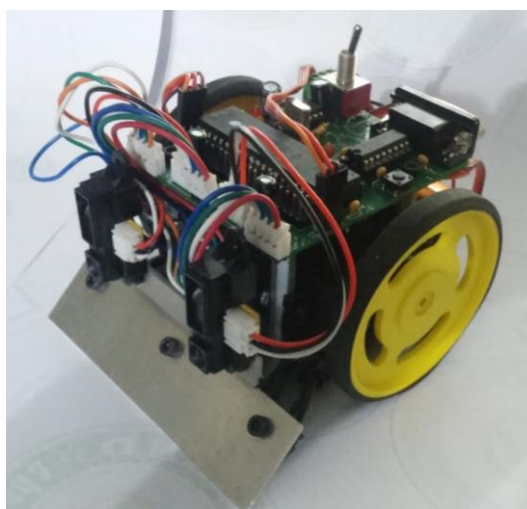


# **C-BOT: ROBOT LLUITADOR DE SUMO**



**Autor: Ismael Douha Prieto**

**21 de Gener de 2016**

**Àmbit Tecnològic**

**Tutora: Encarna Garcia**



*Agrair a la Universitat Politècnica de Catalunya la possibilitat de utilitzar les seves instal·lacions i a l'AESS, més concretament a Roser, Martí i Josep pels cursos de C i de robòtica.*

*També agrair a la meva tutora Encarna Garcia, el seu esforç i dedicació per fer possible aquest treball.*



## Índex

---

|  |    |
|--|----|
| 0.- Introducció.....   | 2  |
| 1.- Introducció a la robòtica .....                          | 5  |
| 1.1.- Tipus de robots .....                                  | 6  |
| 1.2.- Sistemes de control automàtic .....                    | 8  |
| 1.2.1.- Automatització .....                                 | 8  |
| 1.2.2.- Sistemes de control .....                            | 9  |
| 1.3.- Concursos de Robòtica.....                             | 10 |
| 2.- Llenguatges de programació.....                          | 11 |
| 2.1.- Història de la programació .....                       | 11 |
| 2.2.-El llenguatge de programació C.....                     | 12 |
| 2.2.1.- Variables i Constants.....                           | 12 |
| 2.2.2.- Operadors.....                                       | 14 |
| 2.2.3.- Sentències de Control de Flux.....                   | 15 |
| 2.2.4.- Funcions .....                                       | 17 |
| 2.3.- Llibreries .....                                       | 18 |
| 2.4.- Estructura d'un programa en C.....                     | 19 |
| 3.- Elements de disseny del robot .....                      | 20 |
| 3.1.- Especificacions de disseny .....                       | 20 |
| 3.2. Components del robot.....                               | 21 |
| 3.2.1.- Controlador.....                                     | 21 |
| 3.2.2.- Actuadors .....                                      | 28 |
| 3.2.3.- Sensors .....  | 29 |
| 3.2.4.- Elements d'accionament manual.....                   | 30 |
| 4.- Disseny, Implementació i Programació del meu robot ..... | 31 |
| 4.1.- Disseny i implementació del prototipus.....            | 31 |
| 4.1.1.- Disseny del prototipus .....                         | 31 |
| 4.1.2.- Implementació del prototipus.....                    | 32 |
| 4.2.- Programació del prototipus .....                       | 33 |
| Conclusions.....   | 36 |
| Propostes de millora.....                                    | 38 |
| Llista de referències .....                                  | 39 |

|   |    |
|---|----|
| Bibliografia .....  | 39 |
| Webgrafia.....  | 39 |
| Annex 1: Programa del robot.....                          | 41 |
| Annex 2: Pressupost .....                                 | 47 |
| 1.- Costos d'enginyeria.....                              | 47 |
| 1.1.- Càlcul del preu hora .....                          | 47 |
| 1.1.1.- Costos directes .....                             | 47 |
| 1.1.2.- Costos indirectes .....                           | 47 |
| 1.1.3.- Preu total .....                                  | 48 |
| 1.2.- Costos derivats del disseny .....                   | 48 |
| 1.3.- Costos derivats de l'elaboració de la memòria ..... | 48 |
| 1.4.- Cost total enginyeria .....                         | 49 |
| 2.- Costos derivats del muntatge.....                     | 50 |
| 2.1.- Càlcul del preu hora.....                           | 50 |
| 2.1.1.- Costos directes .....                             | 50 |
| 2.1.2.- Costos indirectes .....                           | 50 |
| 2.1.3.- Costos totals derivats del muntatge.....          | 50 |
| 2.2.- Cost total del muntatge del robot .....             | 50 |



## **0.- Introducció**

En els nostres dies, la robòtica ha anat guanyant terreny, de manera que els petits problemes quotidians, i d'altres no tant petits, els resol amb facilitat. Un exemple clar és la cirurgia robòtica, capaç de realitzar operacions amb major rendiment i fiabilitat que un humà.

He decidit fer el meu Treball de Recerca sobre aquest tema perquè des de ben petit m'apassiona tot allò relacionat amb robots i ordinadors i era el moment perfecte per iniciar-me en aquest món. La meva principal motivació és el fet que, en un futur no molt llunyà m'agradaria estudiar un Grau en Enginyeria relacionat amb la programació tant de robots com d'altres dispositius.

Aquest Treball de Recerca consisteix en el disseny, la implementació i la programació d'un robot amb la capacitat de poder participar en combats de sumo. Tinc la intenció de presentar el meu robot al concurs AESSbot, organitzat per l'AESS a la categoria de *sumo*.

La realització d'aquest treball m'ha obligat a enfrontar-me a més d'un repte. En primer lloc, a iniciar-me a la programació en C, necessària per a la programació del meu prototipus. Fins al moment només havia programat una mica en BASIC a l'optativa de Tecnologia de 4t d'ESO i, iniciar-me en un llenguatge de programació com el C, suposava un gran repte. D'altra banda, he hagut de consolidar i aprofundir els meus coneixements d'electrònica, ja que el meu robot és un sistema electrònic força complex.

Per sort, he tingut la gran oportunitat de participar en el programa de cotutories de robòtica TelecomBCN-UPC organitzat per l'Escola Tècnica Superior de Telecomunicació de Barcelona (ETSEETB). En concret, aquestes cotutories les realitzen a l'AESS (Associació universitària de la ETSEETB UPC que té com objectiu promoure el món de la robòtica i les noves tecnologies a l'àmbit universitari). Durant la meva estància, m'he pogut iniciar en la programació en llenguatge C i aplicar aquests coneixements a la implementació del meu robot.

Respecte al contingut del treball, primer faré una breu introducció a la robòtica, passant pels diferents tipus d'automatismes i com s'estructuren internament depenent de les necessitats del creador. També faré cinc cèntims sobre els



diferents concursos sobre robòtica que podem trobar al nostre país i les categories de les quals disposen.

Seguidament, explicaré de forma resumida els conceptes més rellevants sobre la programació en C, primer parlant sobre l'evolució dels llenguatges de programació fins a arribar l' utilitzat per a la programació del robot. Dins d'aquest, comentaré les diferents parts que componen un programa i tot lo relacionat amb la seva estructura.

A continuació, explicaré breument com he realitzat el disseny del meu prototipus. Dins aquest apartat, trobareu les especificacions bàsiques que ha de tenir el robot. També explicaré la funció dels principals components electrònics que fan funcionar el meu robot lluitador de sumo.

Finalment, parlaré sobre el disseny triat i el programa utilitzat per el seu funcionament.



## **1.- Introducció a la robòtica**

Des de l'inici l'ésser humà ha estat intentant trobar procediments, eines o màquines per reduir les dificultats de les feines que havia de realitzar per sobreviure. L'objectiu era desenvolupar tècniques que ens facilitessin el dia a dia i augmentessin la productivitat. Un exemple força revolucionari va ser l'invent de la roda. Aquesta màquina simple va reduir la dificultat de transportar mercaderies i objectes pesats.

Amb el pas del temps, l'ésser humà no només es limita a intentar desenvolupar noves tècniques, sinó que comença a produir maquinària que substitueix a jornalers amb una productivitat més gran que amb les millors tècniques, ja que una màquina no es cansa i sempre produeix al 100% de les seves possibilitats.

La robòtica és la branca de l'enginyeria mecànica, elèctrica, electrònica que s'ocupa del disseny, construcció, operació, disposició estructural, manufactura i aplicació dels robots. Un robot<sup>1</sup> és una màquina automàtica capaç de manipular objectes, executar operacions i moviments diversos segons un programa que pot ser modificable o adaptable, i que pot anar equipat amb sensors per tal de detectar els senyals d'entrada i les condicions ambientals.

La robòtica es basa en la combinació de diferents disciplines: electricitat, electrònica, pneumàtica i mecànica, entre d'altres.

El control automàtic i la robòtica ha permès automatitzar la fabricació de molts productes, evitant a les persones tasques repetitives i perilloses, reduint d'aquesta manera els costos de producció.

La robòtica, a l'actualitat, és una disciplina estratègica als països desenvolupats. L'automàtica i la robòtica ja està integrada a la indústria i el repte és la seva incorporació a altres sectors, com ara la domòtica, medicina, turisme, automoció, etc.

---

<sup>1</sup> Prové del txec de la paraula *robot* que significa "treball forçat"





## 1.1.- Tipus de robots

En l'actualitat, existeixen una infinitat de robots i automatismes, i tots amb diferent forma, estructura, característiques, aplicacions, funcions, etc. Els robots es poden classificar en mòbils, hominoides, industrials i educatius.

- **Robots mòbils:** Tenen la capacitat, com el seu nom indica, de moure's per l'entorn i no quedar-se posicionats en un lloc com els industrials. Avui dia, són un focus important de recerca, fins al punt que les grans universitats tenen un laboratori dedicat a la investigació i desenvolupament d'aquest tipus de robots. Alguns exemples de robots mòbils són els robots aspiradors i els cotxes de ràdio control.



*Figura 1. 2.- Roomba, exemple d'un robot mòbil.*

- **Robots hominoides:** Aquest tipus de robots també són coneguts per andròides. Es caracteritzen per tenir una similitud morfològica amb els éssers humans i tenir la capacitat de realitzar accions humanes, com ara caminar, parlar, moure objectes, etc. Encara que en els últims anys s'ha aconseguit que realitzin accions més complexes com ballar, l'ambició dels desenvolupadors continua i aspiren a crear robots hominoides que puguin expressar els seus sentiments i a prendre decisions seguint patrons humans.



*Figura 1. 3.- Asimo, el robot hominoide creat per l'empresa Honda.*



- **Robots industrials:** És la categoria més coneguda. Es tracta, en gran part, de braços mecànics amb una estètica bastant semblant a la d'un braç humà. Estan formats per tres parts: articulacions, actuadors i sensors. Les aplicacions més habituals solen ser en la fabricació de cotxes, telèfons, aparells electrònics, entre d'altres, ja que presenten un gran avantatge respecte al treball humà: tenen la capacitat de realitzar feines permanents sense baixar el rendiment. A més a més, poden realitzar les feines perilloses per als humans . Gràcies a aquest robots és possible la societat de consum, ja que augmenten la productivitat de forma considerable.



*Figura 1. 4 .- Exemple d'un robot industrial.*

- **Robots educatius:** Acostumen a ser robots amb una estructura simple, de fàcil muntatge i programació. A l'actualitat s'incorpora la robòtica a les aules per tal d'introduir els alumnes al món de la tecnologia i la programació.



*Figura 1. 5.- Exemple d'un robot educatiu.*



## **1.2.- Sistemes de control automàtic**

Un sistema de control automàtic és el conjunt d'elements tècnics que tenen com a objectiu aconseguir que una màquina o un procés realitzi les seves funcions amb una intervenció humana mínima. Aquesta intervenció es limita a donar les ordres per a que comenci a funcionar o per solucionar possibles errors.

