Treball de Recerca

Construcció d'un robot

03/10/2012

Institut de Celrà

Índex

1. Introducció

Aquest treball potser ha estat el més llarg i complicat que hagi fet mai, explicar el seu funcionament a la perfecció no és senzill perquè fer-lo tampoc ho ha estat. Hi ha hagut molts canvis, problemes i dificultats que m'han impedit fer-lo ràpidament que és el que m'hauria agradat.

El motiu i les ganes que m'han animat a escollir un treball de robòtica ve ja des de 4t d'ESO on el professor de tecnologia que tenia ens va ensenyar una mica com funcionava tot el tema de microcontroladors i electrònica.

Aquell mateix any em vaig assabentar per les proves Cangur de matemàtiques que a la Universitat Politècnica de Catalunya feien un curs de programació en C++ en el qual em podia apuntar i no vaig deixar escapar l'ocasió.

A part, a començaments del 1r curs de Batxillerat, a la classe de física el professor ens va proposar una pràctica per treballar el tir parabòlic, que en aquell moment estàvem estudiant a classe. La pràctica consistia en llançar una esfera de metall amb una rampa i, a partir de la velocitat que portava, ser capaços de calcular on cauria i marcar el punt a terra.

Aquell tema em va interessar i el treball el vam intentar fer tant complet com pogués ser, fins i tot vam fer un vídeo explicant el nostre treball. (Es pot trobar a la següent adreça: http://youtu.be/Oim1Yjc7VsU)

A partir del moment en el que vam acabar el treball, vaig pensar que tot el procés, que havia durat uns 30 minuts, en el que vam estar calculant on cauria, ho podria fer una màquina en menys d'1 segon, i que aquesta seria la meva motivació per fer el treball de recerca, la de fer una màquina capaç de calcular on cauria una esfera que es llençava des d'una rampa.

Com que jo volia fer un treball que estigués relacionat amb un dels tres temes anteriors (electrònica, programació i física) la idea de fer un robot que utilitzés la física per saber on aniria a parar una esfera em va semblar interessant.

El títol “Construcció d’un robot” no ens diu molt d’aquest treball ja que de robots n’hi ha de moltes formes i mides, el robot que em proposo fer en aquest treball no s’assembla gens a un robot de forma humana (que és la primera definició de robot que li passa a qualsevol pel cap) ja que un robot pot ser qualsevol màquina que pot realitzar una sèrie de moviments automàticament.

En el projecte he canviat la hipòtesi per objectius, ja que és un treball pràctic i no té sentit fer-se una hipòtesi, així els objectius plantejats són:

- Veure si sóc capaç de dissenyar i construir un robot que marqui el punt de caiguda de l’esfera tal i com he dit anteriorment.
- Aprendre a programar microcontroladors.
- Aprendre el funcionament d’un microcontrolador.
- Conèixer més sobre mecànica per poder fer un sistema eficient.

2. Fonament teòric

2. 1. Procés tecnològic

Aquest és un treball bàsicament pràctic i que posa en pràctica el procés tecnològic per poder dur a terme tot el procés de construcció.

Aquest procés és un mètode de treball que consta d'un conjunt d'etapes que faciliten l'obtenció dels resultats i que aquests siguin adequats als objectius inicials. Les fases són les següents:

1. **Requeriment:** és el començament del procés tecnològic i consisteix en identificar el problema que volem solucionar.
2. **Recerca d'informació:** es tracta de buscar quines solucions s'han donat fins al moment i quines solucions hi ha al mercat per a problemes similars. També s'investiga sobre les possibilitats i el cost de realitzar-ho.
3. **Generació d'idees:** a partir de la informació s'han de plantejar possibles solucions avaluant els pros i els contres de cada idea.
4. **Selecció de la millor solució:** s'han d'analitzar els diversos factors (cost, possibilitats de realització, equipament disponible, originalitat, estètica, impacte ambiental...) que permetin escollir la solució més eficient.
5. **Planificació de la realització:** en aquesta etapa s'ha d'organitzar tot el treball que s'ha de fer. S'han de preveure tots els procediments, materials, eines i maquinari que s'ha d'utilitzar.
6. **Realització:** s'han de realitzar tots els passos planificats prenent notes sobre el que es va fent i les incidències que es vagin produint.
7. **Avaluació:** un cop acabada la realització es valora si el resultat soluciona correctament els objectius inicials. Si no es compleixen els resultats esperats, es torna al principi per intentar solucionar el problema.

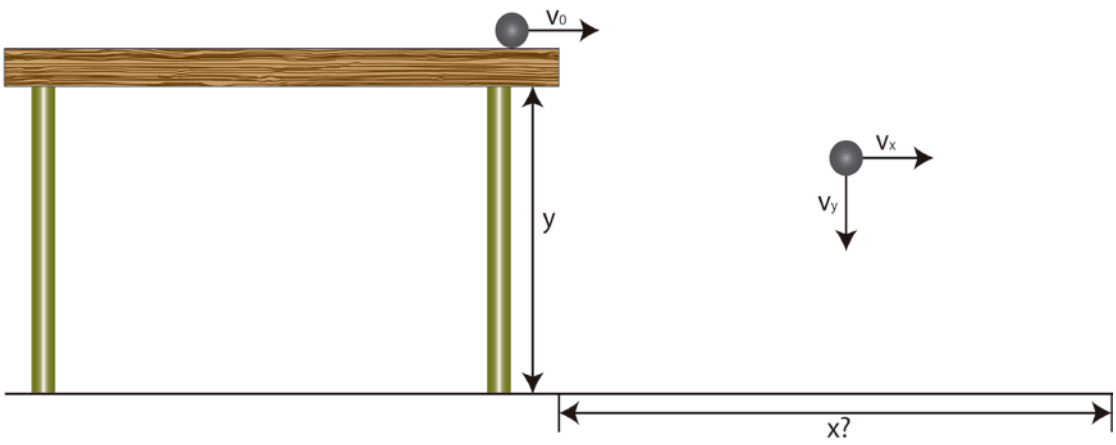
Il·lustració esquemàtica del procés tecnològic

Tot aquest procés es va repetint les vegades necessàries fins que s'assoleixin tots els objectius, d'aquesta manera qualsevol projecte pot estar sotmès a una millora constant.

2.2. Tir parabòlic

Com que l'objectiu del treball és que un robot reculli tot sol una esfera massissa que cau a una certa velocitat d'una taula, primer hem de saber com calcular el punt exacte de la caiguda.

En aquest cas l'esfera només es mourà en dos dimensions (X i Y). Així doncs, per calcular on caurà un objecte que va per sobre una superfície és necessari dividir el moviment en dos vectors. Un és l'horitzontal, on no hi intervé cap força de manera que no hi ha acceleració (Moviment Rectilini Uniforme), i l'altre és el vertical, on sí que hi intervé una força que és la gravetat i que estira el mòbil; per tant, hi ha acceleració (Moviment Rectilini Uniformement Accelerat). Les dades que tindrem per calcular-ho seran l'alçada de la superfície al terra (y) i la velocitat de l'objecte (v_0).



Il·lustració esquemàtica del tir parabòlic

Les equacions de velocitat que surten per al eix X i Y són les següents:

$$v_x = \frac{x}{t} \rightarrow MRU$$

$$v_y = v_0 - at \rightarrow MRUA$$

Les equacions de distància dels dos eixos són les següents:

$$x = x_0 + v_x \cdot t \rightarrow MRU\ (Eix\ X)$$

$$y = y_0 + v_{y0} \cdot t + \frac{1}{2} a \cdot t^2 \rightarrow MRUA\ (Eix\ Y)$$

Per trobar la distància de caiguda (*x*) necessitem trobar el temps (*t*) i aquest ve donat per la distància que hi ha de la taula al punt de caiguda (y_0) de manera que ens centrarem en l'equació de distància de l'eix vertical (Y). L'acceleració (*a*) és la gravetat (que a la Terra val -9,8 m/s^2) i com que quan comença a caure no té velocitat en l'eix vertical (v_x), aquesta és zero. La distància final (y) també és zero ja que l'objecte cau fins al terra i aquest és considerat com una altura de 0 metres. Substituïm els valors que ja coneixem i ens surt la següent equació on ens queden com a incògnites l'alçada de la caiguda i el temps.

$$0 = y_0 + 0 \cdot t + \frac{1}{2} \cdot (-9{,}8) \cdot t^2$$

El que busquem és el temps, per tant, l'aïllem:

$$\frac{y_0}{\left(\frac{9{,}8}{2}\right)} = t^2$$

$$\sqrt{\frac{\boldsymbol{y_0}}{\boldsymbol{4{,}9}}} = \boldsymbol{t}$$

Aquesta serà la fórmula que utilitzarem per saber quan triga a arribar un objecte a terra. Tot i així el que estem buscant és la distància d'arribada respecte el lloc on comença la caiguda i, per tant, encara no hem acabat.

Així doncs, el que farem és utilitzar l'equació de l'eix de les X per trobar la distància de caiguda. Aquí la distància inicial a l'eix de les X (x_0) és nul·la i l'equació ens quedarà únicament en funció de la velocitat que portava l'objecte abans de començar a caure i del temps que hem trobat que trigarà a caure.

$$\boldsymbol{x = v_x \cdot t}$$

Com que el temps ens ve donat per l'alçada que hi ha des del punt de caiguda fins al terra, el substituïm i generem una sola equació que ens permet calcular la distància de caiguda a l'eix X només amb l'alçada que hi ha de caiguda i amb la velocitat que porta el mòbil.

$$\boldsymbol{x = v_x \cdot \sqrt{\frac{y_0}{4,9}}}$$

Aquest serà la fórmula a utilitzar per calcular la distància de caiguda d'un objecte, a la Terra, en funció de la velocitat que porta abans de caure i de l'altura des del punt de caiguda al terra.

Per fer tots aquests càlculs partim del supòsit que, tot i no tenir en compte el fregament amb l'aire i altres pertorbacions que es puguin donar (ja que no fem l'experiment en el buit), no afectaran al resultat perquè el marge d'error serà insignificant.

2.3. Què és un PIC?

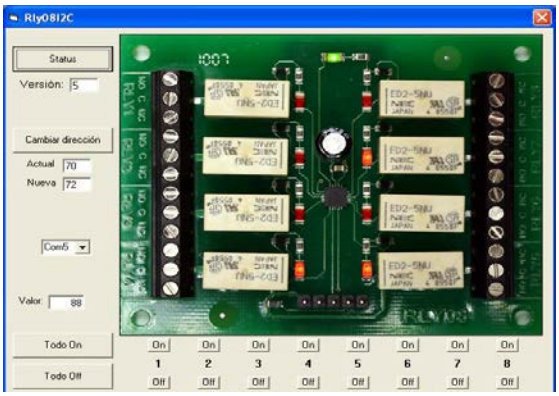
Com que el que volem és agafar una esfera al moment de llençar-la, l'apartat anterior l'haurem de fer a una velocitat molt alta. Per fer això i que, a més a més, sigui tot automàtic necessitarem una eina que faci de "cervell". Així doncs, haurà de realitzar els càlculs el més ràpid possible i alhora haurà de ser capaç de controlar un mecanisme que pugui moure's i aconseguir agafar l'esfera. Aquest controlador també haurà de ser el més econòmic possible.

Per fer tot aquest tipus de control bàsicament hi ha tres possibilitats:

- Un PLC: de l'anglès *Programable Logic Controller*, és un autòmat programable industrial dissenyat per a controlar diferents màquines, és utilitzat a les indústries ja que té entrades i sortides amb una gran potència. Aquesta seria una eina molt pràctica per realitzar el projecte, l'únic inconvenient és el seu elevat preu.

Mòdul d'un PLC

- Un ordinador: es pot utilitzar a través d'USB connectant-hi una placa electrònica tot i que aquest sistema és bastant complicat de fer ja que a l'hora de programar-lo requereix un coneixement molt elevat dels llenguatges de programació i electrònica.



Control d'un circuit amb Visual Basic

- Un microcontrolador: és l'equivalent a un ordinador en miniatura ja que, en general, té totes les característiques d'un ordinador però sense tanta potència. Cada model té unes característiques determinades i cal saber molt bé què és el que es vol fer amb ell perquè podria no tenir les característiques que podríem necessitar. Podem destinar-lo al control de petites pantalles LCD, motors pas a pas, LED o teclats numèrics. Els més comuns són els PIC que desenvolupa l'empresa *Microchip Technology Inc.*

Imatge del PIC 16F628A

Jo m'he decantat per la utilització d'un microcontrolador i en concret les dos opcions entre les que he estat dubtant per escollir el xip que volia utilitzar eren els que desenvolupa l'empresa *PICAXE* o els de *Microchip*.

