin cap al costat per culpa del pes.

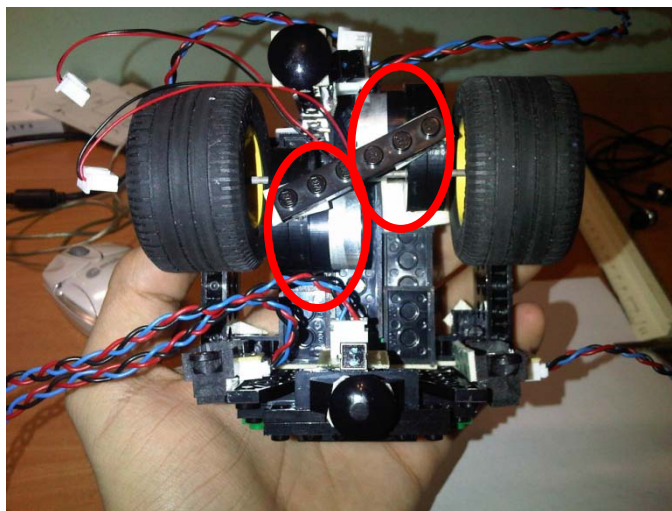


Figura 31. Motors

4.3. Plaques reforçants

Com que l'estructura Lego no és massa resistent, i molt fàcil de ser desmuntada, s'ha dissenyat una carcassa que recobreix tot el robot, per tal de fer-lo més resistent. Es pot dir que aquesta és la defensa del robot.

Les plaques s'han dissenyat de tal manera que recobreixen tot el robot i estan a poques dècimes de mil·límetre del terra. S'ha fet d'aquesta manera perquè molts robots de minisumo porten una espècie de pala davant, on els seu objectiu es aixecar el robot contrari per tal de que perdi adherència amb el terra i d'aquesta manera poder-lo treure'l del tatami. Però amb la carcassa que se lo a incorporat al nostre robot li serà molt difícil aixecar-nos-el al robot contrari.

Les plaques són de poliestirè transparent antireflexant de 2mm de grossor. S'han fet de manera que sigui fàcil la col·locació d'aquestes i l'extracció, ja que si s'ha de manipular el robot, que no creïn cap problema.

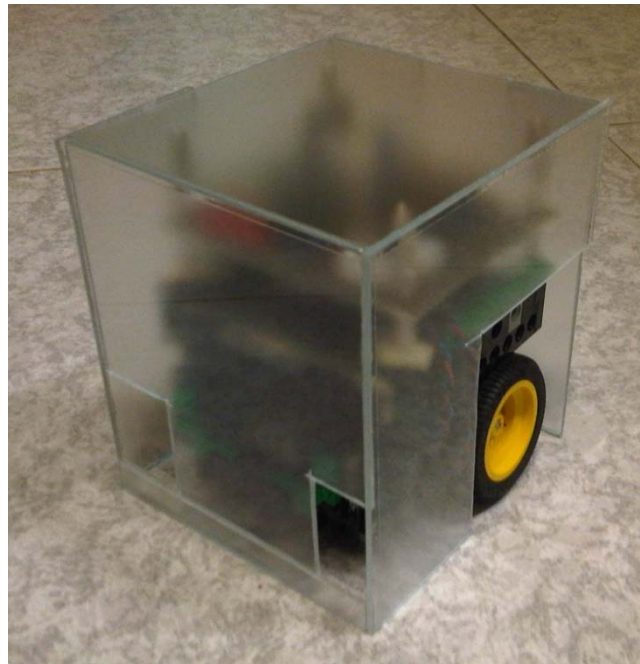


Figura 32. Plaques reforçants

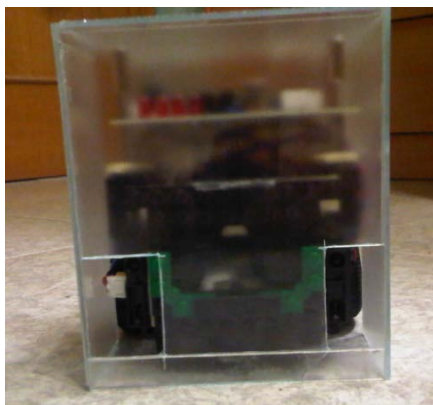


Figura 33a. Vista davant

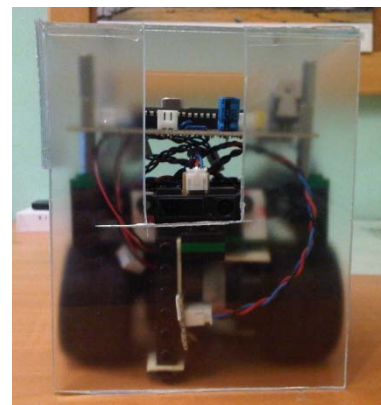


Figura 33b. Vista darrera

Només es deixen forats pels sensors, per les rodes i a un extrem per a poder accedir al boto d'inici, la resta està totalment tapat per les plaques protectores.

CAPÍTOL 5: PROGRAMACIÓ

Un cop construïda tota l'electrònica i la mecànica del robot, fa falta realitzar la unió d'aquestes dues parts, on s'encarrega l'informàtica/programació. Sense aquesta part es podria dir que el robot seria pràcticament inútil, ja que sense una mínima programació el robot no es mourà ni reaccionarà a cap factor.

En el cas dels robots minisumo, la programació acostuma ser el factor que determina la victòria dels combats, ja que la mecànica i l'electrònica és força similar en tots els robots d'aquest àmbit.

5.1. Pins de connexió

Abans de començar a programar, cal tenir molt clar a quin pin del microcontrolador està connectat cada un dels components. A continuació es mostraran tots els pins del μC i el component que se li connecta i el perquè de l'elecció.

- **PA0**

Al pin 0 del port A es connecta el pin de sortida del pulsador d'inici, el qual s'ha configurat com a NO (normalment obert), per tant al posar-lo hi haurà un 1 a l'entrada. S'haurà de llegir aquest pin per determinar el inici del programa.

PA0 - Input - Pulsador d'inici

- **PA4, PA5 i PA6**

En aquests 3 pins es connecten els 3 sensors de distància infrarojos que disposa el robot, s'han connectat aquí perquè són els pins que tenen accés al convertidor

analògic-digital intern del microcontrolador. Per tant cada vegada que es vulgui saber la distància a que es troba el robot oponent, s'hauran de llegir aquests valors.

PA4 - Input - Sensor distància dreta
PA5 - Input - Sensor distància esquerra
PA6 - Input - Sensor distància darrera

- **PB0, PB1, PB2 i PB3**

En aquests pins es connecten 4 leds indicatius, de manera que simplement s'haurà d'enviar un 1 a aquests pins per encendre el led. L'elecció d'aquests pins és degut a la distribució de la PCB, ja que per encendre Leds es pot utilitzar qualsevol pin de qualsevol port del microcontrolador.