### **1.2.1.- Automatització**

En aquest últims temps, les investigacions entorn a la robòtica estan augmentant considerablement. Per consegüent, cada cop es treballa més en una característica fonamental en la robòtica contemporània, l'automatització.

L'automatització es podria definir com la capacitat que té un robot de operar amb la mínima intervenció humana, és a dir, sense que ningú li ordeni realitzar les accions que ha de fer. Un dels grans avantatges que ens atorga, apart de requerir menys mà d'obra, es la velocitat de producció, ja que aplicat en el sector de la indústria, ha premés fenòmens com la Revolució Industrial. D'altra banda, ens evita de fer feines pesants o perilloses, com pot ser l'extracció de minerals, manipulació de productes tòxics, etc.

A part d'aplicar-se a la indústria, té moltes més aplicacions en el món actual. L'agricultura, la ramaderia, les comunicacions, la domòtica i els transports són algunes d'elles.



*Figura 1. 5.- Rellotge despertador, un exemple d'automatització.*



### 1.2.2.- Sistemes de control

Un robot es pot considerar com un conjunt de sistemes automàtics. Per tal de dissenyar i programar un robot és necessari descriure cada procés i estudiar el seu funcionament. En funció de la forma de treballar de l'automatisme podem distingir 2 tipus de control:

**A) Sistema de control de llaç obert:** Aquest tipus de control es caracteritza perquè, un cop realitzada l'acció programada, no controla si s'ha complit l'objectiu, sinó que només fa l'acció (Figura 1. 6 a). Un exemple d'aquest tipus de sistema de control és el funcionament d'un llum temporitzat. El llum està encès un temps determinat i s'apaga sense garantir que l'usuari ha finalitzat la tasca.

**B) Sistema de control de llaç tancat:** A diferència de l'anterior, aquest sí que controla que es compleixi l'objectiu (Figura 1. 6 b). Normalment, per comprovar si s'ha complit l'objectiu, el robot utilitzarà sensors. Un exemple de control de llaç tancat podria ser el control de temperatura d'un sistema d'aire condicionat o calefacció. L'aparell s'encén o s'apaga en funció de si la temperatura ambient està per sobre o per sota de la temperatura de consigna definida per l'usuari.

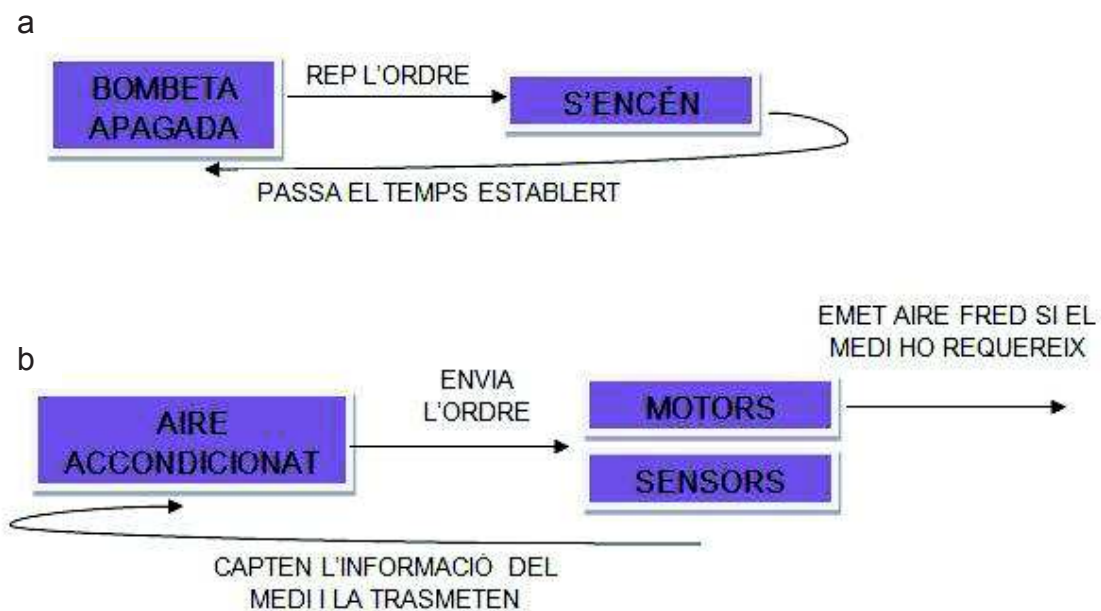


Figura 1. 6 a) Exemple de sistema de control de llaç obert i b) Exemple de sistema de control de llaç tancat



### 1.3.- Concursos de Robòtica

Durant els últims anys, la robòtica s'ha anat introduint al món educatiu fins al punt que en la gran majoria de les escoles del nostre país, a les aules de tecnologia es treballa amb robots. Dins del nostre centre, treballem amb robots a la matèria de tecnologia de 4t d'ESO i una optativa a 2n d'ESO (Electronics and Robotics). Des dels més petits fins als universitaris utilitzen aquesta disciplina per aprendre nous conceptes o consolidar-ne altres. Durant aquestes classes, es pretén que els alumnes s'introdueixin en aquest món que cada cop és més gran i complex.

Aquesta darrera dècada, han començat a aparèixer competicions i concursos de robòtica amb diferents categories: robots seguidors, hominoides, sumo, minisumo, etc. Robolot<sup>2</sup> (realitzat a Olot i organitzat pels instituts de la comarca), AESSbot<sup>3</sup> (realitzat a Barcelona i organitzat per l'associació AESS de l'UPC) o Robolid<sup>4</sup> (realitzat a Valladolid i organitzat per la Univesidad de Valladolid) són algunes de les competicions més populars que podem trobar a Espanya.

Dins de les seves categories, trobem la competició de *mini sumo*, on el meu robot compleix tots els requisits per poder ser inscrit. A part de la categoria esmentada, trobem d'altres com la de sumo (amb robots més grans), rastrejadors, velocistes, programació i projectes personals.

El funcionament dels concursos de sumo i mini sumo són molt similars: els robots es mouen dins d'un ring circular amb l'objectiu de xocar amb l'oponent i treure'l del mateix (Figura 1.7).



*Figura 1. 7 .- Foto del Robolid 2015.*

---

<sup>2</sup> Robolot: **Robot Olot**

<sup>3</sup> AESSbot: **AESS** (Associació universitària de la ETSEETB-FIB UPC) robot

<sup>4</sup> Robolid: **Robot Valladolid**



## 2.- Llenguatges de programació

### 2.1.- Història de la programació

Avui en dia, trobem un gran nombre de llenguatges de programació els quals s'adapten de una manera òptima a les nostres necessitats, però durant l'història no sempre ha sigut així de fàcil.

Els primers llenguatges daten entorn al 1940, encara que ja al segle XIX, es van veure els primers intents de llenguatge, encara que de manera molt simple i, evidentment, sense computadores electròniques. El primer llenguatge de programació va ser el *Plankalkül*, el qual va ser creat per Konrad Zuse l'any 1943. Entre 1950 i 1960, van començar a aparèixer els primers llenguatges de programació moderns, a partir dels quals han evolucionat els actuals. Alguns d'ells són el *FORTRAN*<sup>5</sup> (1955), *LISP*<sup>6</sup> (1958) i el *COBOL*<sup>7</sup> (1959), però el més important va ser el *ALGOL60*<sup>8</sup>, creat al 1960 per un comitè europeu i americà de científics, el qual va destacar per algunes característiques úniques (estructures en blocs, lèxic propi i notacions matemàtiques exactes).

A partir de la dècada dels 70, van començar a desenvolupar-se els llenguatges que avui dia són molt utilitzats, com per exemple el C, que serà el llenguatge amb el qual realitzar el meu Treball de Recerca i que aprofundiré més endavant en la seva estructura i síntesi. L'existència de diversos llenguatges, amb estructures i funcions diferents, ens ha permès crear programes amb característiques que s'adeqüen de manera òptima les nostres necessitats. Podem destacar llenguatges com el *Java Script* (1995), el qual s'utilitza en la creació de videojocs, ja que la seva estructura es idònia per a aquesta funció. Un altre llenguatge a destacar és el *Visual Basic* (1991), un llenguatge similar al *Java Script*, però més senzill de programar i amb menys possibilitats de creació.

Els llenguatges de programació, sovint són classificats segons el nivell, a partir d'aquí podem distingir tres tipus: Alt, mig i baix.

<sup>5</sup> FORTRAN: *Formula Translating System*

<sup>6</sup> LISP: *LISt Processing*

<sup>7</sup> COBOL: *COmmon Business-Oriented Language*

<sup>8</sup> ALGOL60: *Algorithmic Language 60*



Els llenguatges de nivell baix es caracteritzen per la seva simplicitat i per la dependència total del hardware. Respecte als de nivell alt, es caracteritzen per ser més similars al llenguatge natural, i no tant al llenguatge màquina. A més a més, pot ser migrat d'una màquina a una altra. Per últim, el de nivell mig es caracteritza per ser una combinació d'ambdós.

## 2.2.-El llenguatge de programació C

Tal i com abans he esmentat, el llenguatge de programació C va ser creat entorn l'any 1973 als laboratoris Bell, situats a Nova Jersey i creats per l'empresa AT&T l'any 1925, al capdavant del qual trobem a Denis Ritchie, acompanyat de molts més treballadors. A aquest laboratoris no només van crear el llenguatge C, sinó que tenen la patent de grans invents com per exemple la fibra òptica, el laser, la telefonia mòbil, etc.

Aquest llenguatge és molt utilitzat a l'hora de programar petits robots o plaques de desenvolupament de projectes de l'estil d'Arduino, però amb petites variants. Per a programar el meu robot, ha sigut necessari utilitzar aquest llenguatge.

En aquest apartat explicaré de forma resumida, els diferents elements, estructures i funcions necessàries per programar en llenguatge C.

### 2.2.1.- Variables i Constants

Dins d'un programa, podem assignar-li a qualsevol lletra o paraula (sempre que no sigui de les reservades prèviament per el llenguatge) un cert valor. Depenent de si a aquesta lletra o paraula volem canviar-li o no el valor durant el programa l'anomenarem variable o constant.

En una variable, tal i com el seu nom indica, el valor pot canviar en el transcurs del programa. Podem trobar diferents tipus de variables: int (per a valors enters), float (per a valors decimals) i char (per a caracters). Per a poder utilitzar-les han de ser declarades abans de ser utilitzades. Les podem declarar de la següent manera:



- En el cas que només la volguéssim declarar, sense haver de donar-li un valor ho faríem així:

```
tipus nom_variable;
```

*Exemple:* int preu;

- Si volem declarar més d'una variable alhora, ho realitzariem de la següent forma:

```
tipus nom_variable1, nom_variable2;
```

*Exemple:* char preu1, preu2;

- Per últim, si volem donar-li un valor a la nostra variable ho realitzem així:

```
tipus nom_variable = valor;
```

*Exemple:* float preu3 = 4.5;

Un darrer aspecte a tenir en compte, és que les variables poden ser globals (durant tot el programa) o locals (únicament en una funció, que esmento més endavant a l'apartat 3.2.7).