La primera té un gran avantatge ja que aquesta empresa ven una gran quantitat de productes que estan dissenyats per funcionar a la perfecció amb tots els components que ells venen. El problema és que s'ha d'utilitzar és que s'ha d'utilitzar el llenguatge de programació que han dissenyat ells i amb aquest és bastant complicat realitzar el càlcul que es necessita per poder fer aquest projecte. També no hi ha tanta informació perquè no és el que tendeix a utilitzar la gent normalment. S'acostumen a utilitzar als centres escolars a l'assignatura de tecnologia per fer una iniciació a l'electrònica programable.

Microchip, en canvi, només desenvolupa microcontroladors i aquests són dels més utilitzats del mercat, cosa que ajuda a trobar informació més fàcilment. A més a més, hi ha una gran quantitat de llenguatges de programació.

D'entre aquestes dues opcions jo m'he decantat per utilitzar un microcontrolador de l'empresa *Microchip* ja que m'ha semblat molt més pràctic en quant a trobar informació.

D'aquests hi ha bàsicament tres grups que bàsicament són els més utilitzats:

- Els PIC12C508 i PIC12C509 on el programa va gravat en una memòria EPROM, que només es pot esborrar amb llum ultraviolada. Disposen d'un oscil·lador intern i tenen un nombre reduït de potes per això són molt utilitzats per dissenyar petits circuits.
- El PIC16F84A, té una memòria de programa de tipus FLASH (semblant al dels llapis USB), necessita un oscil·lador extern per poder treballar i te 13 pins de Entrada i Sortida. Amb el temps ha resultat ser un dels més populars de tots.
- La família PIC16F8X, amb un gran nombre de millores respecte el PIC16F84A ja que té un número de d'entrades i sortides superior i una major capacitat per a la memòria de programa i la memòria de dades.

2.3.1 Components bàsics dels PIC

Tots els PIC disposen d'uns components bàsics per al seu funcionament i són les següents:

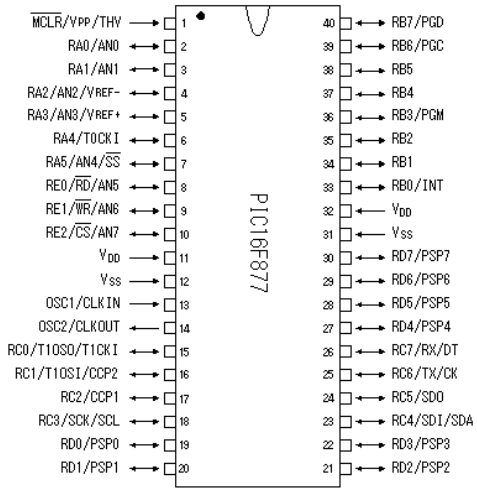
- **La CPU:** s'encarrega d'executar les instruccions que estan a la memòria del programa. També es sol anomenar microprocessador.
  Si el comparéssim amb una persona, seria el cervell. Per tant aquest, que executa unes instruccions, necessita un lloc on guardar-les, la memòria, i un medi amb el que interactuar amb l'exterior, els dispositius d'Entrada/Sortida.
- **La memòria:** són tots els components del microcontrolador que s'encarreguen de guardar informació durant un temps. Bàsicament hi ha dos tipus: la memòria de programa, on es guarden totes les instruccions del programa, i la memòria de dades, on es guarden totes les dades que utilitzarem per a la execució del programa.
  Dintre de la memòria de dades hi ha la volàtil i la no volàtil. La primera s'esborra en el moment en que es desconnecta l'alimentació (memòria RAM) i la segona només s'esborra aplicant-hi una carrega elèctrica més alta de la normal (memòria EEPROM).
- **Els dispositius d'Entrada/Sortida:** són els sistemes que utilitza el microcontrolador per comunicar-se amb l'exterior. Així, els d'entrada ens permeten enviar informació al controlador i els de sortida, perquè aquest la tregui a l'exterior.

2.3.2 El PIC utilitzat en el projecte

Per a fer el treball he escollit el PIC16F877 que pertany a la família del PIC16F8X. Les seves característiques són les següents:

- Microcontrolador de 8 bits
- Freqüència màxima de funcionament de fins a 20MHz
- Memòria del programa de fins a 8000 paraules
- Memòria RAM de 368 bytes
- Memòria EEPROM de 256 bytes
- 24 pins d'E/S

PIC16F877



Esquema de les potes del PIC16F877

Aquesta família de PIC necessiten un cristall extern que els generi una freqüència(F) a partir de la qual regulen la seva velocitat ja que necessiten 4 cicles de rellotge per executar una instrucció, a partir de la següent fórmula podem saber el nombre d'instruccions que es poden executar en un segon (n).

$$n = \frac{F}{4}$$

Si utilitzem un cristall de 20 MHz:

$$\frac{20.000.000\,Hz}{4} = \mathbf{5.000.000\ instruccions/s}$$

2.4. Dispositius de sortida

2.4.1 Motor

2.4.1.1. Components i funcionament d'un motor

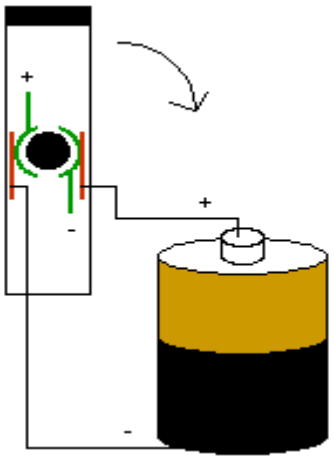
Apart d'un controlador també serà necessari un motor perquè actuï i ens reculli l'esfera en moviment. Però, primer, hem de mirar com funciona un motor, quines característiques té i quines seran les que necessitarem per poder recollir una esfera amb prou rapidesa perquè aquesta no ens caigui al terra.

El secret que hi ha darrere de qualsevol motor és el magnetisme. Si en un imant intentem unir dos pols similars (Nord-Nord o Sud-Sud) aquests es repel·leixen. Un motor no és més que un conjunt d'imants als quals se'ls va canviant la polaritat per tal que facin girar un eix central.

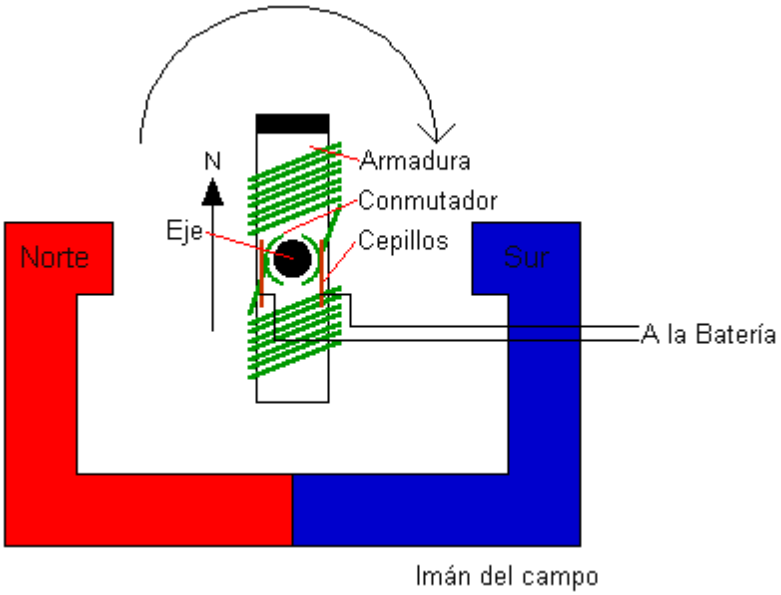
Per a això s'aprofiten les propietats dels electroimants. Un electroimant consisteix en una bobina de cable enrotllada al voltant d'un nucli ferrós. En aplicar corrent a aquesta bobina això genera un camp magnètic orientat segons la polaritat de connexió a la bobina. Si bescanviem el positiu i el negatiu bescanviarem també els pols de l'imant generat. Si orientem per tant un electroimant amb els pols d'un imant permanent i en bescanviem de cop les connexions passarà a repel·lir-se amb l'imant permanent, de manera que si el tenim fixat sobre un eix es veurà obligat a girar.

Aquest és precisament el principi de funcionament d'un motor elèctric. Per a això l'estructura del motor consta d'un imant permanent estàtic que s'anomena estator, un rotor o armadura que consisteix en un electroimant que es fa enrotllant dos o més pols al voltant d'un centre metàl·lic. Aquests bobinats es denominen debanats. L'armadura té un eix i un commutador fixat a l'eix. Per a que el sistema giri cal completar-lo amb les denominades "escombretes" que s'encarreguen del canvi de camp elèctric.

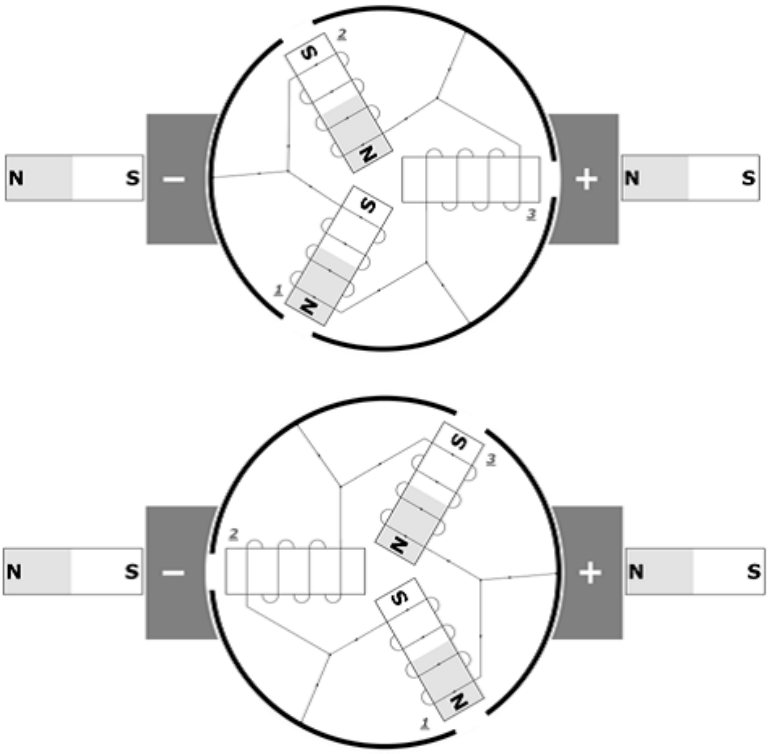
Al diagrama de la dreta es mostra com el commutador i les escombretes treballen junts per a deixar que l'actual flux d'electrons vagi a l'electroimant i que canviïn la direcció del flux d'electrons tan bon punt aquest giri, ja que els contactes del commutador estan fixats a l'electroimant i giren amb aquest.



El conjunt complert es pot veure a la imatge inferior:



El nombre de pols pot ser superior a dos per a que el motor vagi més suau, com a les dues imatges següents on es pot veure com es connecten o desconnecten els diferents pols d'un motor per tal que aquest giri.



Imatges de la polarització d'un motor al moure's

Per poder saber quines característiques té un motor hi ha dos dades que són importants, el parell i les revolucions per minut (rpm).

El parell(P) ens serveix per saber la força que té un motor a una certa distància es sol donar en "N·m" (Newtons x metre) i és la força multiplicada per la distància.

Les revolucions per minut són les voltes que un motor és capaç de donar en un minut i ens indiquen la velocitat a la que és capaç de treballar un motor.

Per trobar un motor que pugui moure 0,5 Kg amb un pinyó de 7mm de radi amb una velocitat prou alta per poder agafar la bola fem el següent càlcul:

$$P = F \cdot x$$

$$P = 4{,}9 \cdot 0{,}007 = \mathbf{0{,}0343\ N \cdot m}$$

Per trobar la velocitat calcularem que com a màxim s'hagi de moure el sistema de recollida a una distància de 0,5m abans que caigui la bola des d'una altura d'uns 65cm. Utilitzarem les fórmules esmentades prèviament a l'apartat 2.2.

$$t = \sqrt{\frac{0{,}65}{4{,}9}} = \mathbf{0{,}39}\boldsymbol{s}$$

$$v = \frac{x}{t}$$

$$v = \frac{0{,}5}{0{,}39} = \mathbf{1{,}28}\ \boldsymbol{m/s}$$

L'eix s'ha de moure a 1,28 m/s per poder arribar a temps, ara buscarem les rpm a les que s'ha de moure l'eix de 0,7 cm de radi per arribar a aquesta velocitat.

$$v = w \cdot r$$

$$(velocitat\ lineal = veloitat\ angular\ \times\ radi)$$

$$1{,}28 = w \cdot 0{,}007$$

$$\frac{1{,}28}{0{,}007} = w = 182{,}86\ \frac{rad}{s} \equiv\ \mathbf{1746{,}16}\ \boldsymbol{rpm}$$

L'eix del pinyó s'haurà de moure amb una velocitat de 1746,16rpm com a mínim i amb un parell mínim de 0,0343 N·m.