PB0 - Output - LED 1
PB1 - Output - LED 2
PB2 - Output - LED 3
PB3 - Output - LED 4

- **PB4, PB5, PB6, PB7 i RESET**

Aquests pins són els encarregats de la programació del microcontrolador, en aquests cas, no hi ha opció de canvi de pins, ja que aquests són els pins específics i únics per a la programació.

PB4 - Input - VCC
PB5 - O/I - MOSI |
PB6 - O/I - MISO | > Programació per SPI
PB7 - O/I - SCK |
PIN9 - RESET

- **PC0, PC1 i PC2**

Aquests pins són els encarregats al control del motor nº1, són pins de sortida, els quals configuren un integrat extern TTL, per tant no hi haurà problema de comunicació entre ells.

PC0 - Input - IN1 |
PC1 - Input - IN2 | > Control motor 1
PC2 - Input - EN1 |

- **PC3, PC4 i PC5**

Aquests pins són els encarregats al control del motor nº1, són pins de sortida, els quals configuren un integrat extern TTL, per tant no hi haurà problema de comunicació entre ells.

PC3 - Input - EN2 |
PC4 - Input - IN3 | > Control motor 2
PC5 - Input - IN4 |

• PD2 i PD3

Aquests pins fan referència als detectors de línia blanca, aquests s'han configurat de manera analògica de manera que al detectar color blanc entregui 0V i al detectar negre uns 3,5V, de tal manera que es pot llegir con a senyal TTL. D'aquesta manera, s'han connectat ambdós senyals als pins de les interrupcions externes que ofereix el microcotrolador, de manera que al detectar color blanc, la resposta del robot serà immediata.

PD2 - Input - CNY1

PD3 - Input - CNY2

5.2. Diagrama de flux i pseudocodi

El diagrama de flux ens mostra de manera visual el funcionament del programa, és la forma més fàcil d'entendre programes no gaire complicats.

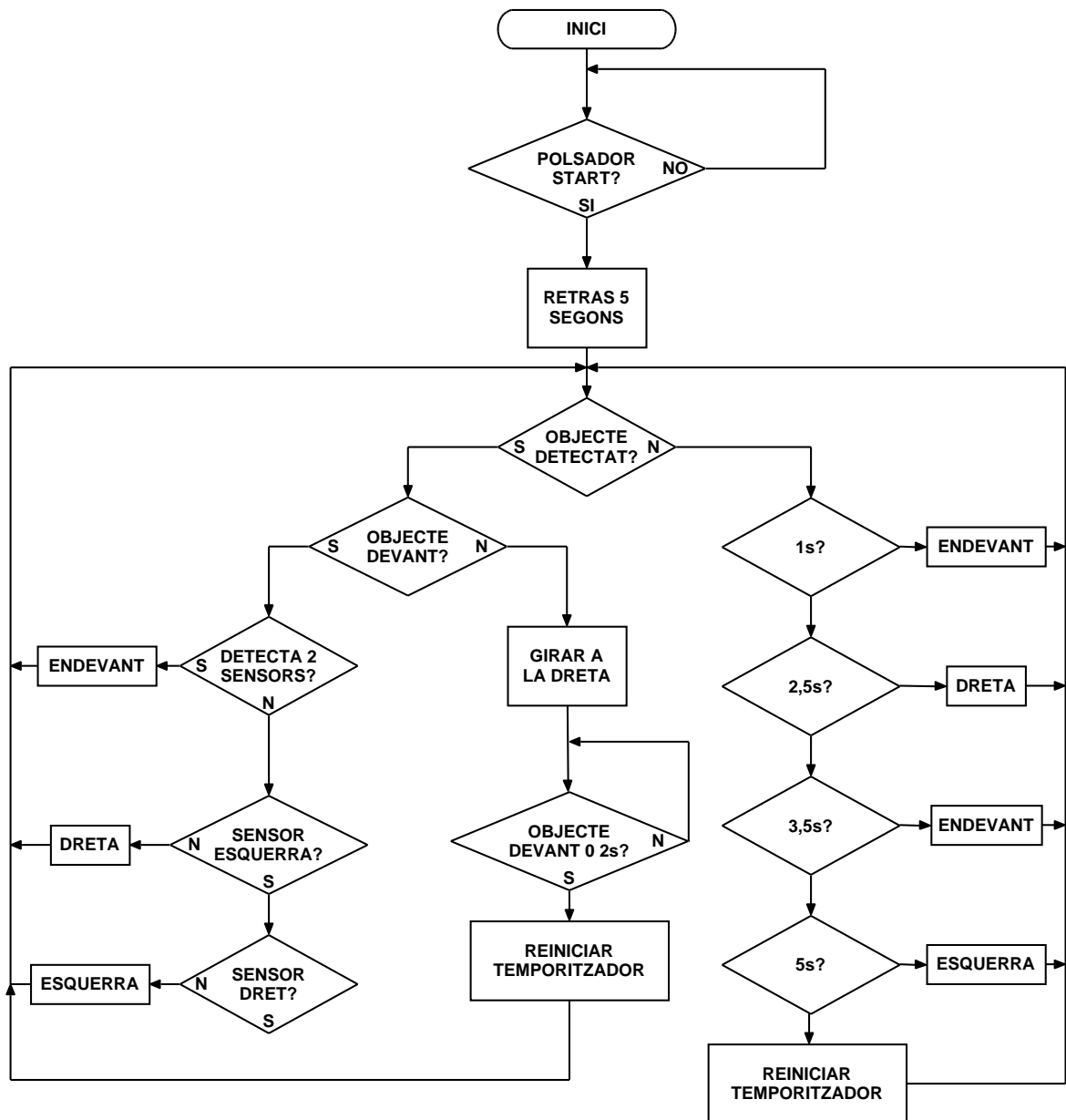


Figura 34. Diagrama de flux general

Apart del diagrama de flux general hi ha un que pertany al de les interrupcions, el qual no es pot mostrar al diagrama general ja que pot saltar en qualsevol posició de diagrama general.