A diferència d'una variable, una constant no canvia el seu valor durant el programa. D'una manera igual a la de les variables, s'han de definir abans de ser utilitzades. N'hi ha de 2 tipus: literals (utilitzem directament el valor i no es necessari definir-la prèviament) i simbòlica (se li assigna a una paraula o lletra un valor). Per definir una constant simbòlica ho farem així:

- #define nom\_constant valor\_constant      *Exemple:* #define IVA 21





### 2.2.2.- Operadors

De la mateixa manera que quan realitzem operacions matemàtiques, al llenguatge de programació C, és necessari utilitzar operadors. Els operadors són molt útils quan volem que el nostre programa realitzi una sèrie d'accions en funció de si compleix o no una determinada condició. Els operadors que utilitzaré per a la programació del meu robot són els següents:

- **Assignació:** està compost únicament pel signe "=", i s'utilitza per assignar un valor a una variable. *Exemple:* IRPF = 4
- **Aritmètics:** Aquest tipus d'operadors s'utilitzen per realitzar operacions matemàtiques: suma (+), resta (-), multiplicació (\*) i divisió (/).
- **Racionals:** són utilitzats per comparar valors. Podem trobar el major que ">", menor que "<", major igual que ">=", menor igual que "<=", igual que "==" i diferent de "!=".
- **Lògics:** són els mateixos que s'utilitzen a la electrònica digital: Destaquen l'operador lògic AND "&&", l'OR "||" i el NOT "!".

A més a més, a l'igual de les matemàtiques, hi ha un ordre de prioritat que assigna quins operadors operen abans que altres.

| PRIORITAT DELS OPERADORS |                      |                             |
|--------------------------|----------------------|-----------------------------|
| 1.-                      | Parèntesis           | ()                          |
| 2.-                      | Negació              | !                           |
| 3.-                      | Operadors Aritmètics | +, -, ++, --, *, /, %, *, - |
| 4.-                      | Operadors Racionals  | <, <=, >, >=, ==, !=        |
| 5.-                      | Operadors Lògics     | &&,                         |
| 6.-                      | Assignacions         | =, +=, -=, *=, /=, %=       |

Figura 2. 3.- Taula de l'ordre de prioritats dels operadors.



### 2.2.3.- Sentències de Control de Flux

Les sentències de control de flux permeten que el nostre programa pugui realitzar una sèrie d'accions dependent d'una o unes condicions prèvies. Podem trobar de dos tipus, les condicionals i les iteratives. Dins de les condicionals, hi ha el If - Else If - Else i el Switch.

- *If - Else If – Else*. És el tipus de condicional més bàsic, en funció de si es compleix la condició, realitza l'acció. En el cas de que aquesta no es compleixi, poden haver més condicions, imposades per un “else If”, el qual té el mateix funcionament que un “If”. Per últim, si no es compleix cap condició, podem posar un “else”, el qual s'encarrega de realitzar una acció si prèviament no s'ha complert cap condició.

#### EXEMPLE DE IF – ELSE IF - ELSE

```
int a = 3;
int b;
if (a == 4)
{
    b = a;
}
else if (a != 3)
{
    b = 2;
}
else
{
    b = 1;
}
```

**Figura 2. 4.-** Exemple d'utilització d'una estructura condicional tipus *if - else if – else*.

- *Switch*: Aquest tipus de condicional s'utilitza per comparar una expressió amb uns valors, els quals anomenarem “case” (cas) . Si aquesta expressió coincideix amb algun valor, es realitzarà una acció. En el cas de que no coincideixi amb l'expressió inicial, podem incorporar un “default”, el té una funció molt semblant a el “else”. Per tancar qualsevol *case* o *default*, s'ha de fer amb un *break*;



```
ESTRUCTURA SWITCH

switch (expressio)
{
case valor1:
    sentencia1;
    break;
case valor2:
    sentencia2;
    break;           //tants casos com volguem
case valor_n:
    sentencia_n;
    break;
default:
    sentencia_x;     //el default és opcional
    break;
}
```

Figura 2. 5.- Exemple d'utilització d'una estructura condicional tipus Switch.

En el cas de les sentències iteratives, trobem de tres tipus:

- *While*: Aquesta sentència té la funció de repetir lo que nosaltres desitgem tants cops com la condició es compleixi. A més a més, podem crear bucles infinits com a condició posem algun dels següents termes: 1, TRUE.

```
ESTRUCTURA WHILE

while (<Condicio>)
{
    sentencies;
    /*Conjunt de sentencies que es repetiran sempre
    que la condició és certa*/
}
```

Figura 2. 6.- Exemple d'utilització de la sentència iterativa While.

- *Do - While*: L'única diferència amb la sentència "While", és que independentment de si es compleix o no la condició, el primer cop realitza l'acció que hi ha en ell.

```
ESTRUCTURA DO - WHILE

do
{
    sentencies;
    /*Executa les sentencies de dins del do – while,
    després les sentencies que es repetiran sempre
    que la condició és certa*/
} while (<Condicio>)
```

Figura 2. 7.- Exemple d'utilització de la sentència iterativa Do-While.



- For: Per últim, el For ens permet, abans de començar el bucle, donar un valor a una variable, imposar una condició i la acció que volem realitzar un cop la finalitza, aplica la instrucció, comprova la condició i si es compleix torna a realitzar l'acció.

**ESTRUCTURA FOR**

```
for (sentencia_inicial; condicio; sentencia_fi_iteracio)
{
    sentencies;
    /*Conjunt de sentencies que s'executen mentre es
    compleixi la condició*/
}
```

**Figura 2. 8.-** Exemple d'utilització de la sentència For.

## 2.2.4.- Funcions

Un programa en "C", està compost per una funció principal, anomenada void main (void), i per una sèrie de funcions secundàries, les quals són opcionals. Una funció pot ser definida com una petita part del programa que realitza una acció concreta. Per poder utilitzar una funció secundària, ha hagut de ser definida prèviament. Un cop definida, és indispensable cridar la nostra funció per incorporar-la al programa principal. Cridarem la nostre funció secundària de la següent forma:

- nom\_funcio (paràmetres);

**ESTRUCTURA**

```
tipus nom_funcio (tipo1 param1, ..., tipoN paramN)
{
    cos
}
```

**EXEMPLES**

```
void multiplicacio(float num1, float num2)
{
    float resultat;
    resultat = num1 * num2;
    printf("El resultat es: %f\n", resultat);
}
```

**Figures 2. 9 a i b.-** Exemple i estructura d'una funció.



Finalment, podem fer que el nostre programa retorni alguna cosa canviant el “void” del inici per un “int, float o char” i utilitzant la següent frase com a última de la funció: “return cosa\_a\_retornar”.

### EN EL CODI PRINCIPAL

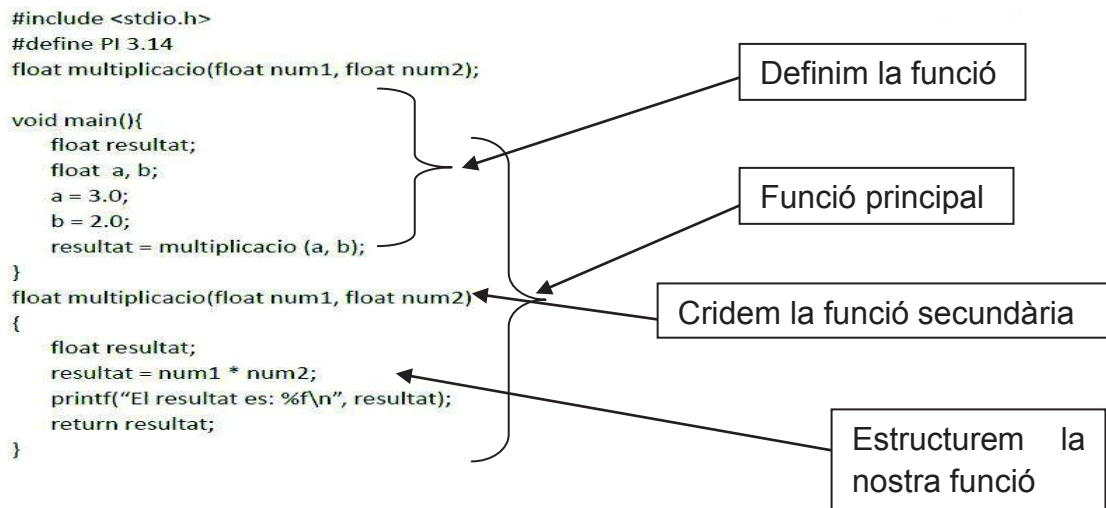


Figura 2. 10.- Exemple d'un programa que utilitza funcions secundàries.

## 2.3.- Llibreries

L'última part a destacar sobre el llenguatge C, són les llibreries. Una llibreria es pot definir com una part reservada on trobem una sèrie de funcions ja creades de les quals no necessitem conèixer la seva estructura. Per poder utilitzar-les, les hem de declarar prèviament de la següent manera: #include <nom\_llibreria.h>. Com a llibreries més utilitzades en el món del C, trobem les següents:

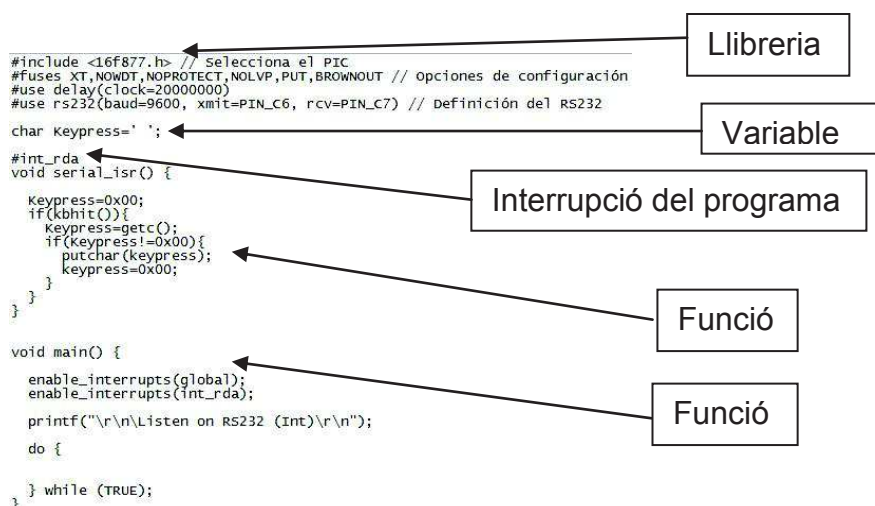
- <stdio.h>: És indispensable per poder interactuar amb l'exterior. També té les funcions *printf* i *scanf*, encarregades de imprimir per pantalla i escanejar per teclat respectivament.
- <math.h>: En aquesta llibreria trobem tot allò necessari i bàsic per a la realització d'operacions matemàtiques.



## 2.4.- Estructura d'un programa en C

A l'hora de realitzar un programa en C, s'ha de seguir una certa estructura per tal que es pugui processar de manera adient el programa i, realitzar així, les accions que nosaltres hem establert.

El primer element que ha d'aparèixer al nostre programa són les llibreries de la manera esmentada prèviament. A continuació, definiríem les constants en el cas que hi haguessin. Tot seguit, trobaríem les capçaleres de les funcions secundàries, les quals estructurarem més endavant. Després, definiríem les variables globals si en són necessàries. D'altra banda, trobem lo més important de tot el programa, la funció principal o "main". Per últim, estructurarem les funcions secundàries, les quals seran utilitzades en la funció principal, tal i com he explicat en els apartats anteriors.



|  |
|--|
| Declaraci3n de llibreries  |
| Definici3n de constants  |
| Capçaleres de les funcions que utilitzarem al programa i variables globals |
| void main(void){<br>//codi main<br>}                                       |
| Estructura de les funcions que hem declarat prèviament al main             |

Figura 2. 11 a i b.- Taula de com estructurar un programa en C i exemple d'un programa en C segons l'estructura anterior.