2.4.1.2. El motor utilitzat

El motor que he utilitzat jo és un motor que vaig aconseguir treure d'una aspiradora vella, ja que els preus d'un motor de les característiques que necessitava eren bastant elevats i aquest no s'utilitzava per tant el vaig agafar.

El parell del motor a nivell teòric no el sé però experimentalment he pogut comprovar que és el suficient ja que no hi ha hagut problemes a l'hora de fer la força que se li demana.

Les revolucions per minut sí que les he pogut mesurar i dóna 14000 rpm. Com que el motor és massa ràpid l'he hagut de reduir, el factor de reducció ha de ser de com a màxim $\frac{14000}{1746{,}16}\ rpm \sim \frac{8}{1}$; és a dir, 8 voltes del motor per cada volta de l'eix de sortida com a màxim.

Em vaig posar a buscar i vaig trobar uns pinyons que donaven una relació de 6 voltes pel motor per cada volta de l'eix $\left(\frac{6}{1}\right)$ i aquests són els que he utilitzat per fer el muntatge.

Amb aquest muntatge també s'augmenta la força, la fórmula per calcular-ho és

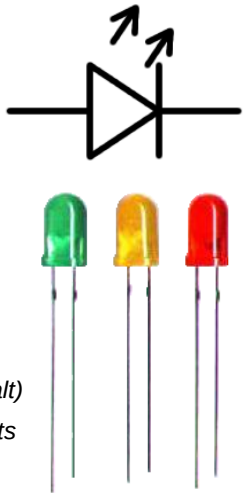
$$F_1 \cdot w_1 = F_2 \cdot w_2$$

$$F_1 \cdot 6 = F_2 \cdot 1$$

Així trobem que la força que té aquest motor (F_1) es veu augmentada per 6 a l'eix de sortida (F_2).

2.4.2. Díodes LED

De l'anglès *Light Emitting Diode* (díode emissor de llum) és un semiconductor emissor de llum. En aquest treball s'utilitza perquè el microcontrolador ens doni informació com que ha acabat de calibrar-se o que ha detectat el pas de l'esfera.



Símbol esquemàtic d'un LED (a dalt) i fotografia de tres LED de diferents colors (a baix)

2.5. Dispositius d'entrada

2.5.1. Codificador rotatiu

Per poder moure el recollidor fins a la posició de caiguda primer és necessari tenir un element que ens indiqui en quin lloc es troba el recollidor.

Aquest element és el codificador rotatiu i n'hi ha de dos tipus, els incrementals i els absoluts.

Els codificadors absoluts, tenen una sèrie de bits (1 i 0) que permeten saber en quin angle l'eix està posicionat. Per treballs de precisió són molt útils. En la figura 1 es pot veure a la dreta, el cercle que s'utilitza per definir la posició, en aquest cas s'utilitzen 5 bits per definir la posició i per tant 32 posicions en una volta ($2^5 = 32$).

Els codificadors incrementals, en canvi, connecten i desconnecten les seves potes quan es mou l'eix. Per fer aquestes connexions acostumen a tenir un sensor de llum i un disc amb forats, quan el sensor detecta un forat, talla el corrent i quan no el detecta, la reactiva, d'aquesta manera es van creant un senyal. A l'esquerra de la figura 1 es veu el disc d'un codificador incremental.

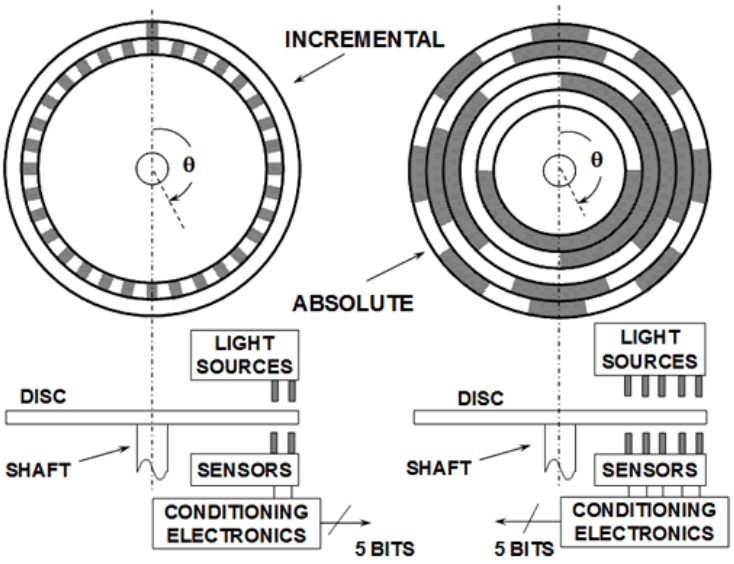


Figura 1

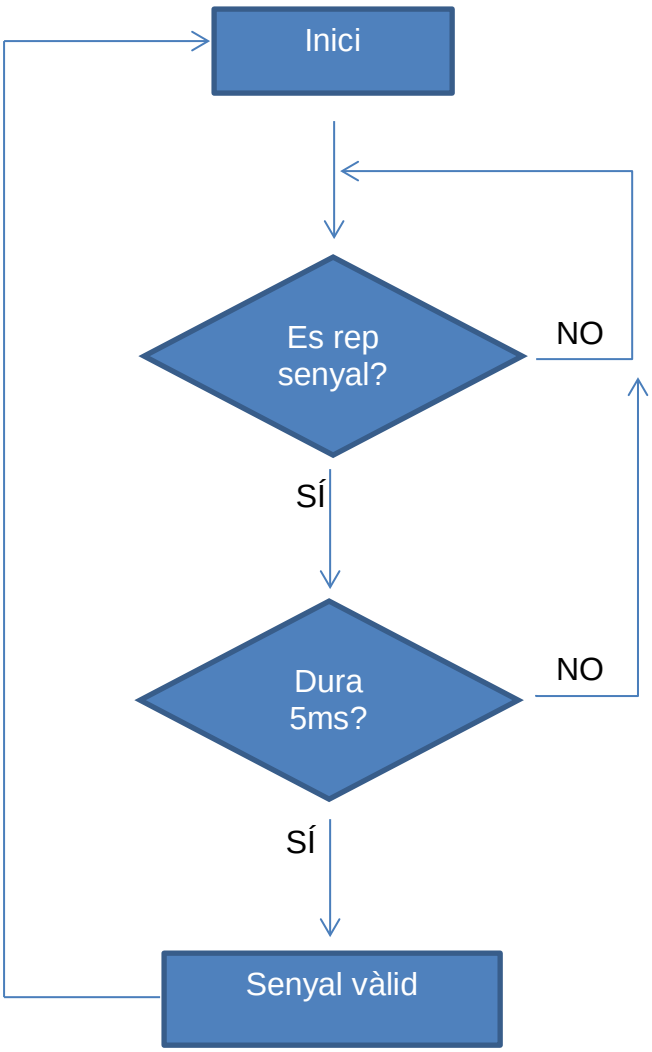
Imatge d'un encoder incremental vist des de fora

El codificador que he utilitzat jo ha estat un incremental perquè era més senzill de detectar la posició a l'hora de fer el programa.

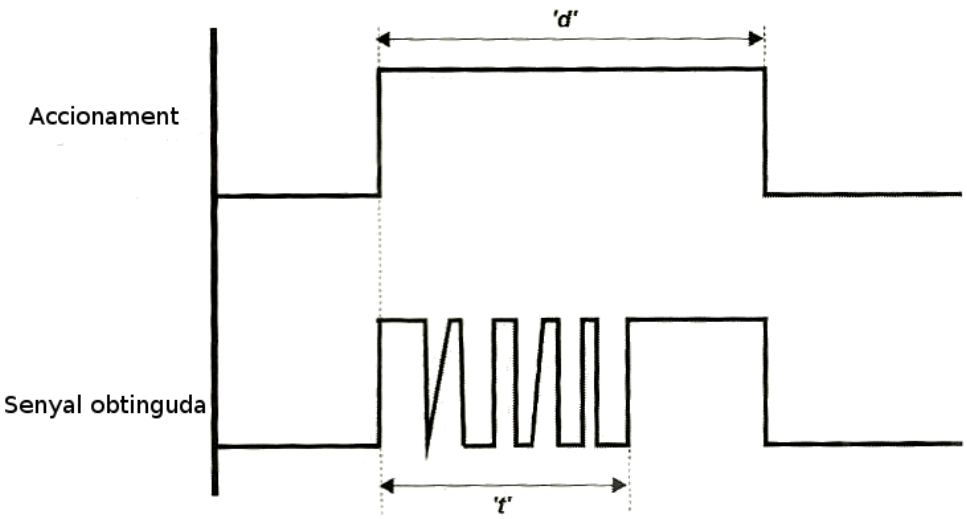
Com que el PIC revisa el programa unes 5.000.000 de vegades per segon s'ha de fer un mètode per no llegir dues vegades la mateixa senyal i donar-la com dues diferents. Per evitar això es fa el següent bucle:

1. Es comprova si hi ha algun senyal i es queda en aquest pas mentre no hi hagi cap senyal provinent del codificador
2. Quan es rep un senyal es comprova que aquesta tingui una durada mínima d'uns 5 milisegons per evitar errors.
3. Quan aquest senyal torna a 0 es dona per vàlida la recepció, si torna a 0 abans de 5 milisegons es descarta.
4. Es torna al principi per detectar un nou senyal.

Això es pot expressar a partir de l'esquema següent:



El fet de comprovar que el senyal que ens dona el codificador dura més de 5 milisegons és per evitar el rebot que es dona en tancar i obrir el circuit.



Sempre que dues plaques metàl·liques es toquen per tal de tancar un circuit aquestes fan un rebot de senyal i, mentre no s'estabilitzen, donen senyals falsos que es poden comptar per error. En el cas de la imatge quan en realitat només s'està donant un únic senyal, per l'efecte del rebot podríem arribar a contar-ne fins a sis i això ens faria perdre moltíssima precisió.

2.5.2. Fotoresistor

LDR

Com a element d'entrada per poder detectar el pas d'una esfera he utilitzat un fotoresistor o LDR (*Light Depending Diode*) que és un component electrònic que varia la seva resistència en funció de la llum.

Tenen una resposta bastant ràpida i permeten així detectar molt bé si tenen algun objecte davant o no. En el treball els he utilitzat com a detectors per activar el cronòmetre.

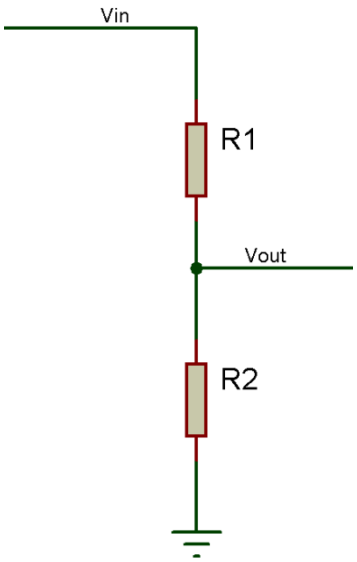
Per provar la velocitat d'aquests components, vaig muntar un rail i vaig posar un LED en un costat i una LDR a l'altre. La LDR estava connectada a un oscil·loscopi que em permetia veure a temps real el voltatge que donava la LDR i aquí vaig veure que tot i que hi haguessin elements que passessin molt ràpid, la velocitat de resposta era també molt ràpida.

2.5.3. Polsadors

Els polsadors són els elements amb els quals podem interactuar amb el PIC. Aquests s'han de connectar amb una resistència per tal de limitar la corrent que pot passar-hi i així no cremar el microcontrolador.

Es connecten amb un divisor de tensió que té el següent esquema i es calcula amb la següent fórmula:

Imatge d'un dels polsadors utilitzats



$$V_{out} = \frac{R_2}{R_1 + R_2} \cdot V_{in}$$

2.6. Algorisme i llenguatge de programació

2.6.1. Algorisme

2.6.1.1. Què és un algorisme?

Un algorisme és un conjunt d'instruccions o passos que serveixen per resoldre un problema o una tasca. S'utilitzen a la vida quotidiana pera resoldre molts problemes com per exemple seguir les instruccions per fer una recepta de cuina.