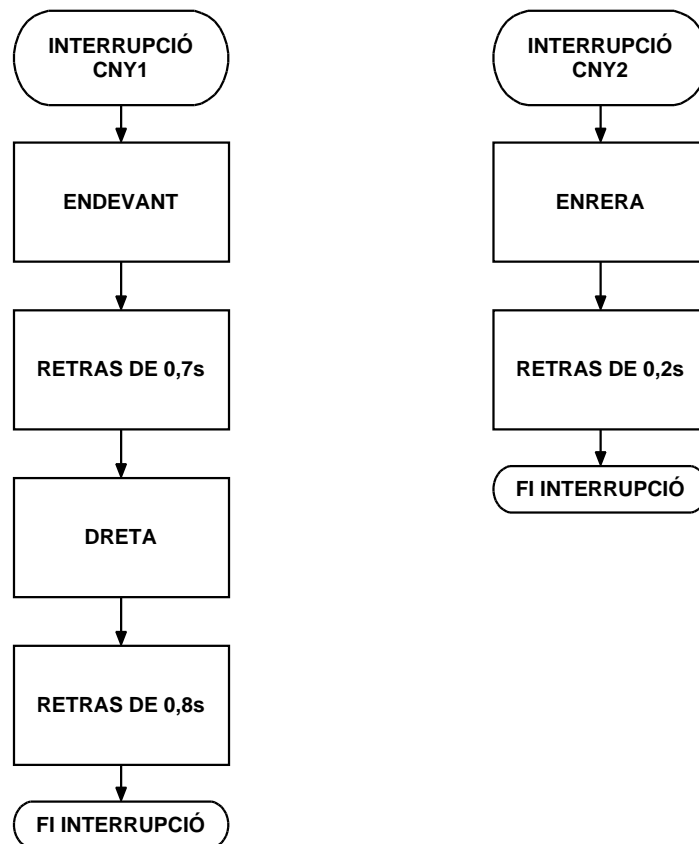


Figura 35. Diagrama de flux interrupcions

Aquests diagrames de flux donen a entendre el funcionament del robot de manera força clara, el problema és que són d'un nivell de programació massa alt per a poder transformar-lo directament amb llenguatge C, és per això que abans de entrar en la programació pura, es baixarà el nivell de programació i es mostrarà el programa en pseudocodi, per tal de estar més a prop del C.

L'inconvenient dels alts llenguatges, com els diagrames o els pseudocodis és que hi ha moltes subrutines que no es mostren, ja que son secundaries però imprescindibles per al correcte funcionament, i la part que no es mostra gens és la configuració inicial, sense la qual res funcionaria. Totes aquestes parts seran explicades amb detall a l'apartat de la programació en C.

Pseudocodi

```

MENTRE ( POLSADOR_START=0)
  ESPERAR 5s
  INICI_PROGRAMA

  INICI_SI (OBJECTE DARRERA)
    GIRAR_DRETA
    ESPERAR (DETECTAR OBJECTE PER DAVANT)0(SI PASSEN 2s )
  FI_SI

  MENTRE (OBJECTE DAVANT DEL ROBOT) FER:
    INICI_SI (OBJECTE DETECTAT NOMES PEL SENSOR DRET)
      GIRAR_ESQUERRA
    ALTRAMENT_SI (OBJECTE DETECTAT NOMES PEL SENSOR ESQUERRA)
      GIRAR_DRETA
    ALTRAMENT_SI (OBJECTE DETECTAT PELS 2 SENSORS)
      ENDAVANT
    FI_SI
  FI_MENTRE

  MENTRE (OBJECTE NO DETECTAT)
    INICI_SI (NO HA PASSAT 1S)
      ENDAVANT
    FI_SI
    INICI_SI (NO HA PASSAT 2,5S)
      GIRAR_DRETA
    FI_SI
    INICI_SI (NO HA PASSAT 3,5S)
      ENDAVANT
    FI_SI
    INICI_SI (NO HA PASSAT 5S)
      GIRAR_ESQUERRA
    FI_SI
  FI_MENTRE
FI_PROGRAMA

```

5.3. Programació en C

Els microcontroladors més convencionals, es poden programar tant en llenguatge ensamblador com en llenguatge C. El mode ensamblador és el nivell de programació més baix que existeix i fa que la programació sigui molt més complicada i complexa. A un nivell una mica més alt està el llenguatge C, el qual és més fàcil tant d'utilitzar com d'entendre. Però el problema que hi ha és que els microcontroladors no entenen el llenguatge C, per tant fa falta un compilador, el qual s'encarrega de transformar el programa en llenguatge C en llenguatge ensamblador.

Tot i l'existència de tot tipus de compiladors, hi ha funcions que no hi ha més remei que escriure-les en ensamblador, com per exemple l'activació de les interrupcions, en el nostre programa s'ha utilitzat aquesta funció, més endavant s'explicarà com fer-ho.

Al mercat hi ha una gran varietat de compiladors, però normalment s'agrupen segons les famílies i marques de microcontroladors. En aquest cas s'ha utilitzat el compilador anomenat *CodeVisionAVR*, el qual permet compilar programes per a tota la família de microcontroladors d'Atmel AVR de 8bits.

Degut a l'ampli informació d'aquest apartat, es divideix l'explicació del programa en tres parts, l'una fa referència a tot el conjunt de configuracions inicials, l'altre a l'explicació de totes les subrutines i finalment el programa principal.

5.3.1. Configuració

L'apartat de configuració és el més important de la programació, ja que no hi pot haver cap error, perquè en cas de que hi sigui, fa que tot vagui malament, ja que el programa sencera depèn de les configuracions inicials. Les configuracions s'efectuen mitjançant l'adjudicació d'uns valor determinats a registres interns de configuració. A continuació es mostra tot el conjunt de configuracions.

- Configuració dels I/O ports

Una característica dels AVR és que al inici del programa s'ha de determinar si un pin serà utilitzat com a sortida o com entrada. La funció que realitza aquesta configuració és la *DDRX.X*. Es pot configurar un port sencera o de pin en pin. Si es vol que el port sigui de sortida se li assigna un 1 i si es d'entrada un 0. Fen un ull a la nostra configuració del hardware, la nostra configuració quedaria de la següent manera:

```
DDRB.0=1;    //activació led 1
DDRB.1=1;    //activació led 2
DDRB.2=1;    //activació led 3
DDRB.3=1;    //activació led 4

DDRC=0XFF;   //Port C com a output(control motors)
DDRD=0x00;   //Port D com a input
DDRA=0X00;   //Port A com a input (lectura sensors)
```

- Interrupcions externes

Una interrupció en un programa, significa que atura el programa principal, realitza el programa de la interrupció i en acabar, retorna a l'estat en que s'havia quedat el programa principal. Les interrupcions externes són utilitzades per a reaccionar ràpidament al detectar la línia de límit del tatami, ja que si se surt del tatami es perd la partida. Per tant cal actuar amb rapidesa. Si no s'utilitzessin les interrupcions i es donés la situació de límit de tatami i el programa estigues dintre d'un bucle, s'haurà d'esperar a acabar el bucle i després reaccionar. Si aquest bucle fos força llarg, segurament alhora de reaccionar el robot ja estaria fora del tatami.

Com que es tracte d'una interrupció, el primer que s'ha de fer és habilitar-les, i no hi ha altre manera de fer-ho que utilitzar llenguatge ensamblador, amb la següent instrucció:

```
//interrupt enable
#asm ("sei");
```

Però perquè les interrupcions no interrompin la configuració inicial, aquesta instrucció es posarà al final de totes les configuracions.

Ara cal seleccionar el tipus d'interrupcions que volem utilitzar. En el nostre programa l'única interrupció que s'utilitzarà serà la de les interrupcions externes, les quals s'habiliten des de el registre GICR.