### **3.- Elements de disseny del robot**

Al llarg d'alguns dels anteriors capítols, he estat descrivint gran part de les característiques del llenguatge de programació que he utilitzat per programar el meu robot però, com és lògic, sense *hardware*<sup>9</sup> és impossible que el *software*<sup>10</sup> programat ens sigui útil. En aquest capítol parlaré sobre el *hardware* del meu robot, és a dir, sobre els components que el formen i el seu funcionament.

#### **3.1.- Especificacions de disseny**

El robot que he dissenyat i implementat és un robot de sumo i té com a finalitat participar i guanyar combats de sumo segons les regles establertes als concursos de robòtica. L'objectiu del meu prototipus als combats de sumo és treure a l'oponent del ring evitant que el nostre robot surti del mateix. Com és lògic, és impossible recrear al cent per cent combats de sumo amb robots, però si que arriba a ser molt semblant.

Així mateix, han de complir unes certes especificacions de disseny, que en el meu cas són les següents:

- 1.- Ha d'estar equipat amb dos servomotors de corrent continu.
- 2.- Ha d'estar equipat amb cinc sensors, dels quals dos seran de distància i tres de línia.
- 3.- Ha d'estar alimentat amb dues piles de 6 i 9 volts, les quals subministren corrent a diferents parts del circuit.
- 4.- Quan no detecta cap oponent s'ha de moure seguint una trajectòria en forma d'estrella.
- 5.- Quan els sensors de línia detecten el color blanc que delimita el final del ring, el robot ha de retrocedir.
- 6.- Quan els sensors de distància detecten un oponent, el robot ha d'avançar en direcció a l'oponent.

<sup>9</sup> Hardware: Mot referent a la part física del sistema.

<sup>10</sup> Software: Mot referent als programes del sistema.



### 3.2. Components del robot

Com he comentat abans, un robot es pot considerar un sistema automàtic i, com a tal, està constituït per les següents parts: controlador, actuadors, sensors i elements de comandament manual. A continuació, explicaré els elements que constitueixen el meu prototipus.

#### 3.2.1.- Controlador

El controlador és el dispositiu del sistema que executa el programa que determina el funcionament del robot. És l'encarregat de processar tota la informació i de donar resposta a tots els possibles problemes que puguin aparèixer durant el seu funcionament.

##### 3.2.1.1.- Descripció de la placa controladora

Abans d'explicar el controlador, és necessari fer cinc cèntims sobre la placa controladora, element indispensable per al funcionament del controlador.

Una placa controladora és la interfície física on es connecta el controlador, ja que no es pot connectar directament a sensors o actuadors. A més a més, serveix com a medi de transmissió entre els elements esmentats prèviament.

La placa utilitzada per aquest projecte Figura 3.1 no és una placa comercial, tipus *Arduino UNO*, on venen preestablerts els punts de connexió dels diversos elements, sinó que permet personalitzar-la segons les necessitats de cada projecte. Les seves reduïdes dimensions permeten que s'integri de manera excel·lent a l'estructura del robot.





Com a font d'alimentació, té dues piles de diferent voltatge, una de 6V i l'altra de 9V (dels quals només s'utilitzen 6V) que s'encarreguen de proporcionar energia a les diferents parts del circuit encara que no alimenten les mateixes parts del circuit. La pila de 6V s'encarrega d'alimentar els servomotors, ja que és la part que més voltatge requereix. L'altra pila controla l'alimentació de la resta de components electrònics del robot (microprocessador, sensors, etc.).

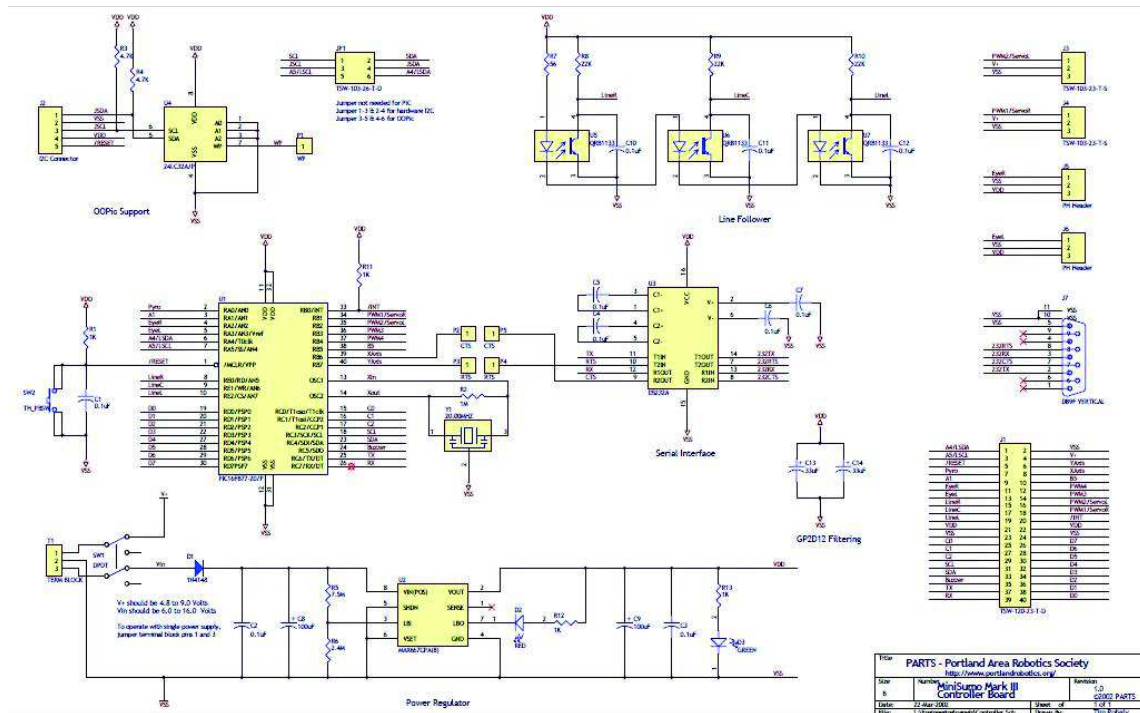


Figura 3. 1.- Circuit electrònic de la placa.

Com es pot veure a la Figura 3.1, el circuit electrònic de la placa és força complex, ja que integra tots aquells elements i connexions necessàries per tal que el microcontrolador pugui realitzar totes les funcions per les quals ha estat programat.

A la Figura 3.2 podem observar les diferents parts que formen la placa controladora.

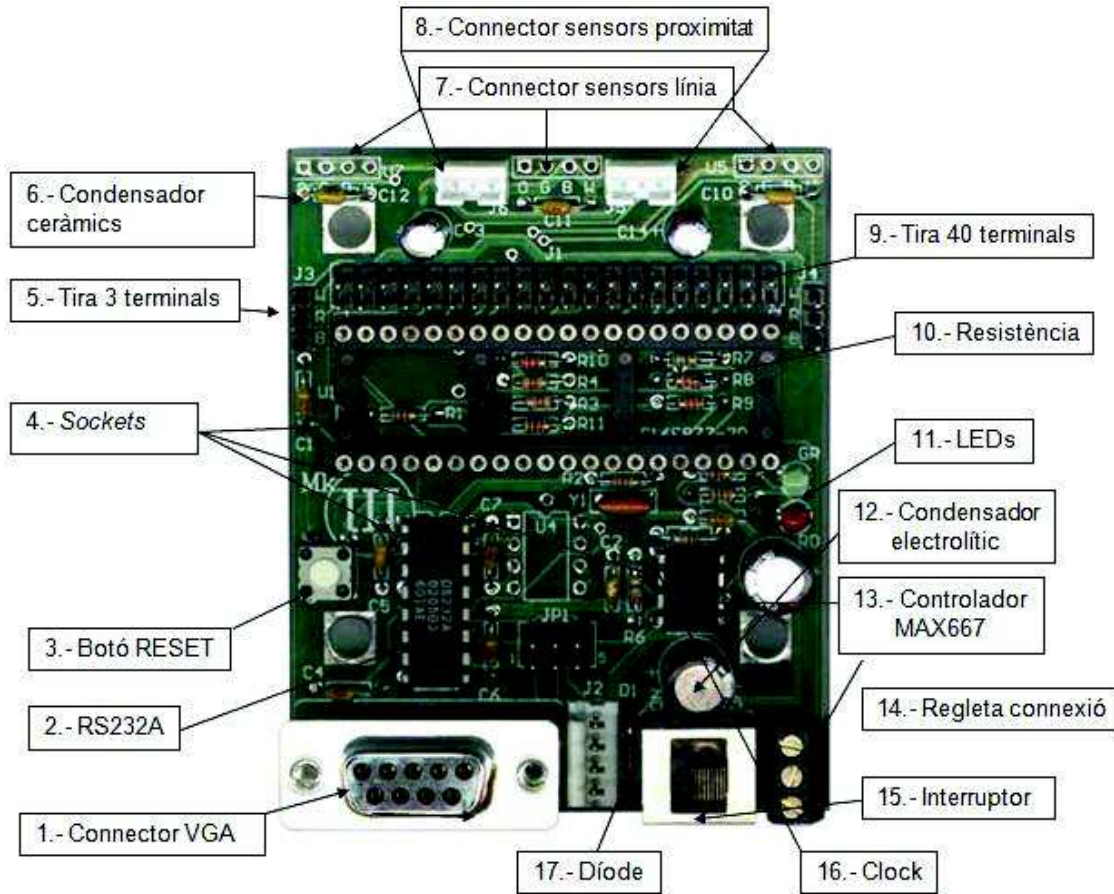



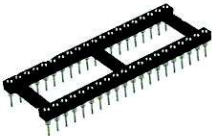




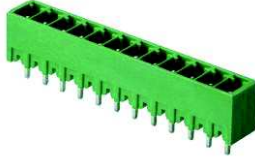
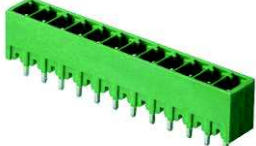





Figura 3. 2.- Placa utilitzada en el projecte.