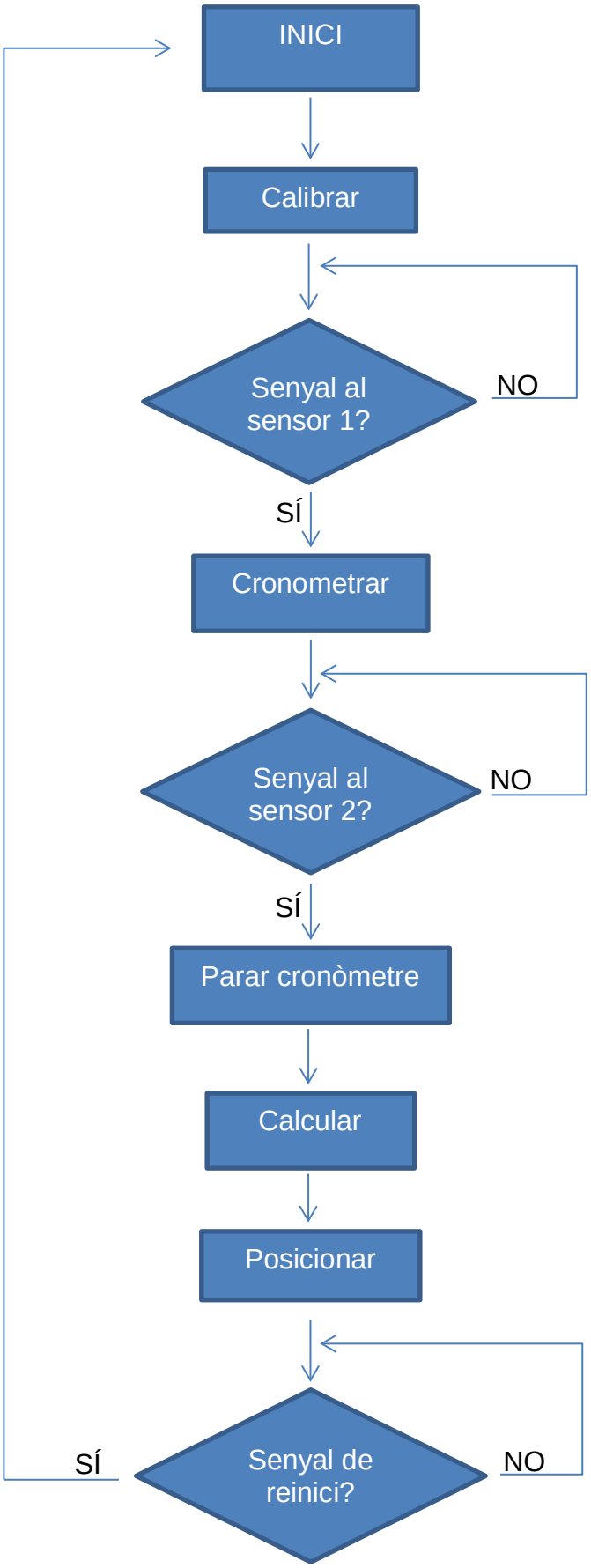
D'aquesta manera sempre que es plantegi el mateix problema, seguint un algorisme per aquell problema, se'n pot trobar la solució.

2.6.1.2. Algorisme utilitzat

Per poder moure el recollidor fins al lloc corresponent:

1. Primer s'engega el robot i aquest mou el recollidor fins al principi i s'atura.
2. El robot posiciona el recollidor al mig per tal de poder anar ràpid a moure's per recollir l'esfera i així recórrer menys distància. Un cop acabat aquest pas s'encenen uns LED per indicar que s'ha acabat de posicionar.
3. Es deixa anar l'esfera per la rampa i passa pel primer detector que engega un cronòmetre.
4. Quan l'esfera passa pel segon detector es para el cronòmetre, s'efectua el càlcul.
5. Es mou el recollidor a la distància de caiguda i s'atura.
6. S'espera a que es premi el botó de reinici, que tornarà a calibrar el robot, per tornar a llançar una altra esfera.

L'algorisme es pot dibuixar de la següent manera, els rectangles indiquen les ordres i els rombes les condicions:



2.6.1. Què és un llenguatge de programació?

Perquè el microcontrolador pugui fer les instruccions de l'apartat anterior necessitem una eina que tradueixi les ordres que nosaltres donem a uns i zeros que és com treballen els microcontroladors. Aquesta és la funció d'un llenguatge de programació.

Un llenguatge de programació és un idioma artificial dissenyat per realitzar processos que pot fer una màquina o un microcontrolador. Cada llenguatge té la seva sintaxi i el seu vocabulari. És com si fos la llengua de comunicació amb les màquines, si es comet un "error ortogràfic" pot donar a lloc a resultats erronis tal i com passaria amb la llengua normal on 'vaca' i 'baca' no són el mateix.

El compilador és la part que interpreta el llenguatge i el converteix en codi binari. És el "traductor" entre nosaltres i el microcontrolador.

Hi ha tres grans llenguatges de programació pels PIC apart del llenguatge màquina (que només són uns i zeros):

- El llenguatge assemblador: és un llenguatge de baix nivell; és a dir, està molt a prop del llenguatge màquina, sense arribar a ser-ho, però aquest tipus està molt orientat a un tipus de màquina en concret i permet un control molt precís del programa.
- El C: és un llenguatge d'alt nivell ja que és més proper al llenguatge humà i amb el que és molt més senzill de treballar. Admet variables de fins a 32 bits (que són 2^{32} possibilitats diferents) i s'hi poden afegir instruccions addicionals a través del sistema de llibreries. Aquestes es defineixen al principi del programa i es poden trobar a molts llocs (pàgines d'internet o compiladors).
- El Basic: també és un llenguatge d'alt nivell, com el C, però aquest té una sintaxi i algunes instruccions que s'anomenen diferent.

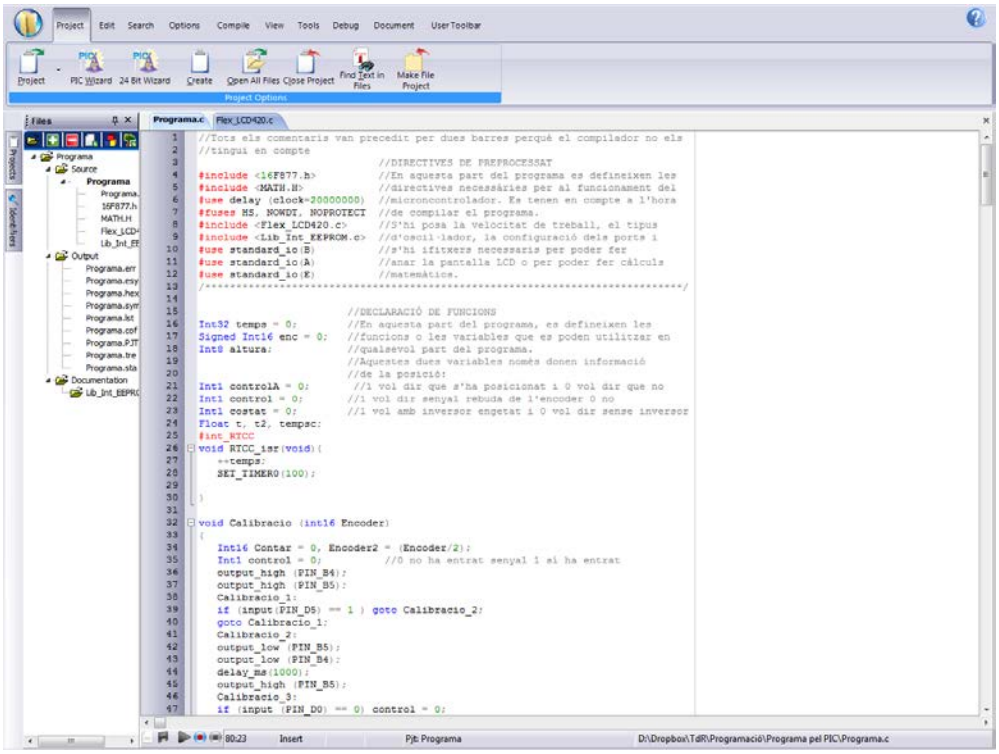
2.6.2. El llenguatge utilitzat

El llenguatge que he escollit jo ha estat el C creat per CCS que és una versió orientada especialment als PIC. Permet una màxima optimització d'aquests dispositius i incorpora una gran quantitat de llibreries per controlar diferents dispositius com pantalles LCD, convertidors d'analògic a digital o rellotges en temps real.

Un programa en C té les següents parts:

- Directrius de preprocessat: controlen la manera en que el compilador converteix el programa a llenguatge màquina
- Programes o funcions: són un conjunt d'instruccions, pot haver-n'hi una o varies però sempre hi ha d'haver la funció principal que es defineix amb '*main()*'.
- Instruccions: indiquen com s'ha de comportar el PIC en tot moment.
- Comentaris: permeten descriure el que significa cada línia de programa, sense afectar al comportament d'aquest.

M'he decidit a escollir aquest tipus de llenguatge perquè, al poder suportar variables més grans, pot fer càlculs molt més grans i amb més precisió.



Imatge del programa utiltizat per fer la programació

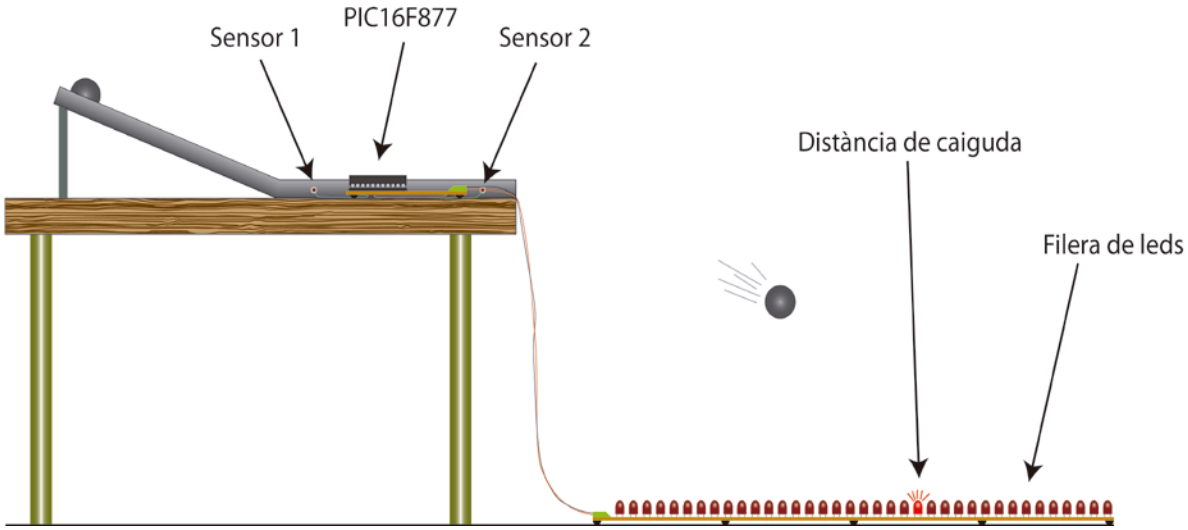
3. Part pràctica

3.1. Disseny

En aquest apartat explicaré el disseny del projecte que jo volia fer al començament i, en funció dels problemes que he tingut, el disseny que he acabat fent.

3.1.1. Idea inicial

Inicialment, vaig pensar en un projecte senzill on només pretenia marcar la distància a la que queia l'esfera, per fer-ho volia posar una filera de LED al terra i encendre el que estigués més a prop del punt de caiguda.