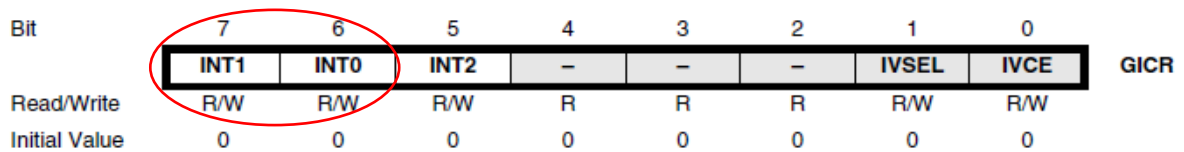


Figura 36. Registre GICR

Les interrupcions externes que utilitzarem seran les INT1 i INT0. Un cop habilitades cal configurar-les quan s'activaran com a resposta de la senyal rebuda. Com que per defecte el robot sempre estarà dins la superfície negra, el sensor proporciona un 1 lògic, i al detectar la línia blanca del límit del tatami el sensor passa a 0 lògic, és configura la interrupció de manera que reaccioni a flancs de baixada, es a dir, quan es passi de 1 a 0.

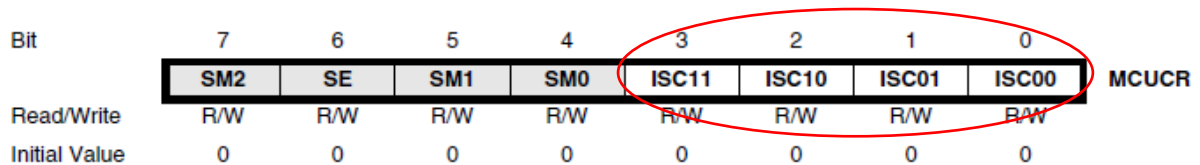


Figura 37. Registre MCUCR

Aquesta configuració es fa mitjançant el registre MCUCR, des de el qual es configuren ambdues interrupcions. Els bits ISC11 i ISC10 serveixen per a configurar la INT1 i els ISC01 i ISC00 per la INT0. A continuació es mostra les opcions de configuració de INT1. La INT0 és totalment igual.

Taula 6. Configuració INT1

ISC11	ISC10	Descripció
0	0	Interrupció per nivell baix de INT1
0	1	Interrupció per a qualsevol canvi lògic
1	0	Interrupció per flanc de baixada
1	1	Interrupció per flanc de pujada

```
//activació de la INT0 i INT1 per flanc de baixada
MCUCR=MCUCR|0b00001010;
//ativació INT1&INT0 exterior
GICR=GICR|0b11000000;
```

- Convertidor Analògic-digital

El convertidor que te aquest microcontrolador és de 10bits i és del tipus d'aproximacions successives, el qual te múltiples usos.

El convertidor analògic-digital és el utilitzar per a llegir els sensors de distància, els quals estan connectats als pins 4,5 i 6 del port A, els quals fan referència a les entrades del convertidor ADC4, ADC5 i ADC6 respectivament. Tot i que hi ha 8 pins que accedeixen al convertidor, a l'interior del integrat només hi ha un convertidor, el que hi ha abans és un multiplexor el qual selecciona l'entrada desitjada. Per tant s'haurà de configurar aquest multiplexor cada vegada que es vulgui llegir cada un dels 3 sensors.

El primer pas és configurar el convertidor per tal de poder llegir correctament la senyal i que no surti de rang. També cal escollir la freqüència de mostreig. Es fa mitjançant el registre ADCSRA. En aquest es configuren les següents funcions.

Bit	7	6	5	4	3	2	1	0	
	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0	ADCSRA
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Figura 38. Registre ADCSRA

- S'habilita la conversió analògic-digital (bit 7=1)
- Conversió de manera manual (bit 6=1)
- Inhabilitació de generar interrupció (bit 3=0)
- Freqüència de mostreig a 115,2kHz (prescala de 32) (bit 2-1-0 =101)

D'aquesta manera cada vegada que es vulgui utilitzar el convertidor s'haurà d'activar manualment posant el bit 6 (ADSC) =1 i esperar a que aquest mateix bit es posi a 0 automàticament, ja que significarà que la conversió ja s'ha realitzat i ja es pot llegir el valor obtingut.

L'altre registre que cal configurar és l'ADMUX, el qual selecciona el multiplexor. Per tant, aquest registre s'haurà de modificar constantment al llarg del programa. Però apart de seleccionar el multiplexor, també configura la tensió de referència que s'utilitzarà, i si el convertidor s'utilitza de 8 o 10 bits.

Bit	7	6	5	4	3	2	1	0	
	REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0	ADMUX
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Figura 39. Registre ADMUX

Les configuracions que es mantindran en els 3 casos seran:

- Tensió de referència de 2,56 V (bit 7-6=11)
- Convertidor de 8bits (bit 5 = ADLAR=0)

Amb la tensió de referència de 2,56 V és més que suficient, ja que el programa actuarà quan detecti algun objecte a 10 cm, i a aquesta distancia els sensors proporcionen 1,2V, per tant és més que suficient. La tensió de referència al ser interna serà molt estable.

Es diu que s'ha configurat el convertidor a 8 bits de resolució perquè només es llegiran els 8 MSB (bits més significants), ja que els altres 2 restants estaran en un altre registre. Posant el bit ADLAR=0, fa que les 8 MSB se situïn al registre ADCH i els 2 restants al ADCL. Un cop explicat aquest punt, veiem que gracies al compilador, ofereix la opció de llegir ambdós registres de cop, i es pot fer mitjançant la lectura del registre de ADCW. Per tant, s'acaba utilitzant els 10 bits dels convertidor.

Ara alhora de llegir cada un dels sensors haurem de configurar els bits 4..0 del registre. Els quals seran de la següent manera.

Taula 7. Configuració multiplexor ADC

Bit 4..0	Descripció
00100	ADC4
00101	ADC5
00110	ADC6

Un cop ja esta tot configurat toca modificar el valor obtingut pel microcontrolador, ja que el valor serà proporcional als valors de 10 bits, és a dir, de 0 a 1024. L'operació que realitza el microcontrolador amb els valors llegits és segons la següent formula:

$$ADC = \frac{V_{in} \times 1024}{V_{ref}} = V_{in} \frac{1024}{2,56} = \frac{V_{in}}{400}$$

Per tant l'operació a realitzar per a poder treballar amb els valors que ens ofereix el sensor és:

$$V_{in} = \frac{ADC}{400}$$

Però, al fer aquesta operació tocaria treballar amb decimals, i per evitar problemes de variables amb el llenguatge C, es multiplicarà per 100 per tal d'eliminar les comes, per tant l'equació final quedarà:

$$V_{in} = \frac{ADC}{4}$$

```
// SETTINGS A/D CONVERTER
```

```
ADCSRA = 0b11000101;
```

```
ADMUX = 0b11000101; //2,56V REF & RIGHT ADJUST & ADC5 SELECTED
```

- Temporitzador

Per a controlar tots els temps fa falta un temporitzador. Aquest microcontrolador té varis comptadors/*timers* interns, dels quals s'ha utilitzat el timer 1, que és de 16 bits. S'ha escollit aquest timer perquè amb la correcta configuració, es poden comptar segons sense que faci falta sobrepassar (overflow) el comptador, i sense ser necessari generar interrupcions.