La figura 3.3 mostra una taula amb tots els elements i components presents a la placa de control. Bàsicament, es tracta d'elements de connexió, senyalització (LEDs), i xips controladors (*RS232A* i *MAX667*). A la taula s'enumeren els components electrònics emprant el seu nom tècnic i es resumeix la funció que realitzen al meu prototipus. D'altra banda, es mostra una fotografia per facilitar el seu reconeixement visual.



| Nom                                    | Foto  | Funció   |
|--|---|--|
| <b>1</b> Connector femella DB9 VGA     |    | Permet la transmissió d'informació entre la placa i l'ordinador. |
| <b>2</b> Controlador RS232A            |    | Controla la transmissió d'informació.                            |
| <b>3</b> Botó RESET                    |   | Reinicia el programa guardat al microprocessador.                |
| <b>4</b> Sockets                       |  | Permet una connexió més polida entre microprocessador i placa.   |
| <b>5</b> Connector per als servomotors |  | Permet la connexió entre la placa i els servomotors.             |
| <b>6</b> Condensador ceràmic           |  | Component electrònic que emmagatzema energia elèctrica.          |



|   |   |  |
|---|---|--|
| <p><b>7</b> Connector dels sensors de línia</p>         |    | <p>Permet la connexió entre la placa i els sensors de línia.</p>                   |
| <p><b>8</b> Connector per als sensors de proximitat</p> |    | <p>Permet la connexió entre la placa i els sensors de proximitat.</p>              |
| <p><b>9</b> Tira de 40 pins o terminals</p>             |    | <p>Permet la connexió entre el microprocessador i algun element extern.</p>        |
| <p><b>10</b> Resistència</p>                            |  | <p>Element que regula el corrent elèctric.</p>                                     |
| <p><b>11</b> LEDs</p>                                   |  | <p>Actuadors lluminosos que ens indiquen el correcte funcionament de la placa.</p> |
| <p><b>12</b> Condensador electrolític</p>               |  | <p>Component que emmagatzema energia elèctrica.</p>                                |
| <p><b>13</b> Controlador MAX667</p>                     |  | <p>Controlar el flux de corrent.</p>   |







|                               |  |   |
|-------------------------------|--|---|
| <p>14 Regleta de connexió</p> |   | <p>Permet la connexió de les piles a la placa.</p>                                  |
| <p>15 Interruptor</p>         |   | <p>Permet controlar el pas del corrent (ens permet encendre o apagar el robot).</p> |
| <p>16 Clock</p>               |   | <p>Temporitzador del programa.</p>  |
| <p>17 Diode</p>               |  | <p>Ajuda a controlar el flux de corrent elèctric.</p>                               |

Figura 3. 3.- Taula explicativa dels elements de la placa

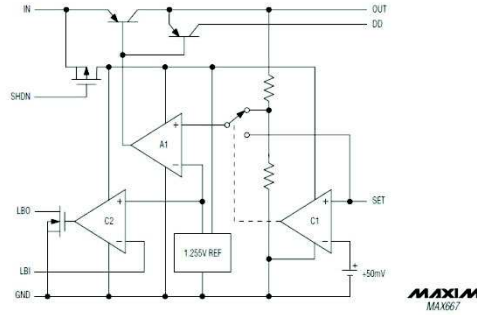
Els xips controladors corresponents als elements 2 i 13 de la taula de la figura 3. 3, no són tant importants com el controlador principal, el qual explicaré més endavant, però són indispensable per al correcte funcionament del robot. El primer s'anomena *RS232A* i està comercialitzat per l'empresa *DALLAS Semiconductor*. Disposa de 16 pins i és l'encarregat de la transmissió d'informació. Aquesta transmissió pot ser tant de manera interna com externa.



Figura 3. 4.- RS232A



Finalment, el microcontrolador *MAX667*, està equipat amb 8 pins i la seva funció és controlar la tensió subministrada per les bateries per a que no es cremi res i tingui un funcionament adient.



**Figura 3. 4 a i b.-** MAX667 i el seu diagrama de

blocs.

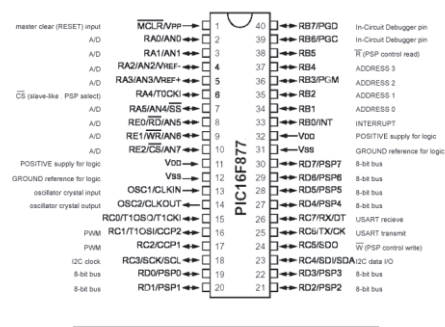
**3.2.1.2.- Descripció del controlador: PIC 16F877A**

Després de parlar sobre pràcticament tota la placa, només ens queda per conèixer la part més important, el *PIC 16F877A*. Aquest model concret es creat per l'empresa *MICROCHIP* i disposa de 40 pins. D'aquest 40 pins, hi ha alguns que no són necessaris per al funcionament del meu prototip, però la gran majoria són imprescindibles.

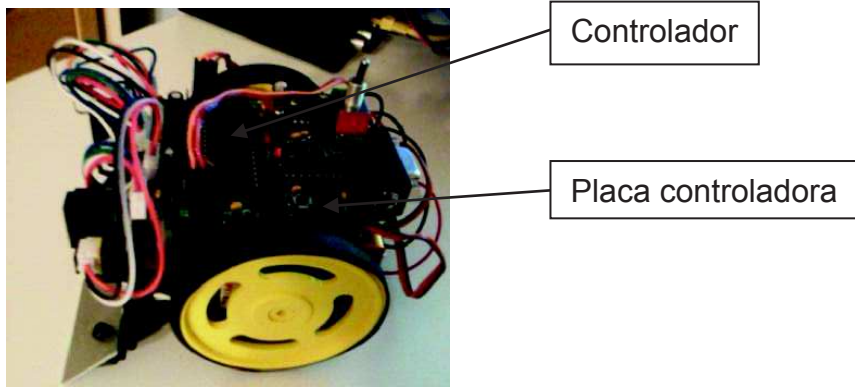
L'elecció d'aquest model és bàsicament perquè el model inferior no tindria suficient potència i el superior té molts pins o ports de sortida/entrada que no caldria connectar.



**Pin Diagram PDIP**



**Figura 3. 6 a i b.-** PIC, Esquema del PIC i les seves connexions.



*Figura 3. 7.- Localització de la placa controladora i el controlador al robot*

Per a la realització d'aquest projecte, he utilitzat els següents pins dels ports d'entrada/sortida:

- Pin 3: Connexió del botó RESET.
- Pins 6 i 7: Connexió dels sensors de distància.
- Pins del 8 al 10: Connexió dels sensors de línia.
- Pins 28 i 29: Connexió dels servomotors.

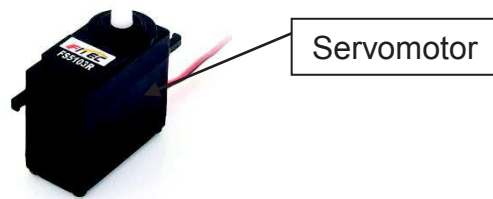
En el cas de les tres últimes connexions, s'ha realitzat un pont per facilitar la connexió.

A part dels pins esmentats, n'hi ha d'altres com els que van des de el 33 fins al 40 els quals es poden connectar LEDs, sense que afectin al funcionament del robot.

### 3.2.2.- Actuadors

Un actuator es podria definir com un dispositiu amb la capacitat de transformar energia elèctrica, en altres formes d'energia com pot ser la mecànica o tèrmica. En robòtica els actuadors més utilitzats són els motors i els relés.

En aquest cas, he escollit un servomotor, més concretament, el model escollit és el FS5103R de l'empresa FITEC. Aquest element, amb l'ajuda d'unes rodes que es connecten a la part giratòria, permeten que el robot es pugui moure dins d'un rang de 360°. Així mateix, permet ajustar amb molta precisió la velocitat i sentit de rotació de les rodes. Per últim, també inclou un petit cargol per poder establir el valor base a partir del qual les rodes no giren.



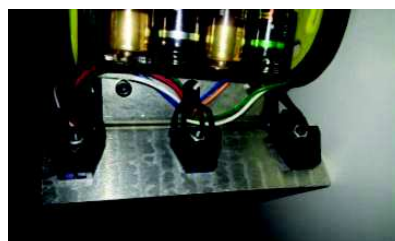
*Figura 5. 8 a i b.- Servomotor i localització en el robot.*

### **3.2.3.- Sensors**

Els sensors són els elements que permeten a un sistema automàtic controlar-se sense que hagi de intervenir l'home. Podem trobar de moltíssims tipus, però en aquest cas he utilitzat els sensors de línia i de distància.

#### **3.2.3.1.- Sensors de línia**

El següents elements a estudiar són els sensors de línia. Al meu prototip he utilitzat tres sensors encarregats d'evitar que el robot surti del ring ell sol. El seu funcionament es basa en distingir diferents tonalitats de colors. Depenent del color que detecti i de com ha estat programat, podem fer que, per un determinat color o fins i tot tonalitat, realitzi un moviment o un altre. Per a aquest robot, he utilitzat tres exemplars (situats al centre, dreta i esquerra) per tal d'evitar que surti lo mínim del ring.



*Figura 5. 9.- Sensors de línia.*



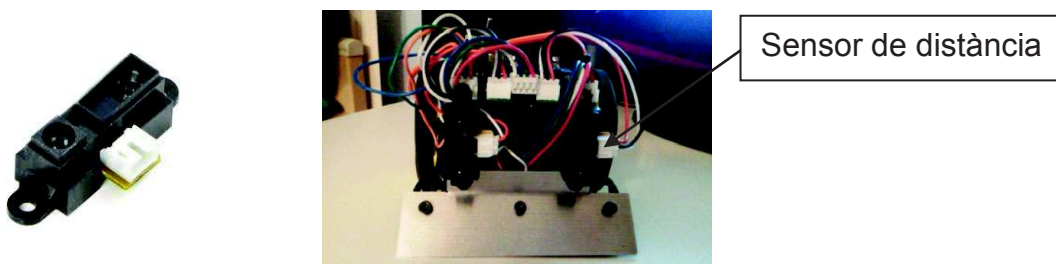


### 3.2.3.2.- Sensors de distància

Per últim, parlaré sobre el que col·loquialment es coneix com a ulls d'un robot, els sensors de distància. En aquest cas concret, el model utilitzat es el GP2D12 de l'empresa SHARP.

Aquest són els encarregats de que quan s'apropi un altre robot, el detecti i s'executin les accions programades amb l'objectiu de treure fora del ring a l'oponent. El meu prototipus disposa, doncs, de dos sensors de distància.

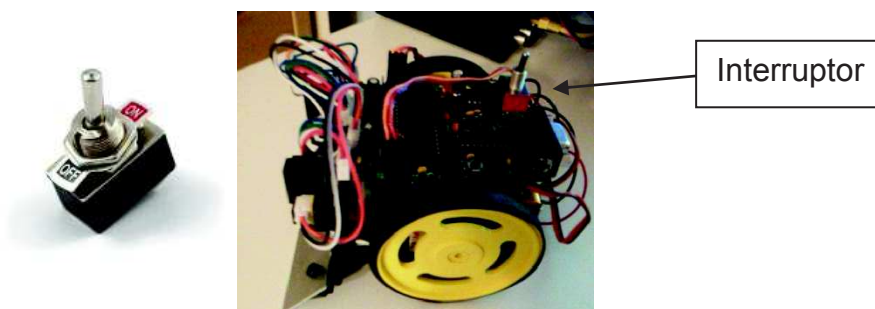
Una característica que tenen en comú els dos tipus de sensors utilitats en el projecte és la metodologia que utilitzen per detectar un objecte. Ambdós tenen un LED d'infrarojos i un fototransistor que s'encarreguen de detectar, en un cas un color i en l'altre la proximitat.



*Figura 3. 10.- Sensors de distància i localització al robot.*

### 3.2.4.- Elements d'accionament manual

Els elements d'accionament manual són els components que ens permeten donar ordres al robot per a que ell les pugui executar. En el cas del meu robot només disposa d'un. Aquest element d'accionament manual és un interruptor de palanca el qual té com a funció encendre i apagar el robot.