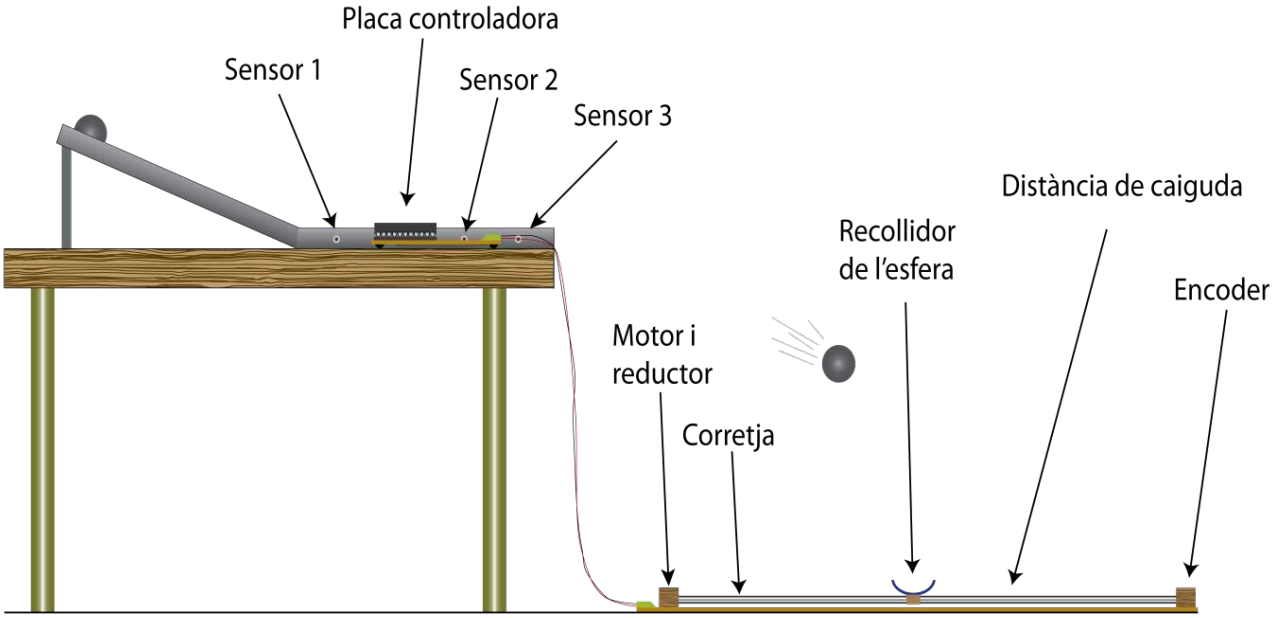


Dibuix del disseny inicial

3.1.2. Disseny final

El disseny final és una mica diferent al que vaig proposar-me al principi però he fet els canvis que he considerat necessaris perquè s'entengui bé la distància de caiguda de l'esfera.

He substituït la filera de LED per una guia motoritzada amb un bol que recull l'esfera. A mitat del disseny se'm va ocórrer posar-hi una petita pantalla LCD per marcar la distància de caiguda però la manca de temps va impedir posar-la en funcionament i provar-la.



Dibuix del disseny final

Apart de fer un dibuix del disseny també en necessitava algunes imatges en detall per poder construir-lo. D'aquesta manera vaig fer una sèrie de plànols i esquemes per poder muntar-ho tot seguint les pautes correctes:

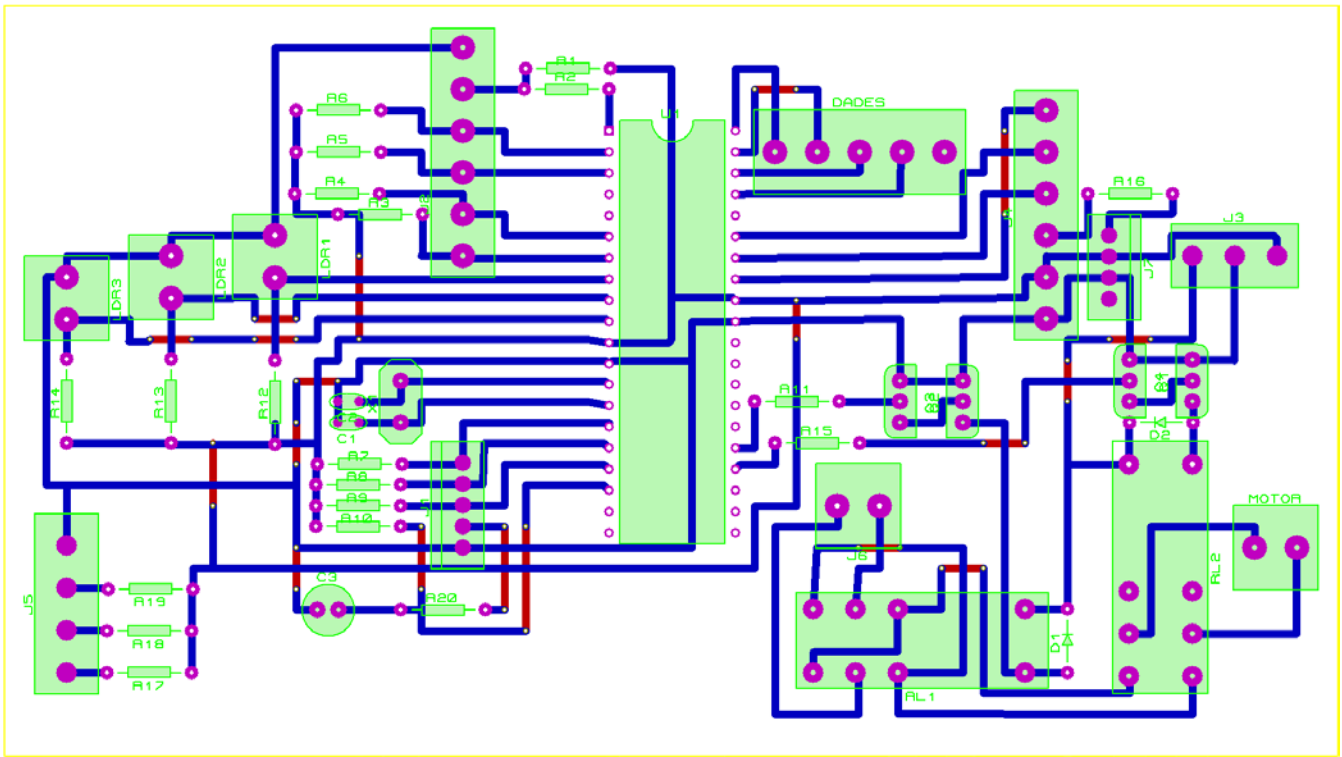
Vaig fer un plànol dels següents elements, que estan inclosos a continuació:

- La placa base
- La caixa que conté la placa base
- La rampa de llançament amb els sensors
- El conjunt del recollidor
- El reductor del motor i el codificador

Per fer tots els plànols he utilitzat bàsicament l'AutoCAD 2012, l'Adobe Illustrator CS5 i el Proteus 7.9.

3.1.2.1. Placa base

La placa base és la part on es munten la majoria de components electrònics, inclòs el microcontrolador. Dissenyar-la bé és molt important ja que si no està ben fet pot haver-hi algun problema i que es cremi algun circuit. En blau hi ha les línies que van per sota i en vermell les que van per sobre, els components són de color verd clar i les seves potes de color magenta.



Imatge del disseny de la placa base

Dissenyar-la ha estat bastant complicat perquè s'han de col·locar tots els elements de manera que puguin soldar-se correctament i després sigui pràctic a l'hora de connectar i desconnectar els pins que han de sortir d'aquesta.

En el disseny he intentat posar a l'esquerra tots els elements que després estaran a la part superior de la taula i a la dreta els que van cap al recollidor per facilitar l'accés.

Pel que fa a les resistències, hi ha sis grups, les dues que es troben a la part superior (R1 i R2) serveixen per a reiniciar el PIC, aquestes són estàndard per a tots els PIC. Hi ha quatre resistències que són pels botons (R3 – R6), són de 1kΩ. Hi ha quatre resistències més d'1kΩ per als fi de cursa i el codificador (R7 – R10). Tres resistències de 4,7kΩ (R12 – R14) per a les LDR. Dues resistències d'1kΩ (R11 i R15) per als relés i tres resistències de 90Ω pels LED (R17 - R19).

Per detectar s'utilitza una LDR, que és un element sensible a la llum que disminueix la seva resistència quan li arriba llum.

Perquè el PIC pugui detectar el pas de la bola amb les LDR, cada detector té una resistència de 4,7KΩ(R_1). Esta calculada per a donar un nivell lògic alt(5V) quan passa la bola i així poder detectar-la. Per fer-ho s'utilitza un divisor de tensió (Figura 2) que es calcula amb la següent fórmula:

$$V_{out} = \frac{R_2}{R_1 + R_2} \cdot V_{in}$$

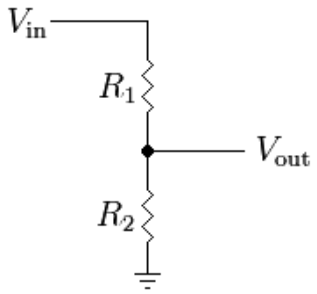


Figura 2

En aquest cas el valor de residència de les LDR(R_2) és de 4,66KΩ quan passa la bola i de 49,6KΩ quan no hi passa res.

Les tres últimes de 90Ω són per alimentar els LED que il·luminen els detectors i que així puguin detectar bé. La fórmula és la següent:

$$V_0 - 2{,}5 = 0{,}03 \cdot R$$

On V_0 és el voltatge aplicat (5V); 2,5 és el voltatge d'un LED i 0,03 és l'amperatge que hi ha de passar.

A part, l'oscil·lador de 20MHz utilitza dos condensadors de 33pF per poder treballar amb suavitat, aquests venen donats en unes taules on, en funció de la freqüència d'oscil·lació, s'utilitzen uns condensadors o uns altres, en aquest cas per 20MHz són necessaris 33pF per poder estabilitzar-lo.

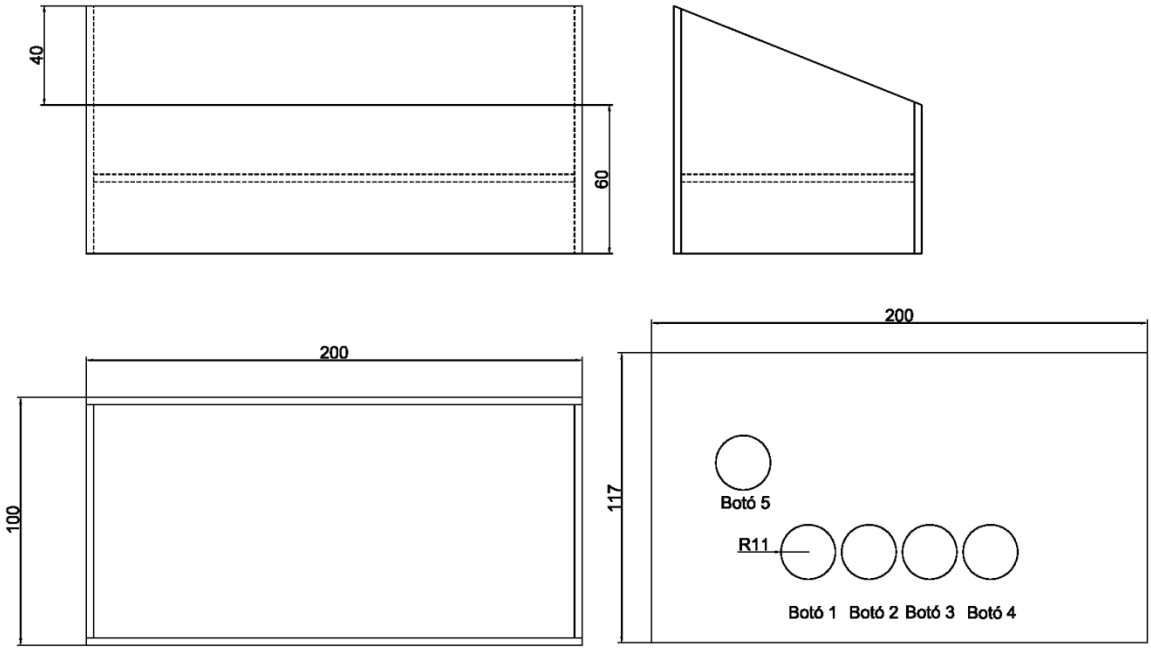
A part, també hi ha un sòcol que permet posar i treure el PIC per programar-lo en cas que no funcionés de la manera esperada i s'hi hagués de posar un programa diferent.

Finalment també hi ha dos relés que serveixen per activar, desactivar el motor i canviar-lo de sentit, permetent així aïllar la potència, que podria malmetre el PIC, d'aquest. Cada relé te un díode per evitar el rebot, que pot malmetre el circuit quan aquest es desactiva. Aquests són activats mitjançant un parell de transistors que amplifiquen el senyal de sortida del PIC fins a assolir la potència necessària per poder activar els relés.

3.1.2.2. Caixa

La caixa és la que conté la placa base i a on es connecten totes les entrades i sortides. Ha de ser suficientment gran perquè hi puguin cabre tots els cables. Si no està ben dissenyada es poden trencar alguns components de dintre perquè poden quedar aixafats.