Com que l'ús del comptador es en mode normal i no es vol generar interrupcions, en aquest cas només cal configurar la reducció de freqüència de comptabilitzar. Es dividirà per la més gran que és 1024, de tal manera que la freqüència de comptabilitzar serà de:

$$f_{timer} = \frac{f_{clock}}{prescala} = \frac{3,6864Mhz}{1024} = 3600Hz$$

D'aquesta manera es podrà comptar 1 segon cada vegada que el comptador arribi a 3600 i els múltiples d'aquest. I hi ha un ampli marge perquè com que el comptador és de 16 bits, pot arribar a comptar fins a 65536. La prescala es configura des de el registre TCCR1B, mitjançant els bits CS12, CS11 i CS10.

Bit	7	6	5	4	3	2	1	0	
	ICNC1	ICES1	-	WGM13	WGM12	CS12	CS11	CS10	TCCR1B
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Figura 40. Registre TCCR1B

Un cop està configurat només cal donar-li un valor en concret al timer i ja començarà a comptar. És important reiniciar-lo cada vegada que se'l vulgui consultar, ja que no para de funcionar. L'inconvenient és que al ser de 16 bits, necessita 2 registres per a ser configurat, ja que utilitzem un microcontrolador de 8 bits, però gracies al compilador, la llibreria <mega16.h> permet controlar ambos registres a la vegada mitjançant la funció TCNT1. Per tant, es pot treballar com si hi hagués un registre de 16 bits.

```
// CONFIG TIMER 16 bits prescaler = ckl/1024, ffreq=36kHz.
TCCR1B=TCCR1B|0b00000101;
```

5.3.2. Subrutines

- Motors

Per tal de poder treballar amb facilitat, es creen un conjunt de subrutines per tal d'agilitzar alhora d'utilitzar el control dels motors en el programa principal. Abans cal mostrar les definicions que s'han fet al pins que pertanyen al control del motor.

```
#define A2 PORTC.0
#define B2 PORTC.1
```

```
#define EN2 PORTC.2
#define EN1 PORTC.3
#define A1 PORTC.4
#define B1 PORTC.5
```

Els noms assignats són els corresponents als pins de control del L293D, per tal de ser més fàcil la configuració. El primer pas de la configuració, considerarà en generar unes subrutines les quals controlin els 2 motors, tot seguint les taules de control del L293D. A continuació es mostrarà la configuració del motor 1, la del motor 2 es totalment igual però canviant les variables.

```
void MOT1_ENDAVANT()          void MOT1_ENRERE()
{
    EN1=1;                    {
    A1=1;                      EN1=1;
    B1=0;                      A1=0;
                                B1=1;
}                                }

void MOT1_STOP()
{
    EN1=1;
    A1=1;
    B1=1;
}
```

Un cop es tenen els moviments individuals de cada motor per separat, es crearan els moviments reals del robot, que són endavant, enrere, dreta, esquerra i stop.

```
void endavant()              void endarrere()
{
    MOT1_ENDAVANT();         {
    MOT2_ENRERE();           MOT2_ENDAVANT();
}                             MOT1_ENRERE();
}

void dreta()                 void esquerra()
{
    MOT1_ENDAVANT();         {
    MOT2_ENDAVANT();         MOT1_ENRERE();
}                             MOT2_ENRERE();
}

void stop()
{
    MOT1_STOP();
    MOT2_STOP();
}
```

Com dona a entendre la configuració, el motor 1 és el que està situat a la dreta del robot i el motor 2 a l'esquerra.

- Sensors de distància

Per tal d'agilitzar la lectura de l'entorn, es creen 3 variables que consisteixen en la lectura dels sensors de distància, d'aquesta manera, cada vegada que es consultin aquestes variables es procedirà a la lectura dels sensors, on inclou l'utilització del convertidor analògic-digital.

Però alhora de realitzar unes quantes proves es troba que els sensors o les lectures no sempre són del tot correctes, per tant es procedirà a la realització d'un filtre digital per tal d'eliminar les lectures errònies. El filtre utilitzat és el de la mitjana, el qual consisteix en realitzar un grup de mesures i agafar com a valor vàlid la mitjana d'aquests valors.

Per a realitzar aquesta es farà la lectura de 100 valors, a continuació es mostra com s'ha fet la subrutina del sensor del darrere.

```
unsigned int distancel()      //sensor darrere
{
    unsigned long i, media=0;
    ADMUX = 0b11000100;
    for (i=0;i<100;i++)
    {
        ADSC = 1;
        // esperar fins que es faci la conversió
        while (ADSC) {}
        media=media + ADCW;
    }
    media=media/400;
    return (media);
}
```

Cal comentar que la mitjana es divideix per 400 en comptes de 100 perquè com s'ha comentat anteriorment, per obtenir el valor de tensió que oferia el sensor era necessari dividir per 4 el valor obtingut, per tant el valor de 400 és el resultat de dividir per 100 i per 4. D'aquesta manera des de la funció ja obtenim el valor real.

- Sensor de línia CNY70

La lectura d'aquest sensor es fa mitjançant interrupció, mes que lectura la paraula seria detecció, ja que quan s'activa la interrupció és que s'ha detectat la línia de límit del tatami. De manera que la reacció a aquesta situació es farà dins de la subrutina de la interrupció.

Cal recordar que per realitzar les interrupcions cal saber el numero d'interrupció al que pertanyen, que en aquest cas la INTO i la INT1 el nombre de vector 2 i 3 respectivament. Això fa que siguin les més importants per el microcontrolador, ja que en cas que s'executessin dues interrupcions alhora, primer atendria a aquestes, però com que són les úniques que s'utilitzen, no existeix aquests problema.

```

interrupt [2] void CNY0 ()
{
  endarrera();
  delay_ms(800);
  dreta();
  delay_ms(700);
}

interrupt [3] void CNY1 ()
{
  endevant();
  PORTB=0b00000100;
  delay_ms(200);
}

```

Es necessari comentar la funció `delay.h`, on aquí s'utilitza la variant "ms", aquesta funció el que fa es generar un retard, i la utilització es tan simple com que a dins del parèntesi s'introdueixen els milisegons que es volen retardar.