*Figura 3. 11.- Interruptor i localització al robot.*



## **4.- Disseny, Implementació i Programació del meu robot**

---

### **4.1.- Disseny i implementació del prototipus**

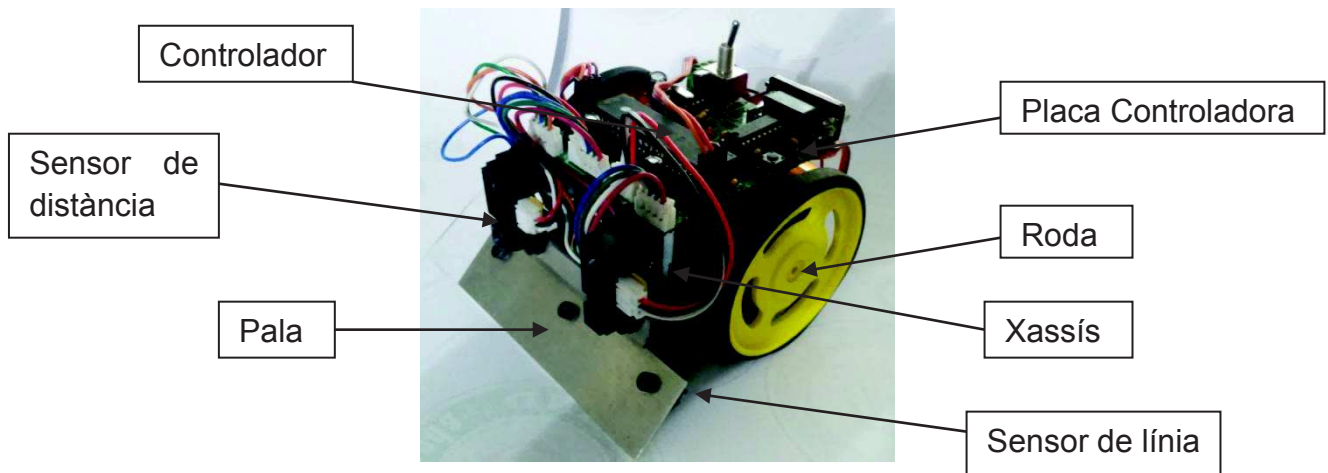
#### **4.1.1.- Disseny del prototipus**

Abans de començar amb el disseny i construcció del meu prototipus, vaig haver de plantejar-me com seria l'estructura del robot per tal de complir amb les principals especificacions de disseny: facilitat de muntatge i alt rendiment durant els combats. També havia de tenir en compte els materials i les dimensions mínimes per allotjar tots els components.

Després de reflexionar sobre aquestes qüestions, vaig haver de descartar alguns dissenys donats que no complien les especificacions de disseny o bé degut al seu elevat cost econòmic. Finalment, vaig escollir aquest disseny ja que té un cost molt reduït de producció, és força resistent i senzill d'acoblar. Un altre motiu pel qual vaig decidir utilitzar aquest disseny era pel fet que únicament utilitza dues rodes, ja que la part davantera està equipada amb una pala que, a part d'eliminar la necessitat de dur dues rodes més, serveix per empènyer els rivals fora del ring i augmentar el seu rendiment. La pala i tota l'estructura està feta d'alumini, un material fàcil de trobar i de manipular.

En relació als elements de la carcassa, s'han construït amb l'ajuda de l'AEISS que m'ha proporcionat les peces necessàries amb els millors acabats.

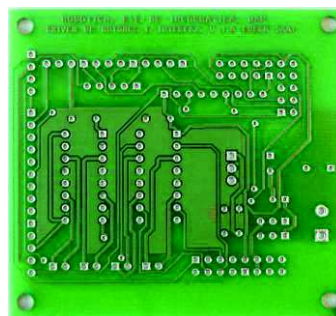
Com es pot veure a la Figura 4.1, l'estructura del robot consta de dues parts. D'una banda podem distingir una estructura que serveix de suport als actuadors i unes petites barres, sobre de les quals es col·loca la placa controladora. D'altra banda, és necessària una segona estructura en forma de pala que va unida a la part principal abans esmentada. Per últim, s'uneixen els sensors, tant de línia com de distància a la pala. Totes aquestes parts es fixen mitjançant cargols de diferents mides.



*Figura 4. 1.- Foto del robot i especificació de les parts*

### 4.1.2.- Implementació del prototipus

La construcció d'aquest robot s'ha realitzat en dues fases. La primera fase consisteix en la implementar la placa de circuit imprès. Com que no es tracta d'una placa comercial, ha estat necessari soldar tots els components. Malgrat no ser una tasca de gran desgast mental, requereix una concentració i precisió important, ja que les soldadures no poden ser ni massa petites, perquè no funcionaria correctament, ni massa grans, pel fet que podrien tocar-se dues i produir un curtcircuit.



*Figura 4. 7.- Exemple de circuit imprès*

La segona fase consisteix en acoblar la placa i la resta de components a l'estructura. El resultat final d'aquest acoblament es pot observar a la Figura 4.1.



## 4.2.- Programació del prototipus

Gràcies als coneixements adquirits durant la meva estància a la UPC i explicats en capítols anteriors, he pogut programar el meu robot per la realització de combats de sumo. Abans de realitzar el programa s'ha de conèixer el funcionament detallat d'un combat de sumo. S'han de tenir en compte totes les possibles situacions en les quals es trobarà el robot i programar una possible solució. Aquestes condicions són evitar sortir del ring i intentar fer fora a l'oponent. Per a aconseguir la primera condició, el programa principal inclou un programa secundari que detecta el color que delimita el final del ring, fa un gir de 180° i s'allunyà de la vora. En canvi, per tal d'assolir segona condició, el robot està programat per empènyer a l'altre robot fins fer-lo fora, sempre i quan els sensors de distància l'hagin detectat.

Com es pot suposar, es tracta d'un programa força llarg i complex i l'he hagut de dividir en parts, tenint en compte quines són les més importants per a la comprensió del funcionament.

Degut a aquesta extensió, comentaré les parts més importants i si desitgeu consultar el programa complet, dirigiu-vos a l'Annex 1.

A la Figura 4.2, es pot observar la primera part del programa on, s'inclou la llibreria del controlador, es desactiven funcions preestablertes com el *WatchDog*<sup>11</sup> i es defineixen constants globals necessàries per a les funcions secundàries.

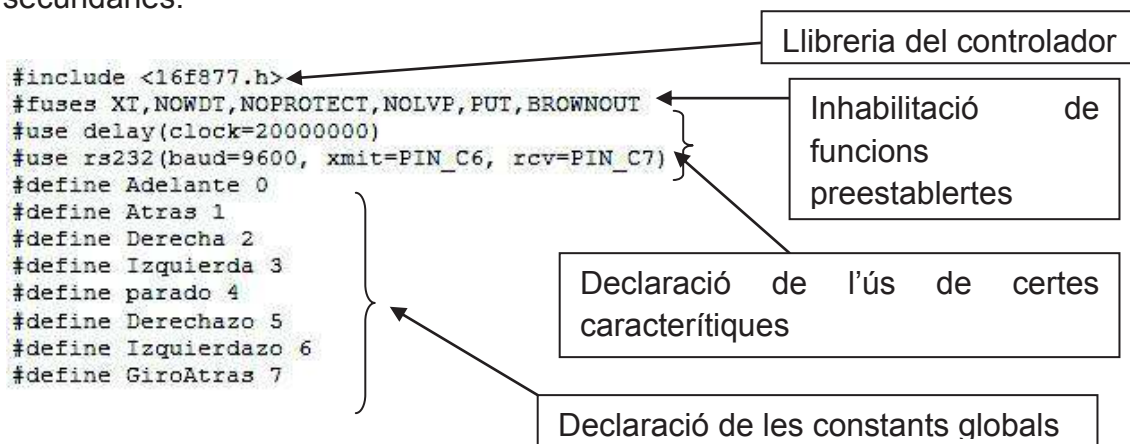


Figura 4. 2.- Primera part del programa final: llibreries, desactivacions i declaració de les constants.

<sup>11</sup> **WatchDog:** L'objectiu d'aquesta funció preestablerta del controlador és que es reinici cada cert temps per evitar que s'espatlli.



Seguidament, a la Figura 4.3, trobem la declaració de variables globals, també necessàries per als programes secundaris i per al principal.

```
int estado=0x04;
int Servo=0x00;
int dreta=0;
int centre=0;
int esquerra=0;
int dret=0;
int izq=0;
int adret=0;
int aizq=0;
int giral000 = 0;
int catras = 0;
int x = 0;
```

Declaració de les variables

Figura 4. 3.- Declaració de les variables globals.

A continuació, a la Figura 4.4, trobem les funcions secundàries, que s'encarreguen d'assignar els valors captats mitjançant els sensors de línia i de distància respectivament a les variables corresponents.

```
void toma_adc_y_transmite(void){
    // Lectura del canal 5 -> AN5 LM35a
    set_adc_channel(5);
    delay_ms(1);
    dreta=read_adc();
    delay_ms(1);

    // Lectura del canal 6 -> AN6
    set_adc_channel(6);
    delay_ms(1);
    centre=read_adc();
    delay_ms(1);

    // Lectura del canal 7 -> AN7
    set_adc_channel(7);
    delay_ms(1);
    esquerra=read_adc();
    delay_ms(1);
}

void toma_distancia(void){
    // Lectura del canal 2 -> AN2
    set_adc_channel(2);
    delay_ms(1);
    dret=read_adc();
    delay_ms(1);

    // Lectura del canal 3 -> AN3
    set_adc_channel(3);
    delay_ms(1);
    izq = read_adc();
}
```

Captar del medi

Assignació del valor captat a la variable

Captar del medi

Assignació del valor captat a la variable

Figura 4. 4.- Programes secundaris encarregats de copsar el medi.



Per al moviment del robot, s'utilitza una estructura de programació anomenada *Switch* que permet seleccionar una opció entre totes les possibles definides al programa amb els diferents moviments que pot realitzar el robot. A la Figura 4.5, podem observar un dels possibles moviments que pot realitzar.

```
Case Adelante:
//printf("adelante\n");
Switch(Servo){

    Case 0: output_high(PIN_B2);
            output_high(PIN_B1);
            set_timer1(65536-313);
            break;

    Case 1: output_low(PIN_B2);
            //output_high(PIN_B2);
            set_timer1(65536-1250);
            break;

    Case 2: //output_low(PIN_B1);
            output_low(PIN_B1);
            set_timer1(65536-10937);
            break;

}
```

Figura 4. 5.- Programa secundari encarregat del moviment del robot.

Per últim, trobem el programa principal. Com es pot observar a la Figura 4.6, aquest relaciona els valors captats mitjançant els sensors amb el diferents programes secundaris encarregats del moviment del robot per saber quins moviments ha de fer quan és a prop del final del ring o quan té un rival a una distància raonable per atacar.

```
void main() {
    set_tris_c(0xff);
    set_tris_d(0x00);
    output_bit(PIN_D7,1);
    printf("\r\n \t a x: %u \r\n",x);

    setup_adc(ADC_CLOCK_INTERNAL);
    setup_adc_ports(ALL_ANALOG);
    SETUP_TIMER_1(T1_INTERNAL|T1_DIV_BY_8);
    enable_interrupts(INT_TIMER1);
    enable_interrupts(global);
    SET_TRIS_E(0b00000111);
    set_tris_b(0x00);
    estado = Adelante;
    do {
        toma_adc_y_transmite();
        if ((dreta<100)|| (centro < 100)|| (esquerra < 100)){
            estado=Atras;
        }
        if(estado == Atras && catras < 50);
        else{
            if(estado == Atras) estado = GiroAtras;
            if(estado == GiroAtras && giral000 < 50);
            else{
                giral000 = 0;
                catras = 0;
                toma_distancia();
                if ((izq < 70 && dret < 70) || (izq >= 70 && dret >=
70)) {
                    estado = Adelante;
                    printf("\r\n Endavant \r\n");
                }
                else if(izq > 70) {
                    estado=Izquierda;
                    printf("\r\n Esquerra \r\n");
                }
                else if(dret > 70){
                    estado=Derecha;
                    printf("\r\n Dreta \r\n");
                }
                else {
                    estado = Adelante;
                    printf("\r\n Endavant \r\n");
                }
            }
        }
    }
}
```

Variacions del moviment del robot en funció de l'entorn

Figura 4. 6.- Part principal del programa final.



