

# Com construir-nos un robot

Antoni Ferraté

**L**a millor manera d'aprendre com funciona el motor d'un cotxe és desmuntar-lo i tornar-lo a muntar. La millor manera d'aprendre robòtica és construir i experimentar amb el propi robot. Això no és una tasca difícil si es disposa d'uns mínims coneixements d'ordinadors, d'electrònica i de mecànica, que la robòtica ens donarà l'oportunitat d'ampliar i, sobre tot, relacionar, a la vegada que ens despertarà l'esperit de recerca.

Si tenim un ordinador personal, un sistema modular de construcció mecànic i una font d'alimentació, podem començar a construir el nostre robot, dividint aquesta tasca en quatre parts:

1. Disseny i construcció de l'estructura mecànica del robot.
2. Implementació de les plaques d'interfície amb l'ordinador.
3. Construcció de l'etapa de potència que alimenta els motors.
4. Programació del *software* de control.

Aquests quatre punts són els que tractaré a continuació, donant uns breus consells fruit de la pròpia experiència en la construcció de robots didàctics controlats per PC.

A mesura que aquests éssers inanimats van cobrant vida és molt estimulante anar-los dotant de característiques humanes, com ara visió, tacte, oïda, veu, intel·ligència...

ANTONI FERRATÉ és alumne de l'Escola Tècnica Superior d'Enginyeria de Telecomunicacions de Barcelona

## 1. ESTRUCTURA MECÀNICA

### 1.1 Material

Si disposem de kits de construcció educacionals estàndard, com ara *Fischertechnik*, *Legó* o *Meccano*, ens evitem la necessitat d'emprar algunes tècniques i eines especialitzades, i ja tenim elements mecànics imprescindibles, com eixos, engranatges, rodes, diferencials i cadenes. Tot això fa més fàcil reproduir cada model tal com el tenim pensat, abans d'experimentar amb modificacions i millores sobre el robot, si així ho desitgem. Els principals desavantatges són que cada peça del model està limitada a mesures discretes estàndard, i que aquests kits solen ser bastant més cars que altres materials, com ara, fusta, plàstic o làmines metàl·liques.

### 1.2 Geometria del robot

L'arquitectura del robot depèn del tipus de robot que volem crear. Per exemple, si pretenem reproduir a escala un robot industrial (braç articulat) hem de fixar el nombre de graus de llibertat (nombre d'eixos de moviment independent). En general, tres graus de llibertat són suficients per posicionar l'eina terminal del robot -generalment una pinça- en qualsevol punt de la zona de treball tridimensional. L'òptim és disposar de tres graus de llibertat addicionals -sis en total- per poder accedir al punt de treball amb qualsevol orientació, però cinc solen ser suficients (molts robots industrials de veritat tenen cinc).

És recomanable situar els motors -sobretot si són pesants- en la base del

robot i transmetre el moviment als braços mitjançant palanques, cadenes, eixos, cables guiats, corretges o cordills. D'aquesta manera es pot aconseguir un millor comportament cinemàtic i dinàmic, i també una independència entre els moviments dels braços: és convenient que el moviment d'un braç proper a la base del robot no modifiqui l'orientació dels braços superiors.

Els kits pneumàtics de *Legó* i *Fischertechnik* són adequats per fer la pinça o grapa terminal del robot, a causa del reduït pes, la facilitat de transmetre l'aire als pistons de la pinça per tubs de goma, i el grau de flexibilitat que tenen els dits, cosa que en evita el control lineal del moviment en obrir i tancar la grapa.

## 2. INTERFÍCIE AMB L'ORDINADOR

Qualsevol ordinador personal (des d'un Sinclair fins a un 486) ens serveix per a controlar el robot, però haurem de perdre la por a fer-lo malbé. L'ús d'optoïlladors entre els busos de l'ordinador i les nostres plaques elimina tot risc. En aquest apartat donaré tres solucions de diferent complexitat i prestacions per comunicar el robot amb un PC:

### 2.1 Port paral·lel

Disposem de 8 línies de transferència de dades que es corresponen amb els pins que van del 2 al 9 del connector de 25 pins del port paral·lel (Centronics). Els nivells de tensió de treball són de 0 i 5 volts. Podem accedir a aquests 8 bits de dades en les següents adreces segons el port d'impressora que usem:

LPT1 : 03BC hex  
LPT2 : 0378 hex  
LPT3 : 0278 hex

## 2.2 Port de jocs

Poca gent coneix que dins del seu PC hi poden haver quatre magnífics conversors A/D, que podem aprofitar en la robòtica per llegir sensors resistius, molt útils en el control lineal d'un moviment.