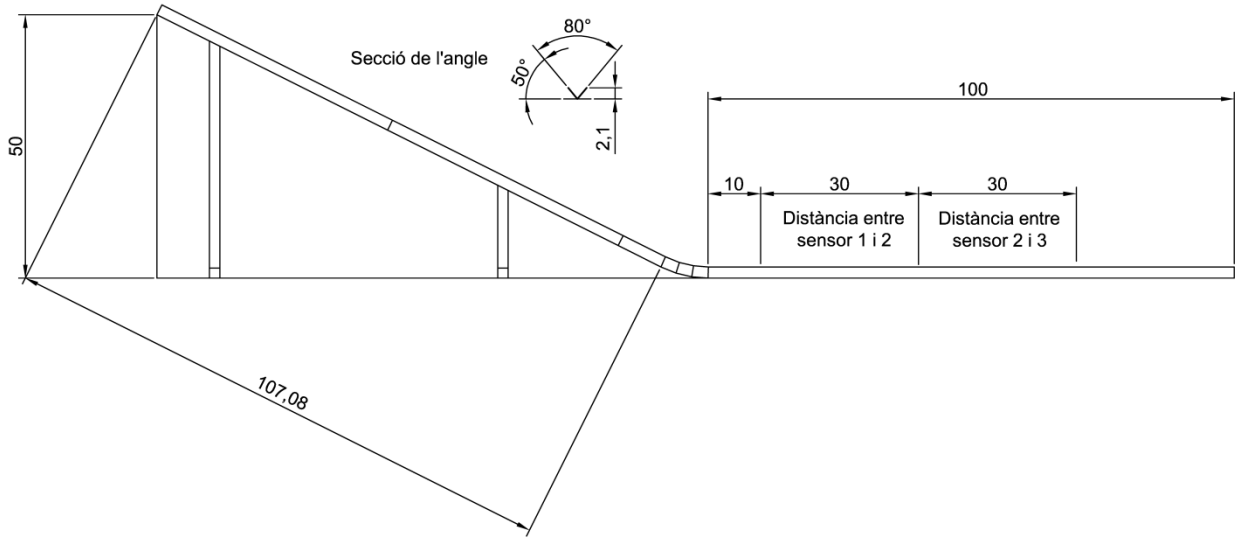
A la part de darrera de la caixa hi he incorporat uns connectors que permeten una connexió més còmoda i ràpida amb tots els components del robot. N'hi ha dos per a la alimentació i dos per a connectar-se amb les parts externes del robot, un és per als detectors i l'altre pel recollidor.



Esquema de la caixa

3.1.2.3. Rampa de llançament i sensors

L'alçada de la rampa està pensada perquè tot i tirant l'esfera des del punt més alt no arribi massa lluny i el robot pugui recollir-la. Els sensors estan separats 30cm els uns dels altres per poder tenir un temps considerablement precís per cronometrar la velocitat, si la distància és massa curta podria no haver-hi prou precisió i si és massa llarga, no hi ha temps perquè es posicioni el recollidor, ja que immediatament després de cronometrar-se l'esfera aquesta cauria i ja no hi hauria temps per posicionar-se; així que, 30cm són la distància idònia.

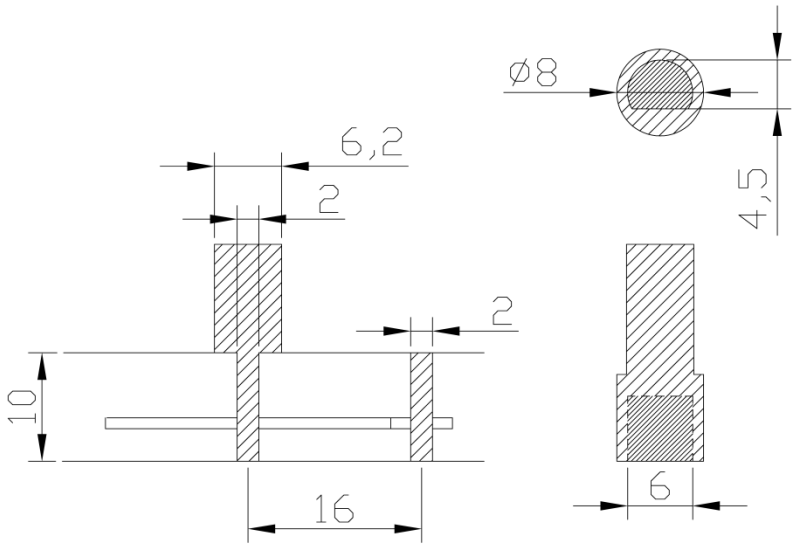


Esquema de la rampa de llançament

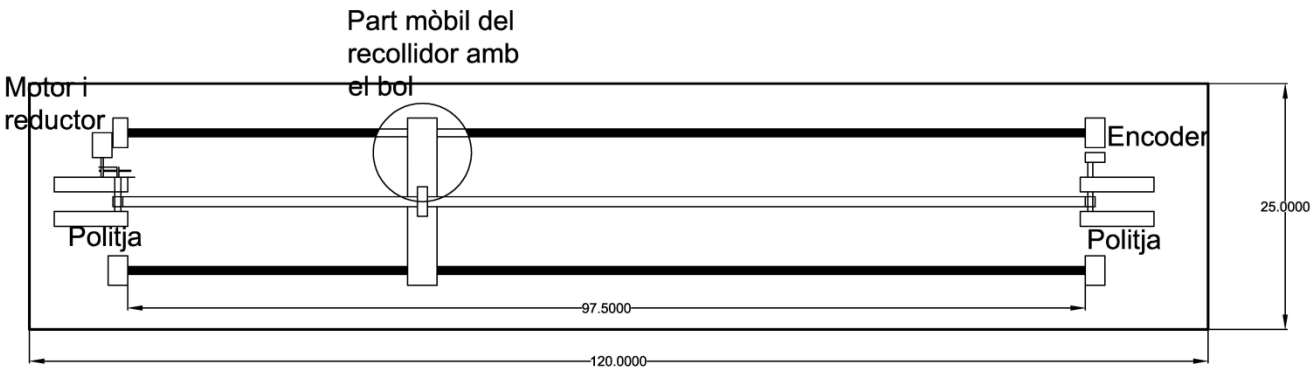
3.1.2.4. Recollidor

El recollidor té la corretja per la part central i als dos costats té les guies per les quals es desplaça això és perquè la corretja pugui moure sense dificultat la part mòbil. La corretja s'enganxa amb una peça metàl·lica que va enroscada a la part mòbil, que és feta de fusta. Al principi i al final del recorregut de les barres d'acer hi ha un fi de cursa que serveix per aturar-se abans que es pugui trencar.

Les distàncies entre els dos engranatges del reductor (a l'esquerra) són les necessàries perquè el motor faci 6 voltes i l'eix de sortida només en faci una per tal que el motor tingui suficient força i no vagi massa ràpid. La peça que encaixa amb el codificador (a la dreta) està feta perquè aquest no rellisqui i pugui contar bé.



Imatge del disseny del reductor (esquerra) i de l'eix de l'encoder (dreta)



Disseny del recollidor

3.2. Material i cost

Aquest és tot el material utilitzat i el seu cost.

Nom	Fabricant	Descripció	Preu	Q[1]	Total
PIC16F877	Microchip	Controlador PIC	9,20 €	1	9,20 €
Sòcol	FCI	Sòcol de 40 potes per al PIC	0,58 €	1	0,58 €
VM134	Velleman	Programador de PIC	53,09 €	1	53,09 €
TIL111	ISOCOM	Optoacoblador	1,01 €	2	2,02 €
Cristall 20MHz	AKER	Cristall ressonador de 20MHz	2,19 €	1	2,19 €
RL 40.52	Finder	Relé de 12V i 2 contactes commutats	2,88 €	1	2,88 €
RL G5LA	OMRON	Relé de 12V i 1 contacte commutat	0,77 €	1	0,77 €
Fi de cursa	---	Fi de cursa DC1C-A1RC	4,70 €	1	4,70 €
Fi de cursa	---	Fi de cursa DG1C-B1LA	1,86 €	3	5,58 €
Encoder	ALPS	Codificador de 12 polsos per volta			
Transformador	NIMO	Transformador de 230V a 12V de 2A	10,00 €	1	10,00 €
Transformador	NIMO	Transformador de 230V a 12/5V de 2A	11,57 €	1	11,57 €
Pantalla LCD	MIDAS	Pantalla LCD de 20x4	13,53 €	1	13,53 €
Connector	WURTH ELEKTRONIK	Connector mascle de 2 pins	0,55 €	5	2,75 €
Connector	WURTH ELEKTRONIK	Connector femella de 2 pins	0,16 €	5	0,80 €
Connector	WURTH ELEKTRONIK	Connector mascle de 4 pins	0,87 €	2	1,74 €
Connector	WURTH ELEKTRONIK	Connector femella de 4 pins	0,20 €	2	0,40 €
Connector	WURTH ELEKTRONIK	Connector mascle de 5 pins	1,07 €	1	1,07 €
Connector	WURTH ELEKTRONIK	Connector femella de 5 pins	0,20 €	1	0,20 €
Connector	IMO PRECISION CONTROLS	Clema de 2 pins	0,29 €	2	0,58 €
Connector	IMO PRECISION CONTROLS	Clema de 3 pins	0,45 €	1	0,45 €

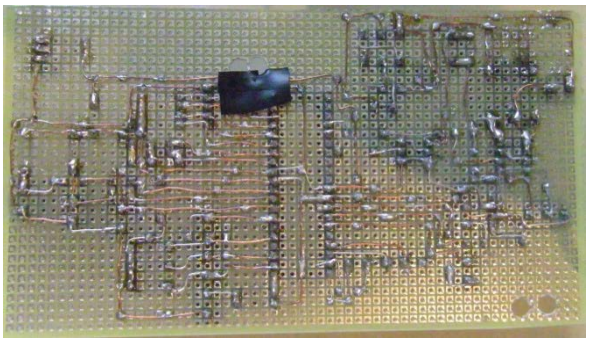
Connector	ICC	Connector d'entrada d'alimentació	0,32 €	1	0,32 €
Connector	---	Jack d'alimentació	0,86 €	1	0,86 €
Connector	---	Connector de 9 pins mascle	0,32 €	1	0,32 €
Connector	---	Connector de 9 pins femella	0,48 €	1	0,48 €
Connector	---	Connector de 15 pins mascle	0,94 €	1	0,94 €
Connector	---	Connector de 15 pins femella	1,20 €	1	1,20 €
Resistència	Multicomp	Resistència 10k	0,01 €	1	0,01 €
Resistència	Multicomp	Resistència 4,7K	0,01 €	3	0,03 €
Resistència variable	VISHAY SFERNICE	Resistència variable de 4,7K	1,31 €	1	1,31 €
Corretja	Alarsis	Corretja de poliuretà de 3mm de pas i 9mm d'amplada	12,00 €	2	24,00 €
Politja	Alarsis	Politja de 15 dents de 3mm de pas i 9mm d'amplada	10,00 €	2	20,00 €
Motor	Como Drills	Motor de 13.360 rpm i 154,4 gcm	7,29 €	1	7,29 €
Planxa de plàstic	---	Planxa de plàstic blanc de 3mm	6,95 €	1	6,95 €
Llistó de fusta	---	Llistó de fusta de 13x40x900mm	2,99 €	3	8,97 €
Barra de fusta	---	Barra de fusta de 12mm	1,99 €	1	1,99 €
Barra d'acer	---	Barra d'acer de 2mm foradada	1,50 €	1	1,50 €
Barra d'acer	---	Barra d'acer de 6mm	2,00 €	1	2,00 €
Cargol	---	Cargol 5x20 mm	3,30 €/Kg	0,79	2,61 €
Cargol	---	Cargol 5x40 mm	4,05 €/Kg	0,50	2,03 €
Femella	---	Femella M5	4,95 €/Kg	0,57	2,82 €
Angle	---	Angle de ferro	0,24 €	21	5,04 €
Frontissa	---	Frontissa de 30x19mm	2,65 €	1	2,65 €
Xapa d'unió	---	Xapa d'unió de 50x15mm	0,54 €	2	1,08 €
			Total		218,50 €

[1]Quantitat

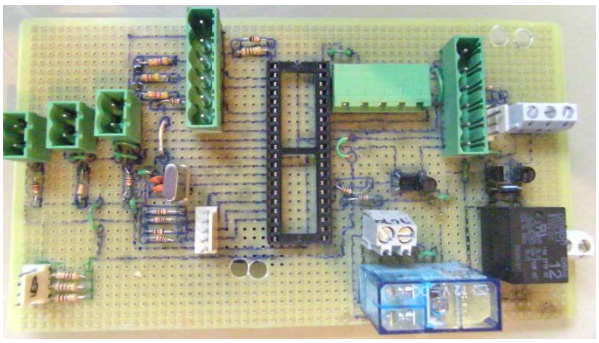
3.3. Procediment

En aquest apartat explicaré tots els passos per muntar el robot.