### Conclusions

- 1.- La integració de robots en el sistema educatiu ha suposat una nova metodologia de treball més interactiva i didàctica per a l'alumnat.
- 2.- Durant els últims anys, el nombre de concursos de robòtica va en augment, proporcionant un nou al·licient per als competidors, que desenvoluparan nous prototips més avançats tant en *software* com en *hardware*.
- 3.- Des dels inicis de la programació, s'han desenvolupat multitud de amb propietats, característiques i funcions diferents, amb l'objectiu de cobrir unes necessitats específiques de l'usuari.
- 4.- Els motius pels quals he escollit el llenguatge de programació C per programar el meu robot són: el nivell mitjà de programació, la multitud de possibilitats que ofereix envers al controlador utilitzat, i la compatibilitat amb la placa i els seus components.
- 5.- Per a realitzar un programa en llenguatge C, s'han de tenir en compte les diverses parts de les que consta: declaració de llibreries, definició de constants, capçalera de les funcions secundàries, definició de les variables globals, programa principal i estructura de les funcions secundàries.
- 6.- Per tal que el robot pugui participar en les competicions, ha de complir unes certes condicions de hardware, entre les quals destaquen el funcionament de servomotors i sensors.
- 7.-Malgrat les restriccions en el hardware del robot, el reglament de les competicions dóna molta llibertat en el hardware, permetent un munt de possibilitats en el programa.
- 8.- El component més important del meu robot és el controlador. Es considera el cervell de qualsevol robot i es l'encarregat d'executar el programa. Malgrat ser un component imprescindible per al funcionament, les seves capacitats es veurien molt limitades si no s'equipés d'actuadors i sensors.



9.- En relació al disseny de l'estructura, s'ha de tenir en compte que el prototipus disposi del major rendiment i funcionalitat amb el menor preu. Per aquest motiu, he decidit equipar el robot amb només dues rodes, la qual cosa permet una reducció dels costos i de la complexitat del programa.

10.- La realització d'aquest treball m'ha servit per tenir una experiència en aquest món i consolidar la meva idea de realitzar un Grau en Enginyeria relacionat amb la programació.





### Propostes de millora

---

Després de tots aquests mesos realitzant aquest treball, he viscut situacions amb les que he pogut aprendre nous conceptes i conèixer millor aquest món. Malgrat això, també he tingut alguns problemes tècnics que m'han endarrerit en la creació del robot. Gràcies a l'ajuda i el recolçament de AESS i de la meva tutora he pogut superar aquests problemes i portar a terme el meu robot.

A partir de la meva experiència proposo l'utilització d'un *hardware* més simple, amb una comprensió més ràpida i senzilla del seu funcionament. El fet d'haver utilitzat controladors com el *PIC 16F877A* els quals tenen una gran capacitat i potència, fa que siguin necessaris uns coneixements previs referents al seu funcionament, estructura i programació que al inici del treball no tenia.

Per a millorar aquest aspecte, proposaria la utilització d'alguna placa comercial tipus Arduino UNO. Malgrat la poca capacitat que disposen de personalització i creativitat, tenen un hardware i software més simple lo qual suposa una síntesi més ràpida del funcionament d'aquesta i menys dificultats alhora de la creació del robot.



## Llista de referències

### Bibliografia

- AESS. *Apunts curs de C*. Barcelona; AESS, 2015.
- BARRIENTOS, Antonio; PEÑIN, Luis Felipe; Balaguer, Carlos; Aracil, Rafael. *Fundamentos de robótica*. Madrid; McGraw–Hill, 2007.
- DURÁN, José Luis; GÁMIZ, Juan; BARGALLÓ, Ramón; DOMINGO, Joan; MARTÍNEZ, Herminio; MORÓN, Juan. *Electrotècnia. Instal·lacions elèctriques i automàtiques*. Barcelona; Marcombo, 2009.
- GARCIA, Encarna. *Apunts d'Automàtica i Robòtica*. Barcelona; Ins Rovira-Forns, 2010.
- GIAMARCHI, Frédéric. *Robots mòviles: Estudio y construcción*. Madrid; Ediciones Paraninfo, 2001.
- JOYANES AGUILAR, Luís; ZAHONERO MARTÍNEZ, Ignacio. *Programación en C*. Madrid; McGraw–Hill, 2005.
- REYES CORTÉS, Fernando. *Robotica: control de robots manipuladores*. Barcelona; Marcombo, 2011.
- SUBIR KUMAR, Sasha. *Introducción a la robótica*. Madrid; McGraw–Hill, 2010.
- TOJEIRO CALAZA, Germán. *Taller de Arduino. Un enfoque práctico para principiantes*. Barcelona; Marcombo, 2014.

### Webgrafia

- Historia de los lenguajes de programación* [en línia].  
[https://es.wikipedia.org/wiki/Historia\\_de\\_los\\_lenguajes\\_de\\_programaci%C3%B3n](https://es.wikipedia.org/wiki/Historia_de_los_lenguajes_de_programaci%C3%B3n) [consulta: 9.7.2015].
- C (lenguaje de programación)* [en línia].  
[https://es.wikipedia.org/wiki/C\\_\(lenguaje\\_de\\_programaci%C3%B3n\)#Ventajas](https://es.wikipedia.org/wiki/C_(lenguaje_de_programaci%C3%B3n)#Ventajas) [consulta: 9.7.2015].



*Tipos de lenguajes de programación* [en línea].  
<http://www.desarrolloweb.com/articulos/2358.php> [consulta: 9.7.15].

*Lenguaje de Programación* [en línea].  
[http://www.ecured.cu/Lenguaje\\_de\\_Programaci%C3%B3n](http://www.ecured.cu/Lenguaje_de_Programaci%C3%B3n) [Consulta: 12.7.15].

*Historia y definición de los lenguajes de programación* [en línea].  
<http://www.monografias.com/trabajos99/historia-y-definicion-lenguajes-programacion/historia-y-definicion-lenguajes-programacion.shtml> [Consulta 12.7.15].

*Which Programming Language Should I Learn First?* [en línea].  
<http://carlcheo.com/startcoding> [Consulta: 24.7.15].

*Historia del Lenguaje C* [en línea].  
[http://www.ecured.cu/Historia\\_del\\_Lenguaje\\_C](http://www.ecured.cu/Historia_del_Lenguaje_C) [Consulta: 30.7.15].

*Lenguajes de programación* [en línea].  
<http://www.areatecnologia.com/informatica/lenguajes-de-programacion.html>  
[Consulta: 30.7.15].

*Sistemas de control* [en línea].  
[http://recursostic.educacion.es/secundaria/edad/4esotecnologia/quincena11/4quincena11\\_contenidos\\_2a.htm](http://recursostic.educacion.es/secundaria/edad/4esotecnologia/quincena11/4quincena11_contenidos_2a.htm) [consulta: 5.8.15].

*Robótica* [en línea]. <https://es.wikipedia.org/wiki/Rob%C3%B3tica> [consulta: 5.8.15].

*Robolot* [en línea]. <https://sites.google.com/site/robolot/> [Consulta: 20.8.15].

*Cire* [en línea]. <http://webdelcire.com/wordpress/competicion-robotica-en-espana>  
[Consulta: 20.8.15].

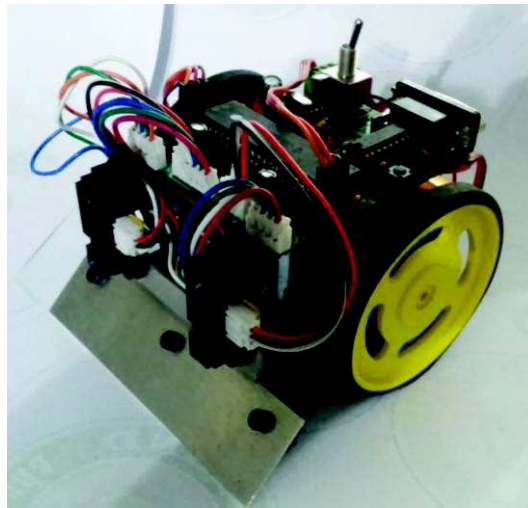
*TD Robótica* [en línea]. <http://tdrobotica.co/> [Consulta: 3.9.15].

*Como crear un robot de sumo autónomo y barato* [en línea].  
<http://es.slideshare.net/di3go10/como-crear-un-robot-de-sumo-autnomo-y-barato> [Consulta: 1.9.15].

*Origen y Desarrollo de la Robótica* [en línea].  
<http://proton.ucting.udg.mx/materias/robotica/r166/r63/r63.htm> [Consulta: 10.9.15].

*Diccionari* [en línea]. <http://www.diccionari.cat/> [Consultes Varies].

# ANNEXOS





# **ANNEX 1: PROGRAMA DEL ROBOT**





```
#include <16f877.h>
#fuses XT,NOWDT,NOPROTECT,NOLVP,PUT,BROWNOUT
#use delay(clock=20000000)
#use rs232(baud=9600, xmit=PIN_C6, rcv=PIN_C7)
#define Adelante 0
#define Atras 1
#define Derecha 2
#define Izquierda 3
#define parado 4
#define Derechazo 5
#define Izquierdazo 6
#define GiroAtras 7
```

```
int estado=0x04;
int Servo=0x00;
int dreta=0;
int centre=0;
int esquierda=0;
int dret=0;
int izq=0;
int adret=0;
int aizq=0;
int gira1000 = 0;
int catras = 0;
int x = 0;
```

```
void toma_adc_y_transmite(void){

    // Lectura del canal 5 -> AN5 LM35a
    set_adc_channel(5);
    delay_ms(1);
    dreta=read_adc();
    delay_ms(1);

    // Lectura del canal 6 -> AN6
    set_adc_channel(6);
    delay_ms(1);
    centre=read_adc();
    delay_ms(1);

    // Lectura del canal 7 -> AN7
    set_adc_channel(7);
    delay_ms(1);
    esquierda=read_adc();
    delay_ms(1);
}
```

```
void toma_distancia(void){

    // Lectura del canal 2 -> AN2
```



```
set_adc_channel(2);
delay_ms(1);
dret=read_adc();
delay_ms(1);

// Lectura del canal 3 -> AN3
set_adc_channel(3);
delay_ms(1);
izq = read_adc();
delay_ms(1);
if (aizq < 80 && izq > 80) estado = Izquierda; //se va por la izquierda
if (adret < 80 && dret > 80) estado = Derecha; //se va por la derecha
if(izq > dret && izq < 80 && dret < 80) estado = Derecha; //esta por la
derecha
else estado = Adelante; //esta delante
if(izq < dret && izq < 80 && dret < 80) estado = Izquierda; //esta por la
izquierda
else estado = Adelante; //esta delante
if(izq == dret) estado = Adelante; //esta delante
adret = dret; //guardo el estado actual
aizq = izq; //guardo el estado actual
}

#int_TIMER1
void timer1_isr(){

// Modulaci?n

Switch(Estado){

Case Adelante:
//printf("adelante\n");
Switch(Servo){

Case 0: output_high(PIN_B2);
output_high(PIN_B1);
set_timer1(65536-313); //0,5
break;

Case 1: output_low(PIN_B2);
//output_high(PIN_B2);
set_timer1(65536-1250); //1,5
break;

Case 2: //output_low(PIN_B1);
output_low(PIN_B1);
set_timer1(65536-10937); //18
break;
}
break;
```



Case Atras:

```
++catras;
//printf("atras\n");
Switch(Servo){

    Case 0: output_high(PIN_B1);
            output_high(PIN_B2);
            set_timer1(65536-313); //0,5
            break;

    Case 1: output_low(PIN_B1);
            //output_high(PIN_B1);
            set_timer1(65536-1250); //1,5
            break;

    Case 2: //output_low(PIN_B2);
            output_low(PIN_B2);
            set_timer1(65536-10937); //18
            break;
}break;
```