Efectivament, molts PC ja porten incorporat el port de jocs, el qual permet llegir la posició de dos *joysticks* mitjançant la lectura del valor de la resistència dels dos potenciòmetres de cada *joystick*, obtenint, de forma digitalitzada, les coordenades X i Y de cada *joystick* (un valor proporcional a la resistència). També ens permet llegir l'estat dels dos botons de cada *joystick*, tal i com s'il·lustra en el diagrama de connexions de la figura 2.

La interrupció 15h de la BIOS del PC ens permet llegir les 4 entrades resistives. La següent subrutina escri-

ta en llenguatge C ens il·lustra com fer-ho:

```
void joystick(int *ax, int *ay, int *bx, int *by)
{
  union REGS regs;
  regs.h.ah=0x84;
  regs.x.dx=0x1;
  int86(0x15,&regs,&regs);
  *ax=regs.x.ax; /*potenciòmetre X, Joystick A */
  *ay=regs.x.bx; /*potenciòmetre Y, Joystick A */
  *bx=regs.x.cx; /*potenciòmetre X, joystick B */
  *by=regs.x.dx; /*potenciòmetre Y, joystick B */
}
```

L'estat dels 4 botons es pot llegir en els 4 bits de més pes del port 0201 hex.

## 2.3 Targeta d'entrades i sortides

Les targetes d'entrades i sortides es connecten al bus d'expansió del PC i es poden comprar als distribuïdors de plaques per PC com *PClabcard*,

*Axiom*, etc, o ens la podem fer nosaltres mateixos en una tarja prototipus.

Aquí exposaré el disseny d'una placa molt simple i barata amb només dos CI, però que ens permet disposar de fins a 24 entrades o sortides digitals, que podem aprofitar per connectar con-versors A/D i D/A, enviarsenyals de control o llegir l'estat d'interruptors, entre d'altres coses. El cor del disseny és el *Programmable Peripheral Interface (PPI) 8255 A* i té com a principal avantatge la possibilitat de programar diversos subgrups dels 24 bits com a entrada o com a sortida, en funció de les necessitats del nostre robot.

Les adreces per accedir als ports són les següents:

port A: 0240 hex.  
port B: 0241 hex.  
port C: 0242 hex.  
port de control: 0243 hex.

La paraula de control que enviarem a la direcció 0243 hex. per a programar la placa té el següent format: 1 0 0 A<sub>h</sub> 0 B<sub>h</sub> C<sub>h</sub>.