5.3.3. Programa principal

El programa principal, tal i com diu el seu nom, és el més important, ja que controla i utilitza totes les funcions dissenyades anteriorment. El programa principal es divideix en 3 parts diferenciades, l'una és l'inici del programa, que és la més simple, i després es troben les dues accions del combat, l'una és l'atac i l'altre la cerca del robot contrari.

- Inici de combat

Tal i com marca el reglament de les competicions de minisumo, els robots s'han de deixar sols dins del tatami i al prémer el boto d'inici s'han de esperar 5 segons abans no comencin a lluitar.

Per a realitzar aquesta funció es farà que al donar tensió al robot es realitzaran totes les configuracions i el robot esperarà a que el polsador sigui polsat per tal d'arrencar el programa principal.

```

//encendre LED 1
PORTB.0=1;
//esperar fins a polsar el polsador d'inici
while (PINA.0==0) {};
delay_ms(5000);

//interrupt enable
#asm ("sei");
TCNT1=0;
while (1)
{...programa principal...}

```

Mitjançant el bucle `while` d'aquest apartat, el programa estarà totalment aturat fins que no es premi el polsador d'inici. Un cop polsat, s'iniciarà el retard de 5 segons, tal i com marca la normativa. A continuació s'activaran les interrupcions, és molt important que no s'activin abans del polsador d'inici, ja que el robot podria arribar a moure's abans dels 5 segons, i faria que fos desqualificat abans de començar. Tot seguit, ja començaria el programa principal, el qual esta format per el bucle infinit `while (1) {}`.

- Buscar objectiu

Aquesta part del programa s'executa quan els sensors no detecten cap objecte, en aquest cas entre en el mode cerca d'objectiu, on es realitza un moviment aleatori per el tatami per a buscar el robot contrari.

Per tal que el robot comenci a buscar s'haurà de donar el cas en que els 3 sensor a la vegada no detectin res a menys de 30 cm, en donar-se aquesta situació, es procedirà a la cerca del robot contrari i el moviment que realitzarà és el descrit al diagrama de flux i al pseudocodi.

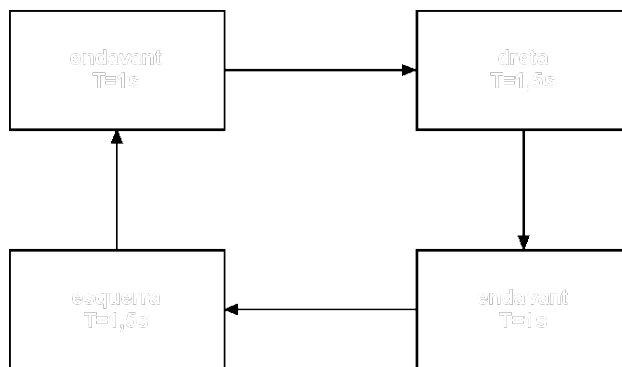


Figura 41. Moviment per buscar contrincant

Aquest moviment aleatori s'ha dissenyat de tal manera que en la realització d'aquest, no hi ha cap retard existent, de tal manera que si durant la realització d'aquest moviment de cop i volta detecta algun objecte, el robot procedirà ràpidament a atacar-lo per a treure'l del tatami.

```

while ((distance2() < 40) & (distance3() < 40) & (distance1() < 40))
{
    if (TCNT1 <= 3600) // 1seg
    {
        endavant();
    }
    else if (( TCNT1 > 3600)&(TCNT1 < (2.5*3600)))
    {
        dreta();
    }
    else if (( TCNT1 > (2.5*3600)) & ( TCNT1 < (3.5*3600)))
    {
        endavant();
    }
    if (TCNT1 > (3.5*3600))
    {
        esquerra();
    }
    if(TCNT1 > (5*3600))
    {
        TCNT1=0;
    }
}
  
```

- Atacar objectiu

Al detectar un objectiu s'entra en el estat d'atac, però com que els objectius també poden ser detectats per el sensor del darrere, en aquest cas el primer que caldrà fer és girar el robot per encarar-se al contrincant per poder atacar-lo correctament.

Per tant, per començar s'explicarà la rutina de lectura del sensor de darrera i la resposta corresponent a aquest. El sensor del darrere està configurat per a detectar un objecte a uns 25cm. Un cop detectat, es programa que el robot giri cap a la dreta fins que l'objecte sigui detectat per els sensors de distancia frontals. Aquest programa podria ocasionar un bucle infinit i fer que el robot no pares de donar voltes, ja que si al girar l'altre robot marxa, el nostre robot no pararia de girar cap a la dreta, ja que mai no el trobaria. És per això que al començar a girar cap a la dreta s'activa un comptador, de tal manera que si al passats 2 segons el robot no ha detectat res amb els sensors frontals, el programa ometrà la cerca d'aquest i tornarà al estat inicial.

```
if (distance1() > 50)
{
  dreta();
  TCNT1=0;
  while ((distance2() < 40)|(distance3() < 40)|(TCNT1 > (2*3600))) {}
}
```

Un cop l'objectiu està encarat, el robot ja està preparat per atacar-lo, ja que davant disposa de 2 sensors, un a cada banda, gracies als quals el robot pot saber per a quin costat el robot contrari s'està escapant i pot perseguir-lo.

Per entrar en aquest estat solament caldrà que algun dels 2 sensors frontals detecti algun objecte a menys d'uns 30cm. Un cop detectat, es crea una variable la qual serà molt útil alhora de generar la rutina de seguiment del robot contrari. Aquesta variable consisteix en la diferencia entre els 2 sensors frontals, és a dir, la diferencia de tensió que generen cada un dels sensors, de manera que depenent del valor d'aquesta variable es pot saber la posició del robot contrari. Els rangs establerts són de: més petit que -40, entre -40 i 40 i més gran de 40.

```
diferencia= distance2()-distance3();
```

Taula 8. Significat variable diferencia

Valor de la variable: DIFERENCIA	Posició robot contrari
Diferencia < -40	S'escapa per l'esquerra
$-40 \leq \text{Diferencia} \leq 40$	Just davant
Diferencia > 40	S'escapa per la dreta

```
while ( (distance2() > 40) | (distance3() > 40) )
{
    PORTB=0b00000011;

    diferencia= distance2()-distance3();
    if (diferencia >= 40)
    {
        dreta();
    }
    else if (diferencia <= -40)
    {
        esquerra();
    }
    else if ((diferencia < 40)& (diferencia > -40))
    {
        endavant();
        delay_ms(100);
    }
}
```

Mitjançant aquest programa el robot contrari no es podrà escapar de davant del nostre, i en cas de que ho fes, el robot entraria automàticament en el mode de cerca.

5.4. CodeVisionAVR

El programa utilitzat per a programar el microcontrolador ha estat el *CodeVisionAVR*, el qual és un programa de pagament que ofereix una versió per a estudiants la qual pot realitzar les mateixes funcions que l'original amb un límit alhora de compilar, però aquest límit és molt més gran que el que es necessita per a aquest projecte, per tant és útil.