1. A partir de l'esquema de la placa base, soldem tots els components a la placa base



La placa electrònica vista des de sota

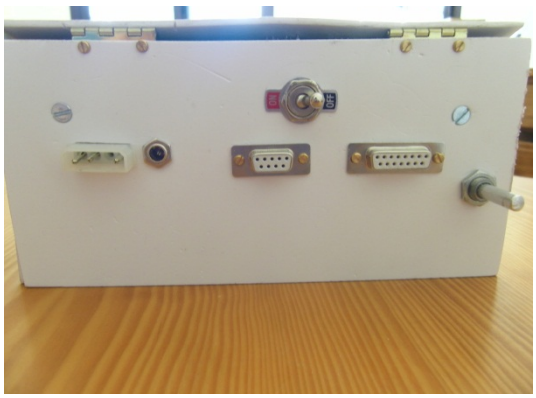


La placa electrònica vista des de sobre

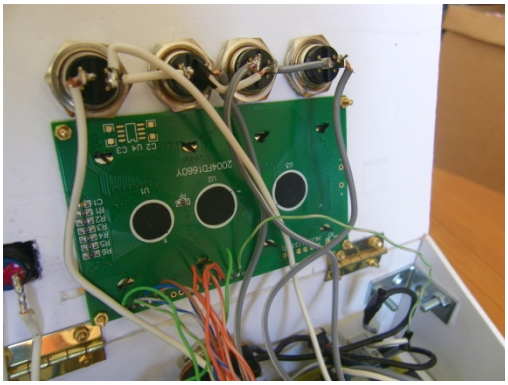
2. Muntem la caixa que contindrà tota la part electrònica, amb els botons i la pantalla LCD.

Imatge de tots els components de la caixa

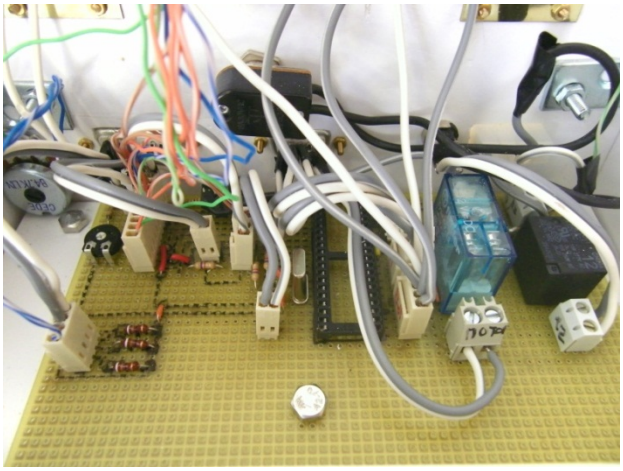
La caixa vista des de forai per davant



Connexions posteriors de la caixa

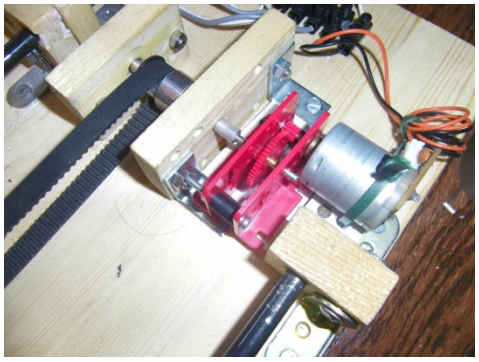


Connexió dels botons



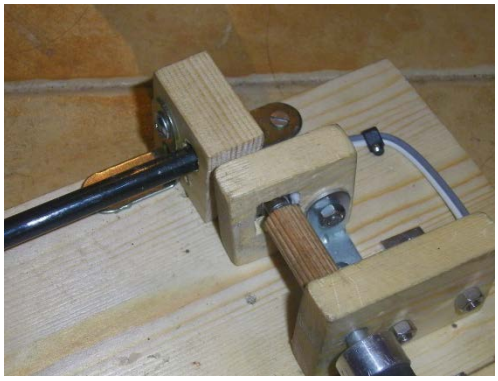
Imatge del cablejat interior de la caixa cap als elements externs

3. Un cop hem muntat la caixa comencem a muntar la part del recollidor, primer hem de fer el reductor i l'eix que en surt d'aquest.



Imatge del reductor muntat ja sobre la fusta que conté la part del recollidor

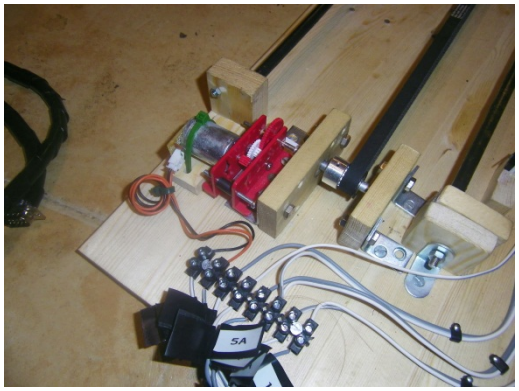
4. Un cop hem muntat el reductor muntem l'altre costat del pinyó que fa girar la corretja, amb l'encoder



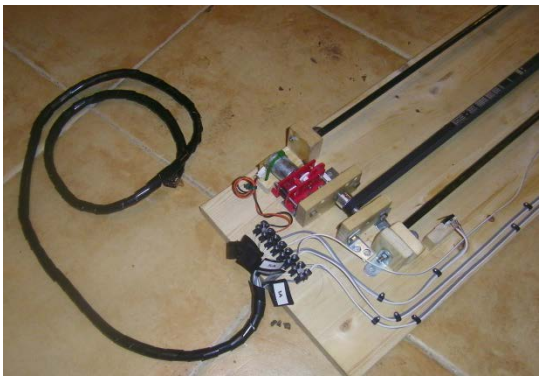
Imatge de l'encoder muntat, unit amb l'eix i connectat elèctricament

Imatge de la part de la corretja oposada al reductor

5. Un cop ja hem muntat els dos costats, ja podem posar la corretja i la part mòbil que serà la que recollirà l'esfera i podem cablejar-ho tot.

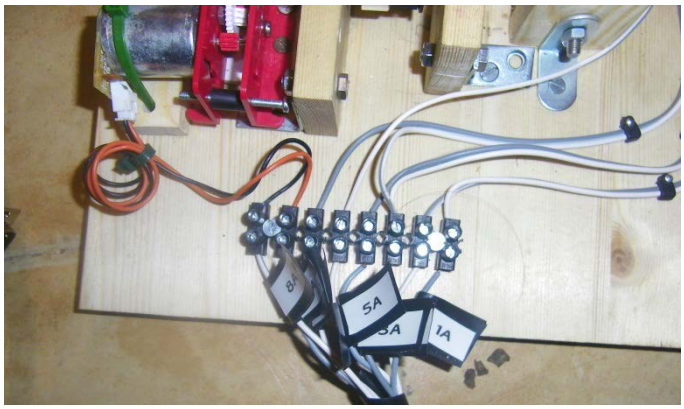


Imatge del motor ja cablejat



Imatge del recolidor muntat, amb el port de connexió cablejat

Imatge de tot el recollidor cablejat, sense port de connexió



Detall de les conexions del recollidor

6. Un cop s'ha acabat de muntar el recollidor s'han d'engreixar les guies per tal que la part mòbil es desplaci correctament.

7. Muntar la rampa de llançament i els detectors i els LED, aquests han d'anar dins d'un forat que apunti directament a la zona per on passarà l'esfera per tal de poder detectar aquesta zona únicament i que no hi hagi falsos senyals, els LED també han d'anar dins d'un forat per poder enviar la llum enfocada únicament en el detector.

Detall de la part superior de la rampa

Detall de la corba de la rampa

Detall dels detectors

Detall de les connexions dels detectors

3.4. Avaluació del procés

Un cop he acabat de construir el projecte faig una llista de les dificultats que he tingut a l'hora de muntar-lo i anar-lo construint.

Primerament, no sabia quin tipus de microcontrolador utilitzar ja que n'hi havia que tenien característiques molt semblants, dubtava entre la família PIC i la família PICAXE de microcontroladors. Un cop me'ls vaig haver mirat vaig veure que els PIC estaven molt més oberts a qualsevol llenguatge de programació, la qual cosa em permetia escollir el llenguatge que més coincidís amb les meves necessitats.

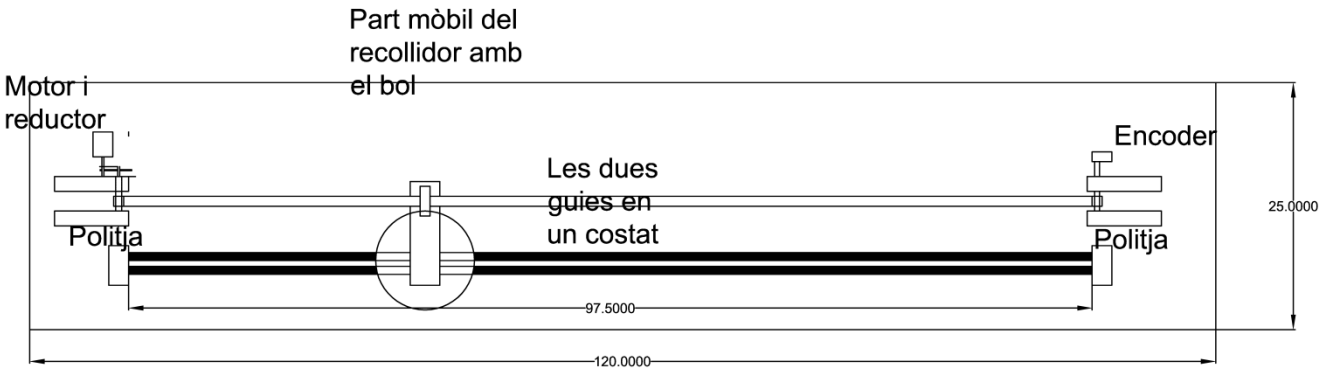
També vaig dubtar del llenguatge que havia d'utilitzar, dubtava entre el BASIC o el CCS i finalment em vaig decantar pel segon ja que aquest permetia afegir-hi moltes llibreries (parts de programa que ja estan fetes i són d'ús públic, de les quals pots agafar-ne trossos en qualsevol punt del programa) cosa que el BASIC no deixava fer o no vaig trobar la manera de fer.
A més a més, el BASIC només permetia treballar amb variables fins a 65536 valors i amb el CCS podia agafar-ne que arribessin fins a 2^{32} i jo per fer els càlculs necessitava variables que fossin molt grans per poder tenir la precisió necessària.

Quan em vaig posar a buscar un motor tampoc trobava el que jo necessitava ja que primerament vaig fer els càlculs malament i pensava que necessitava un motor amb molta potència, després vaig veure que no en necessitava tanta com em pensava i vaig agafar un motor d'una aspiradora vella de casa.

Mentre estava construint el reductor, estava fent-me un eix i aquest es va trencar en dos, cosa que em va retardar uns dies a l'hora de construir el robot mentre no en feia un altre.

A part, mentre muntava la part del recollidor em vaig adonar que amb el sistema de guies que havia muntat al principi, les dues en un costat en lloc de una a cada costat com és ara, la part mòbil quedava encallada. Això feia que a l'estirar la part mòbil, aquesta es torcés i no es pogués moure perquè fregava amb les guies.

Per què no passes això havia de tensar molt la corretja però aquesta relliscava i es deixava anar. Per tant vaig haver de moure una de les guies a l'altra costat de la corretja per tal que aquesta quedes al centre de les dues guies i anés més suau. Aquest canvi va donar bons resultats.



Disseny vell del recollidor

Finalment, quan ja havia aconseguit muntar-ho tot vaig anar a provar el funcionament del robot i aquest només va engegar els LED però no feia el que se li havia demanat, a continuació el microcontrolador es va sobreescalfar i es va fondre internament així que no vaig poder provar-lo.
La solució va ser tornar a fer una placa base nova afegint-hi resistències als botons perquè com que aquests no en tenien al prémer un botó el PIC va rebre una intensitat de corrent massa alta i això va ser el que va provocar que es cremés.