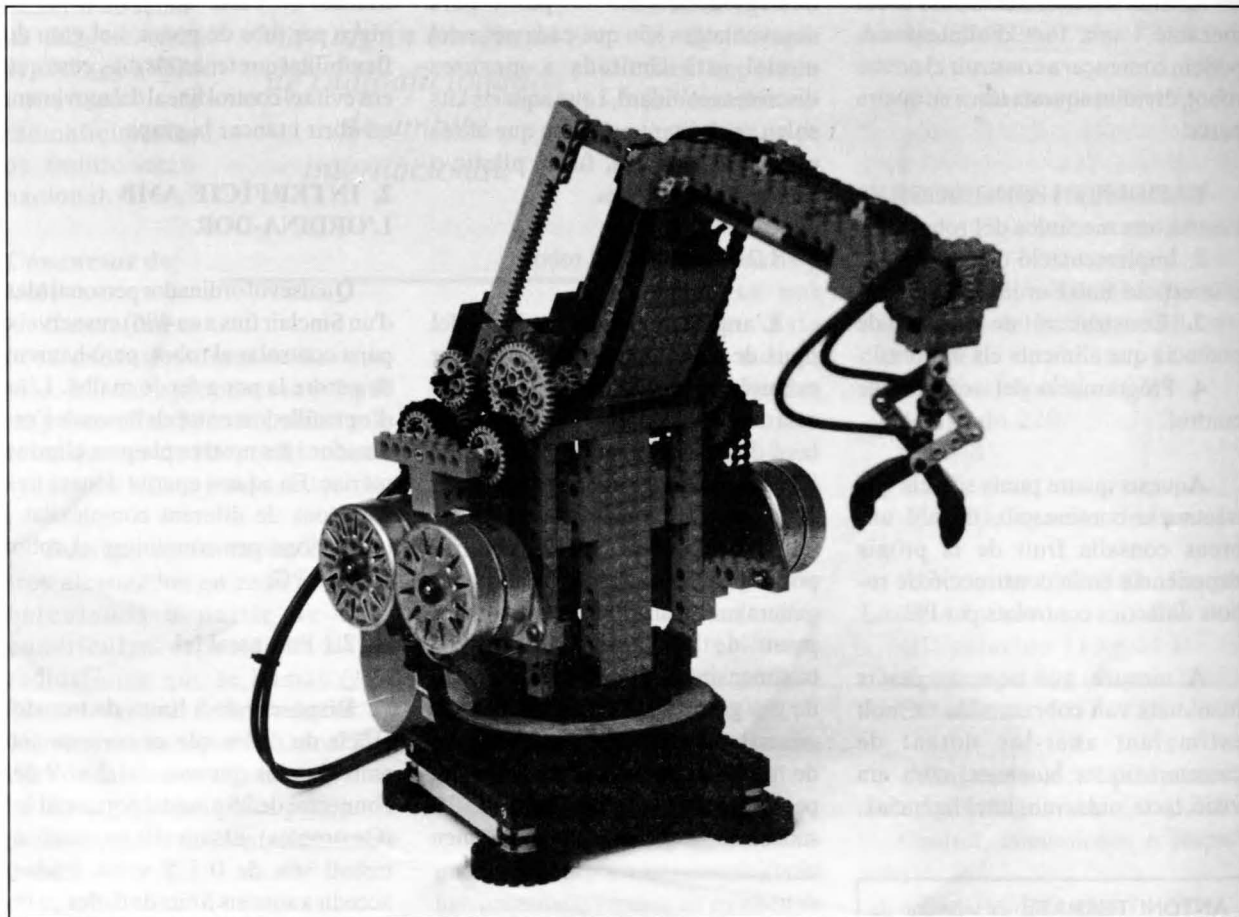


Figura 1.- Robot industrial amb 5 graus de llibertat i pinça pneumàtica construït amb *Lego*.

A, B, C<sub>b</sub>, i C<sub>1</sub> es corresponen respectivament amb el port A, port B, 4 bits més significatius del port C i 4 bits menys significatius del port C. A cada un d'ells assignarem un 1 si el volem definir com a entrada al PC, o un 0 si els volem definir com a sortida.

**Exemple:** El següent programa escrit en llenguatge C assigna els ports A i B del PPI (16 línies) com a sortida i les 8 línies del port C com a entrada. A continuació envia 8 uns pel port A i llegeix el port C.

```

outport(0x243,0x89); /* 89 hex =
10001001 bin */
outport(0x240,0xFF); /* FF hex =
11111111 bin */
c=inportb(0x242);

```

### 3. ETAPA DE POTÈNCIA I MOTORIS

Les sortides del 8255 A i del port paral·lel són compatibles TTL, això significa que no tenen una gran capacitat de corrent. El xip *ULN2001* (*Buffer darlington* de 7 canals) és adequat per excitar les bobines de relés i de motors (de cc o pas a pas).