Case Derecha:

```
//printf("derecha\n");
Switch(Servo){

    Case 0: output_high(PIN_B2);
            output_low(PIN_B1);
            set_timer1(65536-313); //0,5
            break;

    Case 1: output_high(PIN_B2);
            output_low(PIN_B1);
            set_timer1(65536-313); //0,5
            break;

    Case 2: output_low(PIN_B2);
            output_low(PIN_B1);
            set_timer1(65536-11562); //19
            break;
}break;
```

Case Izquierda:

```
/*printf("izquierda\n");*/
Switch(Servo){

    Case 0: output_high(PIN_B1);
            output_low(PIN_B2);
            set_timer1(65536-313); //0,5
            break;
```





```
Case 1: output_high(PIN_B1);
      output_low(PIN_B2);
      set_timer1(65536-800); //1,28
      break;

Case 2: output_low(PIN_B1);
      output_low(PIN_B2);
      set_timer1(65536-11388); //18,22
      break;
}break;
Case parado:
/*printf("parado\n");*/
Switch(Servo){

Case 0: output_low(PIN_B1);
      output_low(PIN_B2);
      set_timer1(65536-313); //0.5
      break;

Case 1: output_low(PIN_B1);
      output_low(PIN_B2);
      set_timer1(65536-625); //1
      break;

Case 2: output_low(PIN_B1);
      output_low(PIN_B2);
      set_timer1(65536-11562); //18,5
      break;
}break;
Case Derechazo:
// ++gira1000;
//printf("derecha\n");
Switch(Servo){

Case 0: output_low(PIN_B2);
      output_low(PIN_B1);
      set_timer1(65536-313); //0,5
      break;

Case 1: output_high(PIN_B1);
      output_high(PIN_B2);
      set_timer1(65536-800); //1,28
      break;

Case 2: output_low(PIN_B2);
      output_low(PIN_B1);
      set_timer1(65536-11562); //18,22
      break;
}break;
```



```
Case Izquierdazo:
  // ++gira1000;
/*printf("izquierda\n");*/
Switch(Servo){

Case 0: output_high(PIN_B1);
        output_high(PIN_B2);
        set_timer1(65536-313); //0,5
        break;

Case 1: output_high(PIN_B1);
        output_high(PIN_B2);
        set_timer1(65536-700); //1,12
        break;

Case 2: output_low(PIN_B1);
        output_low(PIN_B2);
        set_timer1(65536-11562); //18,38
        break;
}break;
Case GiroAtras:
  ++gira1000;
//printf("derecha\n");
Switch(Servo){

Case 0: output_high(PIN_B2);
        output_high(PIN_B1);
        set_timer1(65536-313); //0,5
        break;

Case 1: output_high(PIN_B2);
        output_low(PIN_B1);
        set_timer1(65536-625); //1
        break;

Case 2: output_low(PIN_B2);
        output_low(PIN_B1);
        set_timer1(65536-11562); //18,5
        break;
}break;
}

if(++Servo>2){
  Servo=0;
} //SERVO=1-SERVO;}

}

void main() {
```



```
set_tris_c(0xff);
set_tris_d(0x00);
output_bit(PIN_D7,1);
printf("\r\n\ La x: %u \r\n",x);

setup_adc(ADC_CLOCK_INTERNAL);
setup_adc_ports(ALL_ANALOG);
SETUP_TIMER_1(T1_INTERNAL|T1_DIV_BY_8);
enable_interrupts(INT_TIMER1);
enable_interrupts(global);
SET_TRIS_E (0b00000111);
set_tris_b(0x00);
estado = Adelante;
do {
    toma_adc_y_transmite();
    if ((dreta<100)||((centre < 100)||((esquerra < 100))){
        estado=Atras;
    }
    if(estado == Atras && catras < 50);
    else{
        if(estado == Atras) estado = GiroAtras;
        if(estado == GiroAtras && gira1000 < 50);
        else{
            gira1000 = 0;
            catras = 0;
            toma_distancia();
            if ((izq < 70 && dret < 70) || (izq >= 70 && dret >= 70)) {
                estado = Adelante;
                printf("\r\n Endavant \r\n");
            }
            else if(izq > 70 ) {
                estado=lzquierda;
                printf("\r\n Esquerra \r\n");
            }
            else if(dret > 70){
                estado=Derecha;
                printf("\r\n Dreta \r\n");
            }
            else {
                estado = Adelante;
                printf("\r\n Endavant \r\n");
            }
        }
    }
} while (TRUE);
}
```



## **ANNEX 2: PRESSUPOST**





## 1.- Costos d'enginyeria

### 1.1.- Càlcul del preu hora

Amb l'objectiu de realitzar el càlcul aproximat del preu/hora, suposarem que el projecte ha estat realitzat per un únic enginyer. Per realitzar aquest càlcul s'han de tenir en compte els costos directes i els indirectes.

#### 1.1.1.- Costos directes

Els costos directes es poden definir com la suma dels costos de material, mà d'obra i equip necessari per a la realització d'un producte .

|                               |             |
|-------------------------------|-------------|
| Nombre d'enginyers            | 1           |
| Salari anual enginyer         | 48.000 €    |
| Nº d'hores anuals treballades | 1700 h      |
| Hores venudes                 | 60%         |
| Nº hores venudes              | 1020h / any |
| <b>Preu / hora</b>            | <b>48 €</b> |

*Figura 2. 1.- Variables a tenir en compte per al càlcul dels costos directes*

Per a la realització d'aquest treball s'ha tingut en compte que únicament el personal es componia d'un enginyer amb un salari anual de 48.000 €. A partir de les dades observables en la taula es pot veure que el preu / hora del enginyer es de 48 €.

#### 1.1.2.- Costos indirectes

Els costos indirectes són els que s'extreuen de la realització d'un procés constructiu del qual deriva un producte. No s'ha de tenir en compte els costos explicats en l'apartat anterior.

|                                |                 |
|--------------------------------|-----------------|
| Cost anual lloguer del local   | 4.200 €         |
| Equips d'instrumentació        | 2.500 €         |
| Equips informàtics             | 1.200 €         |
| Material d'oficina             | 1.500 €         |
| Despeses oficina               | 1.250 €         |
| <b>Total costos indirectes</b> | <b>10.650 €</b> |

*Figura 2. 2.- Variables a tenir en compte per al càlcul dels costos indirectes.*



De la Figura 2. 2, es pot extreure que el preu / hora dels costos indirectes és de 6'50 € aproximadament.

### 1.1.3.- Preu total

La suma dels costos directes i indirectes és el preu total. Aquest valor en el projecte és de 54'5 €.

### 1.2.- Costos derivats del disseny

A continuació trobem la Figura 2. 3, on es pot observar el preu i la quantitat dels material utilitzats en el Treball de Recerca.

| NOM                                     | Preu unitat | Quantitat | Preu Final     |
|---|-------------|-----------|----------------|
| Condensador ceràmic multicapa 0'1uF 50V | 0,11 €      | 11        | 1'24 €         |
| Condensador ceràmic DS tipus lletia     | 0'05 €      | 2         | 0'1 €          |
| Condensador electrolític 33uF 25V       | 0'25 €      | 2         | 0'5 €          |
| Condensador electrolític 100uF 25V      | 0'2 €       | 2         | 0'4 €          |
| Diode 1N4148                            | 0'1 €       | 1         | 0'1 €          |
| LED Red & Green T-100                   | 0'2 €       | 2         | 0'4 €          |
| 40 pin 100mil pitch dual-row header     | 1'7 €       | 1         | 1'7 €          |
| 100 mil pitch 3-pin header              | 0'32 €      | 2         | 0'64 €         |
| 2.0mm pitch 3-pin shrouded header       | 0'1 €       | 2         | 0'2 €          |
| DB9 Femella PCB                         | 1'85 €      | 1         | 1'85 €         |
| Socket 40 pin per microcontrolador      | 0'64 €      | 1         | 0'64 €         |
| DPDT Slide Switch                       | 4 €         | 1         | 4 €            |
| 6.0mm Tact Switch                       | 0'15 €      | 1         | 0'15 €         |
| Terminal Block 3 positions 3.5mm pitch  | 0'20 €      | 1         | 0'20 €         |
| PIC16F877A DIP 40                       | 10'18 €     | 1         | 10'18 €        |
| MAX667                                  | 12'58 €     | 1         | 12'58 €        |
| DS232A                                  | 3 €         | 1         | 3 €            |
| 20.00MHz Resonator w/caps               | 1'15 €      | 1         | 1'15 €         |
| Estructura robot                        | 40 €        | 1         | 40 €           |
| <b>TOTAL</b>                            |             |           | <b>89'33 €</b> |

*Figura 2. 3.- Materials emprats en la fabricació del Treball de Recerca.*

### 1.3.- Costos derivats de l'elaboració de la memòria

El total de nombre d'hores emprades per realitzar la memòria del treball és de 98 hores. Com que el projecte a requerit un total de 139 hores, el preu total s'obtindrà multiplicant el nombre d'hores emprades pel preu/hora. Aquest preu és de 6.672 €.



#### 1.4.- Cost total enginyeria

El cost total del projecte s'obté sumant els costos de disseny i els costos de la elaboració de la memòria del treball. Com es pot veure a la figura 1. 4, el cost total de l'enginyeria és de 3.052 €.

| Costos             | Hores | Preu / hora | Cost total     |
|--------------------|-------|-------------|----------------|
| Disseny            | 6     | 54'5 €      | 327 €          |
| Elaboració memòria | 50    | 54'5 €      | 2.725 €        |
| <b>TOTAL</b>       |       |             | <b>3.052 €</b> |

*Figura 2. 4.- Càlcul del cost total de l'enginyeria*



## 2.- Costos derivats del muntatge

### 2.1.- Càlcul del preu hora

Per fer el càlcul del preu hora suposarem que el projecte ha estat realitzat per un operari qualificat, treballant d'autònom.

Com passava amb els costos de l'enginyeria, aquests costos es divideixen en costos directes i indirectes.

#### 2.1.1.- Costos directes

Els costos directes són els relacionats amb el salari del nostre treballador autònom. Suposarem que l'operari té un sou de 20 €/h.

#### 2.1.2.- Costos indirectes

Els costos indirectes són les despeses generals: lloguer del local, màquines, eines i materials del taller.

|                     |                |
|---------------------|----------------|
| Lloguer del local   | 4.200 €        |
| Màquines i eines    | 1.000 €        |
| Material del taller | 750 €          |
| Despeses generals   | 800 €          |
| <b>TOTAL</b>        | <b>6.750 €</b> |

*Figura 2. 5.- Variable a tenir en compte per al càlcul dels costos indirectes.*

Els costos derivats del muntatge ascendeixen a uns 6.750 €. Per a calcular el preu/hora que suposen els mateixos es consideren 1.800 h de treball i aplicant el coeficient del 65%, el cost aproximat és de 6 €/h.

#### 2.1.3.- Costos totals derivats del muntatge

Sumant els costos directes i els indirectes el preu total del muntatge és de 26 €.

## 2.2.- Cost total del muntatge del robot

A la Figura 2. 6, es poden observar les hores emprades en la construcció del robot.





| Tasques realitzades         | Duració (hores) |
|-----------------------------|-----------------|
| Compra materials            | 1               |
| Disseny de l'estructura     | 4               |
| Construcció de l'estructura | 5               |
| Implementació               | 1               |
| Programació                 | 20              |
| <b>TOTAL</b>                | <b>41</b>       |

*Figura 2. 6.- Variables a tenir el compte en el temps emprat en la creació del robot*

A partir d'aquestes dades es conclou que per realitzar el muntatge de la maqueta són necessàries 41 h. El cost del muntatge de la maqueta ascendeix a 1.066 €.



**CREACIÓ, IMPLEMENTACIÓ I PROGRAMACIÓ D'UN ROBOT LLUITADOR DE SUMO**

---