Aquest programa serveix per a programar tota la família de microcontroladors Atmel AVR de 8bits, i reconeix perfectament el kit STK500 utilitzat per a programar.



HP InfoTech
C Compiler, Integrated Development Environment,
Automatic Program Generator and In-System Programmer
for the Atmel AVR Family of Microcontrollers

Version 2.04.7a Evaluation
© Copyright 1998-2010 Pavel Haiduc, HP InfoTech s.r.l.
<http://www.hpinfotech.com>

Freeware, for evaluation and non-commercial use only

Figura 42. Capçalera del programa

Per a programar amb aquest compilador s'han de seguir una sèrie de passos, i s'explicaran els més importants.

1. Comunicació amb els microcontrolador

Òbviament la comunicació entre ordinador i microcontrolador és el primer pas. En aquí es troba el reconeixement del kit STK500, pera això s'ha a la configuració del programador, i seleccionar el següent:

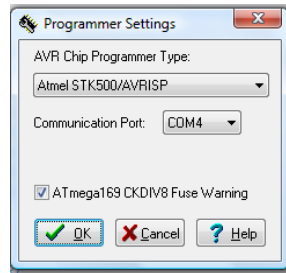


Figura 43. Comunicació

2. Configuració del projecte

Un cop configurada la comunicació s'ha de configurar el projecte per tal que s'adeqüi al microcontrolador utilitzat, que en aquest cas és el Atmega16. En la configuració s'ha de seleccionar el microcontrolador, la freqüència del rellotge, que és 3,6864MHz, i un conjunt d'opcions que és poden observar a la següent imatge.

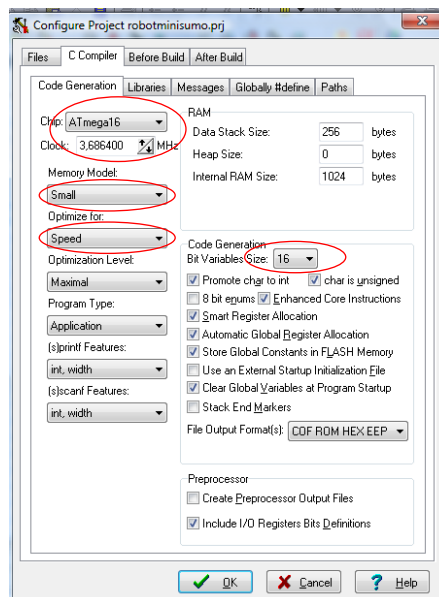


Figura 44. Configuració del projecte

Les parts encerclades amb el cercle vermell són les més importants a canviar, ja que les altres ja venen per defecte. Tot i que el microcontrolador és de 8 bits, el compilador pot treballar amb WORDS (variables de 16 bits), és per això que es marca la mida de les variables de 16 bits, ja que si no es fes, no es podrien utilitzar funcions explicades anteriorment, com ara la lectura directe del ADC o la configuració del Timer 1, el qual és de 16 bits.

3. Selecció del document C

Aquesta part, és el típic error, ja que tot i que estiguis treballant sobre un document i li manis que programi el microcontrolador, si abans no s'ha configurat quin document C s'ha de introduir, no ho farà correctament.

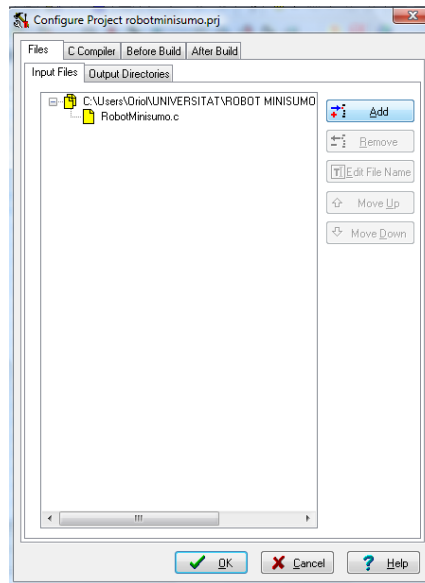


Figura 45. Selecció de document C

4. Confirmació de la programació

Per últim, en l'apartat de configuració, s'ha de seleccionar que apart de la generació del codi assemblador, que programi el chip de manera automàtica, i això es fa simplement de la següent manera.

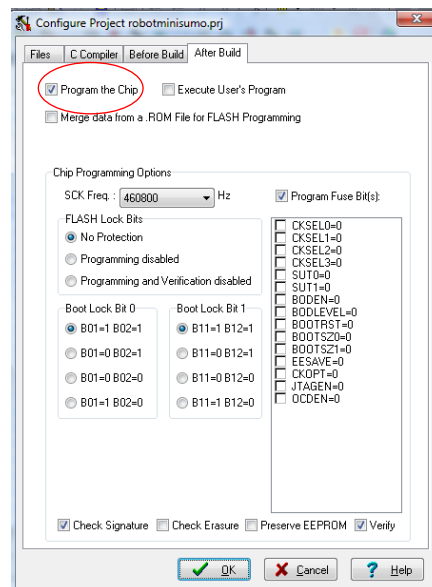


Figura 46. Confirmació

És molt important marcar el camp ressaltat amb el cercle vermell.

5. Compilar i Programar

Aquest programa permet compilar i programar el xip en un sol clic, de manera que compilarà tot el programa i si no detecta cap error ens demanarà la confirmació per a procedir a la programació del microcontrolador.

Per a realitzar aquesta funció només cal fer clic a:

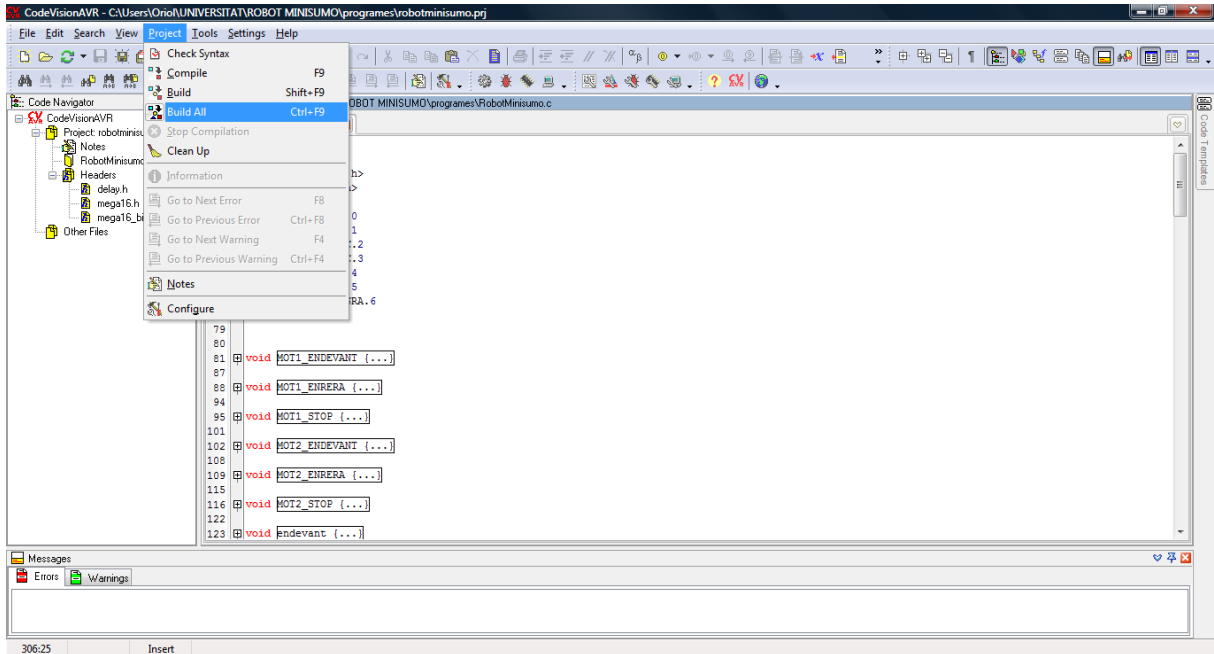


Figura 47. Compilació

Tot seguit, si no hi ha cap error ens apareixerà la següent finestra, la qual ens demana que confirmem que volem programar, simplement polsem sobre programar i ja s'haurà programat.

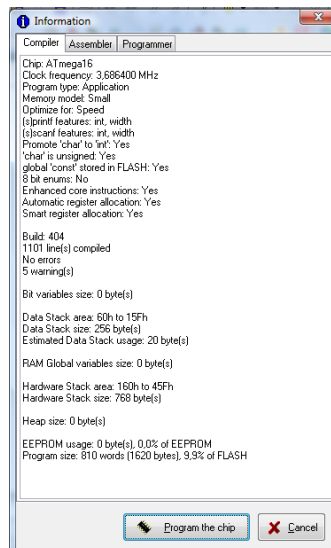


Figura 48. Programació

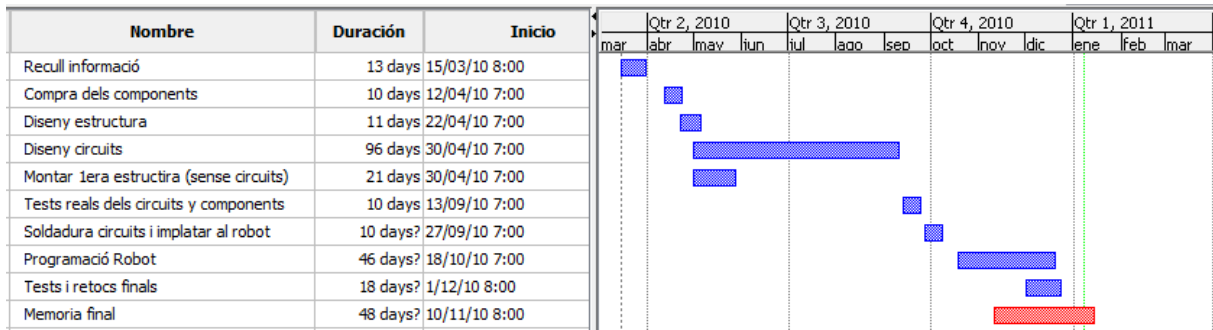
En aquesta finestra ens informa de la mida del nostre programa i si hi haguessin errors ens indicaria quins. També ens fa un petit resum de la configuració assignada.

CAPÍTOL 6: DIAGRAMA DE GANTT

El diagrama de Gantt es va dissenyar al inici de la realització d'aquest projecte, i ha estat una guia a seguir, i com a resultat final a estat molt útil, ja que s'ha seguit força i ha permès organitzar-me i saber el que s'ha de fer en cada moment.

Partida	Dies dedicació	inici	Fi
Recull informació	13	15/03/10	31/03/10
Compra dels components	10	12/04/10	23/04/10
Disseny estructura	11	22/04/10	06/05/10
Disseny circuits	11	22/04/10	06/05/10
Muntar 1era estructura(sense circuits)	21	30/04/10	28/05/10
Tests reals dels circuits y components	10	13/09/10	24/09/10
Soldadura circuits i implantar al robot	10	27/09/10	08/10/10
Programació Robot	46	18/10/10	20/12/10
Tests i retocs finals	18	01/12/10	24/12/10
Memòria final	48	10/11/10	14/01/11

El programa utilitzat per a generar el diagrama de Gantt ha estat l'anomenat Open Project, el qual és un programa gratuït. A continuació hi ha una captura de pantalla del diagrama.



CAPÍTOL 7: CONCLUSIONS

Un cop acabat el projecte i veient el resultat obtingut es pot afirmar que s'han aconseguit els objectius plantejats al inici del projecte.

- S'han optimitzat els circuits electrònics, ja que s'ha aconseguit construir una única placa electrònica la qual inclou tota l'electrònica. Aquest ha estat un gran èxit, ja que ens ha permès tenir molt més marge d'espai per a resoldre els altres problemes, sobretot mecànics.
- Els sensors escollits funcionen força be, però cal recordar que es van trobar problemes en els sensors de distància, però s'ha solucionat amb èxit gràcies a un filtre digital per software.
- Pel que fa als motors, és la part que es podria millorar, ja que s'han utilitzat uns motors una mica petits per i no gaire potents. Però s'han escollit aquests degut a la fàcil accessibilitat i al preu, ja que són molt econòmics.
- La part mecànica tot i semblar força dèbil a primera vista, ja que s'ha realitzat amb LEGO, s'ha reforçat l'estructura amb cinta i loctite per tal de fer-la més estable i fixa. I per últim, s'ha creat tota una carcassa amb poliestirè, la qual es fàcil de treure, per tal de tenir protegit tots els costats del robot, i impedir que sigui aixecat del terra per al robot contrari. Cal remarcar en aquest punt, que la part mecànica s'ha solucionat mitjançant material que es tenia per casa, per tant és un punt a tenir en compte, sobretot en l'estalvi econòmic.
- Finalment, la programació ha estat tot un èxit, ja que s'ha elaborat un programa prou fort, el qual és molt ràpid de reaccionar, a causa de la inexistència de retards i de bucles que puguin generar un bucle infinit i que facin penjar el robot.

Com a conclusió final, aquest robot està preparat per a competir, només caldria substituir els motors per uns de més potents.

CAPÍTOL 8: BIBLIOGRAFIA

La bibliografia consultada és bàsicament pàgines web.

- <http://robocat.weebly.com/minisumo.html>
- <http://www.superrobotica.com/>
- <http://www.ridgesoft.com/>
- <http://jet-upc.blogspot.com/>
- <http://upcommons.upc.edu/pfc/>