A diferència dels motors de cc en què l'eix, senzillament, gira, un motor pas a pas gira en passos discrets molt precisos a la freqüència dictada per l'ordinador. En alguns aspectes és el somni de l'enginyer de robots fet realitat, ja que ens descarrega de la tasca de llegir la posició de l'eix mitjançant un sensor de posició

(normalment òptic o resistiu), perquè un simple comptatge dels impulsos aplicats al motor pas a pas ens determina l'angle girat. Per tant és un sistema de control en llaç obert, però l'alimentació dels motors pas a pas no és tan simple com connectar els terminals a una bateria: es necessiten circuits electrònics que vagin excitant, en la seqüència adequada, les bobines del motor. Afortunadament hi ha xips especialitzats en aquesta tasca, que, a més a més, subministren la potència necessària per alimentar els motors. Un d'aquests xips és el *SAA 1027* (fig.4), el qual connectem directament a dos línies d'un port de sortida de l'ordinador: per la línia T enviem un tren de polsos, la freqüència del qual

determina la velocitat dels passos i per la línia R indiquem el sentit de rotació de l'eix del motor. Els nivells de tensió de T i R són compatibles TTL i cada etapa de sortida (Q1, Q2, Q3 i Q4) és capaç de commutar un corrent de 350 mA. Incorpora un díode integrat de protecció (línia D) contra els pics transitoris produïts durant la commutació de les bobines del motor.

El xip *SAA 1027* és apte per a diversos motors pas a pas unipolars. A títol d'exemple, dono la referència d'un motor pas a pas de 7,5 graus per pas, usat en el robot de *Lego* de la figura 1: *AIRPAX 9904 112 35014*.

### 4. SOFTWARE DE CONTROL

El *software* que controli el nostre robot el podem fer en qualsevol llenguatge de programació que coneguem: Pascal, Basic, C. Primer farem les subrutines de baix nivell: aquelles que treballen a nivell de bits, per generar els senyals que enviarem a cada motor per tal de que l'eix giri un pas, però sense sobrepassar en cap cas els límits físics del moviment de cada braç (ho podem assolir mitjançant microrup-tors situats en els finals de carrera, i connectats a un port d'entrada de l'ordinador).

L'operativitat del nostre robot, a l'igual que un robot industrial de debò, requereix que el programa de

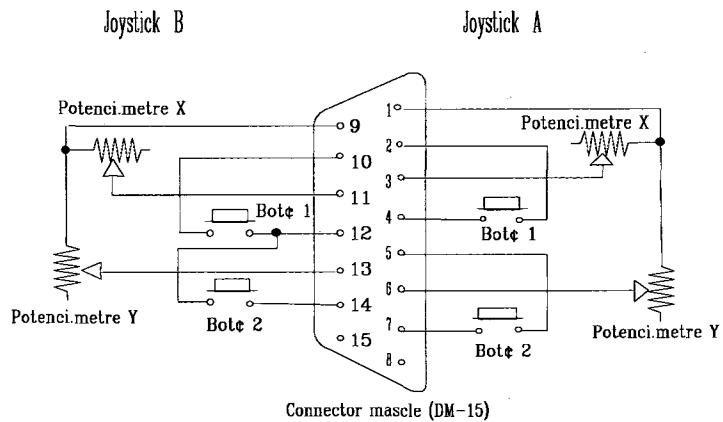


Figura 2.- Esquema de les connexions al connector del port de jocs del PC per llegir 4 sensors resistius i l'estat dels botons.

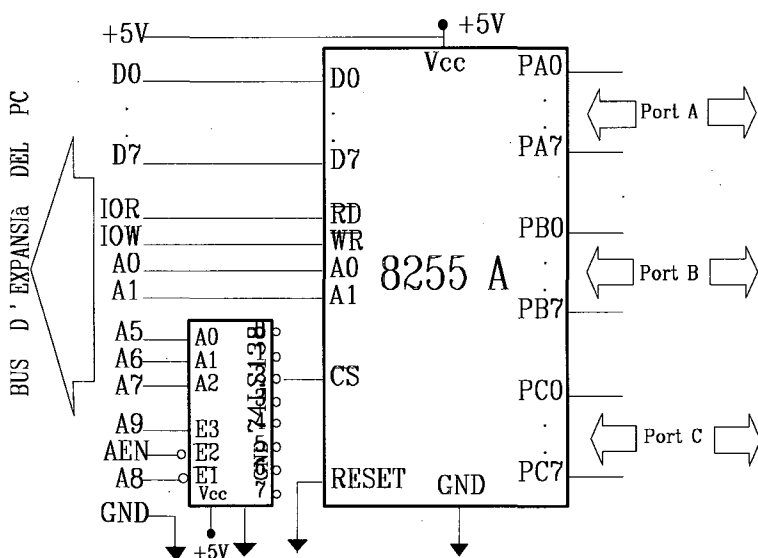


Figura 3.- Esquema de la placa programable per PC amb un circuit PPI 8255A i un decodificador d'adreces 74LS138.