Tot i havent solucionat el que semblava que era l'últim problema llavors em vaig trobar amb que el robot no feia el que se li demanava i no agafava l'esfera o obeïa les ordres donades malament. Per solucionar això vaig haver de crear un programa nou fent tot de proves, cosa que va portar un mes de feina i també aquest error venia provocat per un rebot que donava el codificador a l'hora d'obrir i tancar el circuit.

Tot i els problemes, el projecte sempre pot estar sotmès a millores i no acabar mai, ja que sempre es pot millorar algun punt.

4. Conclusions

Un cop ja he acabat aquest treball he extret una sèrie de conclusions, que em permetrien orientar-lo d'una manera diferent si l'hagués de tornar a fer.

Pel que fa als objectius que em vaig plantejar al principi d'aquest treball n'hi ha alguns que els he assolit d'immediat i d'altres que han tingut més dificultat.

Del primer objectiu, que consistia en veure si era capaç de dissenyar i construir un robot que marqui el punt de caiguda d'una esfera, el puc considerar assolit tot i que ha costat assolir aquest punt ja que al principi no pensava que la dificultat fos tant alta. A més a més el fet d'haver tingut tants errors m'ha ajudat a aprendre com s'hauria de fer si es fes tot de nou.

El segon i tercer objectius, aprendre a programar microcontroladors i aprendre el seu funcionament, sí que els dono com a complert ja que he pogut entendre molt millor quina és la seva manera de treballar i de quines maneres es poden programar. A més a més, per poder escollir el llenguatge per programar-lo vaig haver d'aprendre dos llenguatges de programació per veure quin s'adaptava més a les meves necessitats.

L'últim objectiu, conèixer més sobre mecànica, també considero que s'ha complert ja que he aprés més sobre el funcionament d'un motor i les seves característiques. També he pogut veure com funciona un reductor i quines qualitats té.

Considero que fent aquest treball he assolit els objectius que m'havia proposat tot i així com que és un treball bàsicament pràctic i de construcció sempre pot estar disposat a noves millores i modificacions per tal d'augmentar-ne l'eficàcia.

El funcionament del robot es pot veure al CD ubicat al final del treball o també a través del següent enllaç de YouTube:

http://youtu.be/ZQyZHSKnY0c

5. Bibliografia i webgrafia

Manejo de interrupciones PIC

http://samacomint.blogspot.com.es/2012/05/manejo-de-interrupciones-pic.html

[Consulta: 26 de juny de 2012]

PICAXE

http://www.picaxe.com/

[Consulta: 26 de juny de 2012]

Microchip Technology Inc.

http://www.microchip.com

[Consulta: 26 de juny de 2012]

Partes del microcontrolador

http://trecedb.wordpress.com/2009/02/13/partes-del-microcontrolador/

[Consulta: 26 de juny de 2012]

Araguz López, Carles / Estació Meteorològica Digital Domèstica

http://www.evostudio.com/litus-host/tdr_cal/index.html

[Consulta: 26 de juny de 2012]

Guide to use PIC

http://www.piclist.com/images/www/hobby_elec/e_pic.htm

[Consulta: 27 de juny de 2012]

Superobòtica

http://www.superrobotica.com/Default.htm

[Consulta: 27 de juny de 2012]

Rocatelc

http://www.rocatek.com/forum_plc1.php

[Consulta: 27 de juny de 2012]

MikroElektronika

http://www.mikroe.com/products/view/476/pic-microcontrollers-programming-in-basic/

[Consulta: 5 de septembre de 2012]

Procès tecnològic

http://www.edu365.cat/batxillerat/comfer/projecte/index.htm

[Consulta: 25 de gener de 2013]

Bodington Esteva, Christian, *Basic para microcontroladores PIC,* Trafford Publishing – International, 2008

García Breijo, Eduardo, *Compilador CCS y simulador PROTEUS para Microcontroladores PIC*, Alfaomega Grupo Editor, 2008

Institut d'Estudis Catalans, *Diccionari de la llengua catalana,* Enciclopèdia catalana i Edicions62

Annex: Programa utilitzat

```
//Tots els comentaris van precedit per dues barres perquè el
compilador no els
//tingui en compte
                                          //DIRECTIVES DE PREPROCESSAT

#include <16F877A.h>
#device adc = 10             //En aquesta part del programa es
defineixen les
#include <MATH.H>            //directives necessàries per al
funcionament del
#use delay (clock=20000000)  //microncontrolador. Es tenen en compte a
l'hora
#fuses HS, NOWDT, NOPROTECT  //de compilar el programa.
//#include <LCD420.c>     //S'hi posa la velocitat de treball, el
tipus
#include <Lib_Int_EEPROM.c>  //d'oscil·lador, la configuració dels
ports i
#use standard_io(A)          //s'hi ifitxers necessaris per poder fer
#use standard_io(B)          //anar la pantalla LCD o per poder fer
càlculs
#use standard_io(C)          //matemàtics.
#use standard_io(D)
#use standard_io(E)

/*********************************************************************
********/

/*
PIN A0 = LDR 1
PIN A1 = LDR 2
PIN A3 = LDR 3
PIN A5 = BOTÓ 1 - INVERTIT

PIN E0 = BOTÓ 2 - INVERTIT
PIN E1 = BOTÓ 3 - INVERTIT
PIN E3 = BOTÓ 4 - INVERTIT

PIN C0 = FI 1 - INVERTIT
PIN C1 = FI 2 - INVERTIT
PIN C2 = FI 3 - INVERTIT

PIN C3 = ENCODER - INVERTIT

PIN C5 = MOTOR
PIN C6 = CANVI DE SENTIT

*/

                        //DECLARACIÓ DE FUNCIONS
Int32 temps = 0;        //En aquesta part del programa, es defineixen
les
Signed Int16 enc = 0;
Signed Int16 enc2 = 0;  //funcions o les variables que es poden
utilitzar en
Int8 altura;            //qualsevol part del programa.
Float t;                //Aquestes dues variables només donen
informació
                        //de la posició:

Int8 control = 0;       //1 vol dir sin rebut de l'encoder 0 no
```

```
Float tempsc;
Int8 duracio = 50;
Int8 duracio2 = 50;
#int_RTCC
void RTCC_isr(void){
   ++temps;
   SET_TIMER0(100);

}

void Calibracio (int16 Encoder)
{
   Int16 Contar = 0; //(Encoder/2);
   Int8 control = 0; //0 no ha entrat senyal 1 sí ha entrat
   output_high (PIN_C6);       //Encendre el canvi de sentit
   delay_ms(50);
   output_high (PIN_C5);       //Encendre motor

   while (Input (PIN_C2) == 1);
   output_high(PIN_B4);
   output_high(PIN_B5);
   Encoder = 150;

   output_low (PIN_C5);       //Apagar motor
   output_low (PIN_C6);       //Apagar canvi de sentit
   delay_ms(1000);
   output_high (PIN_C5);      //Encendre motor
   while (Contar != Encoder)
   {
      if (input (PIN_C3) == 1) control = 0;
      if (input (PIN_C3) == 0)
      {
         if (control == duracio2) ++Contar;
         if (control <= (duracio2 + 5)) ++control;
         if (input(PIN_C0) == 0) Contar = Encoder;
      }
   }
   output_low(PIN_B4);
   output_low(PIN_B5);
   output_high(PIN_B6);
   output_high(PIN_B7);

   output_low (PIN_C5);
   return;
}

////////////////////////////////////////////////////

void Alt (void)
{
   Int16 altura = 0;
   Float alt;
   altura = 69;

   Altura3:
   alt = (float)altura;
   alt /= 100;
   tempsc = sqrt(alt/4.9);
}
```

```
void Detecta1 (void)
{
   t = 0;
   temps = 0;
   enable_interrupts(GLOBAL);
   SET_TIMER0 (100);
   enable_interrupts(INT_TIMER0);
   int16 Voltatge = 0;
   while(Voltatge <= 313)
   {
      set_adc_channel(3);
      delay_us(20);
      Voltatge = read_adc();
   }

   return; //Si la LDR 2 dona senyal sortir

}

////////////////////////////////////////////////////

Float Calcul_distancia(Float tempsc, Float t)
{
   Float v = (0.3/t);    //0,3 és la separació dels detectors
   Float d = (v*tempsc); //distancia = velocitat * temps de caiguda
   if (d <= 0) return 0;
   else return d;
}




////////////////////////////////////////////////////4

void main()             //COMENÇA EL PROGRAMA PRINCIPAL
{
   Int8 control2 = 0, control3 = 0;
   Int16 voltatge = 0;
   Int16 contar2 = 0,contar3=0;
   Int16 Encoder = 45, posicio1 = 0;
   Float dist = 0;
   Int32 tempsA = 0;
   Int16 tt = 0;

   disable_interrupts (GLOBAL);
   setup_timer_0(RTCC_INTERNAL|RTCC_DIV_32);

   output_low(PIN_B4);
   output_low(PIN_B5);
   output_low(PIN_B6);
   output_low(PIN_B7);

   disable_interrupts(GLOBAL);
```

```
Main0:
enc = 0;
enc2 = 0;
Alt();
output_low(PIN_B4);
output_low(PIN_B5);
output_low(PIN_B6);
output_low(PIN_B7);

Calibracio(Encoder);
delay_ms(700);
output_low(PIN_B4);
output_low(PIN_B5);
output_low(PIN_B6);
output_low(PIN_B7);
output_high(PIN_B4);
delay_ms(500);
output_high(PIN_B5);
delay_ms(500);
output_high(PIN_B6);
delay_ms(500);
output_high(PIN_B7);
delay_ms(500);

output_low(PIN_B4);
output_low(PIN_B5);
output_low(PIN_B6);
output_low(PIN_B7);


Main1:
setup_adc_ports(AN0_AN1_AN3);
setup_adc(adc_clock_div_32);
for(;;)
{
   set_adc_channel(1);
   delay_us(20);
   voltatge = read_adc();
   if (voltatge >= 313) goto Main11;
   if (Input(PIN_E2) == 0) goto Main0;

}
Main11:
output_high(PIN_B4);
Detecta1();
output_high(PIN_B5);
t = temps;
tempsA = temps;
t /= 1000;
dist = Calcul_distancia(tempsc, t);
posicio1 = (int16) (dist*4000);
posicio1 /= 10;
Encoder=150;
contar2=0;
contar3=0;
enc = (posicio1 - Encoder);
enc2=-enc;
disable_interrupts(GLOBAL);
```

```
      if (enc < 0)
      {
         if (enc2>40)
         {
         output_high (PIN_C6);      //Encendre canvi de sentit
         output_high (PIN_C5);      //Encendre motor
         while (contar2 <= 40)
         {
         if (Input (PIN_C3) == 1) control2 = 0; //Control de l'encoder
         if (input (PIN_C3) == 0)
         {
            if (control2 == duracio) ++contar2;
            if (control2 <= (duracio + 5)) ++control2;
            output_high(PIN_B7);
         }
         if (Input (PIN_C0) == 0) break;
         if (Input (PIN_C2) == 0) break;
         }
      }
      }

      if (enc > 0)
      {
         if (enc>20)
         {
         output_high (PIN_C5);      //Encendre motor
         while (contar3 <= 40)
         {

         if (Input (PIN_C3) == 1) control3 = 0; //Control de l'encoder
         if (Input (PIN_C3) == 0)
         {
            if (control3 == duracio) ++contar3;
            if (control3 <= (duracio + 5)) ++control3;
         }
         if (Input (PIN_C0) == 0) break;
         if (Input (PIN_C2) == 0) break;
         }
         }
      }
   output_low(PIN_C5);
   output_low(PIN_C6);

   control2 = 0;
   control3 = 0;
   temps = 0;
   enable_interrupts(GLOBAL);
   SET_TIMER0 (100);
   enable_interrupts(INT_TIMER0);
```

```
while (Input(PIN_E2) != 0)
   {
      tt = (temps%500);
      if (tt == 2)
      {

         if (control3 == 0)
         {
            output_high(PIN_B4);
            output_high(PIN_B5);
            output_high(PIN_B6);
            output_high(PIN_B7);

         }
         if (control3 == 1)
         {
            output_low(PIN_B4);
            output_low(PIN_B5);
            output_low(PIN_B6);
            output_low(PIN_B7);
         }
      }
      if (tt == 0) control2 = 0;
      if (tt == 1)
      {
         if (control2 == 0)
         {
            control3 += 1;
            control3 = control3%2;
         }
            control2 = 1;
      }

   }
   goto Main0;
}
```