control sigui capaç d'executar tres tasques ben diferenciades: Una primera fase d'inicialització, una fase d'aprenentatge, en què el robot es mou obeint uns comandaments manuals i es memoritzen certes posicions, i, per últim, una fase de reproducció, en què es repe-teixen els moviments memoritzats en el mode d'aprenentatge.

#### 4.1 Fase d'inicialització

En un sistema en llaç obert, com ara el cas d'usar motors pas a pas, tenim dos inconvenients: d'una banda ens cal conèixer una posició inicial per tal d'agafar una referència a partir de la qual hem de portar el recompte dels passos. D'altra banda, si apliquem un parell massa elevat a l'eix del motor, aquest es mourà un nombre desconegut de passos (pèrdua de passos) i no tindrem més remei que tornar a buscar la posició inicial de referència (reinicialitzar el robot). Això ho podem assolir, en el cas d'un braç articulat, aprofitant els microrruptors que detecten els finals de carrera dels moviments de cada braç, i llavors, el programa d'inicialització es limita a fer girar tots els motors i parar cada motor quan es detecta el final de carrera corresponent. Així doncs, això serà el primer que farà el robot quan el connectem per primer cop, o quan sospitem que la

posició real del robot no es correspon amb la que pressuposa el *software* de control.

#### 4.2 Mode d'aprenentatge

En aquest mode el robot es mou obeint un comandament manual (generalment un *joystick* o el mateix teclat de l'ordinador), mentre per soft portem un seguiment de la posició actual. Podem optar entre dos estratègies: o bé guardem en memòria la seqüència d'impulsos de control que estem enviant a cada motor (la trajectòria), o bé únicament memoritzem la posició actual del robot en els punts de pas que ens interressi (registrem el nombre net de passos que ha girat cada motor). Aquest segon cas rep el nom de programació punt a punt, i té l'avantatge que estalvia molta memòria.

#### 4.3 Mode de reproducció

El robot ha d'ésser capaç, en aquest mode, de repetir els moviments apresos durant el mode d'aprenentatge. En el cas de programació punt a punt, és suficient enviar tants impulsos al motor com passos havíem memoritzat. El problema és que no podem conèixer a priori com serà la trajectòria entre un punt i el següent. Si volguéssim fer que les trajectòries fossin rectilínies, ho podríem fer interpolant un nombre suficientment elevat de punts

intermitjos en línia recta entre el punt inicial i el punt destí, però això requereix un tractament matemàtic més elevat, ja que per cada punt que interpolem necessitem fer conversions entre coordenades cartesianes i angulars.

L'ús de rampes d'acceleració i desacceleració en cada motor, milloren molt el funcionament i la velocitat del robot durant el mode de reproducció i disminueix el risc de perdre el coneixement de la posició actual del robot, com hem comentat en l'apartat 4.1.

### CONSIDERACIONS FINALS

L'experiència em diu que prendre's la robòtica com un *hobby*, és un bon complement a l'estudi d'una enginyeria, amb uns plans d'estudi sovint massa densos i teòrics. Construir el meu propi robot m'ha donat alguns coneixements que no es donen a l'Escola de Telecomunicació, però, sobre tot, m'ha despertat un esperit de recerca que no tenia: cada problema amb que m'anava enfrontant, m'obligava a fer una visita a la biblioteca de l'Escola per consultar llibres, o visites a fires com Robotec o Expotrònica, a distribuïdors de components electrònics, o simplement discutir certs aspectes amb veritables experts en robòtica, com en Xavi Florensa, també estudiant de l'Escola de Telecomunicació, que comparteix amb mi la passió pels robots.

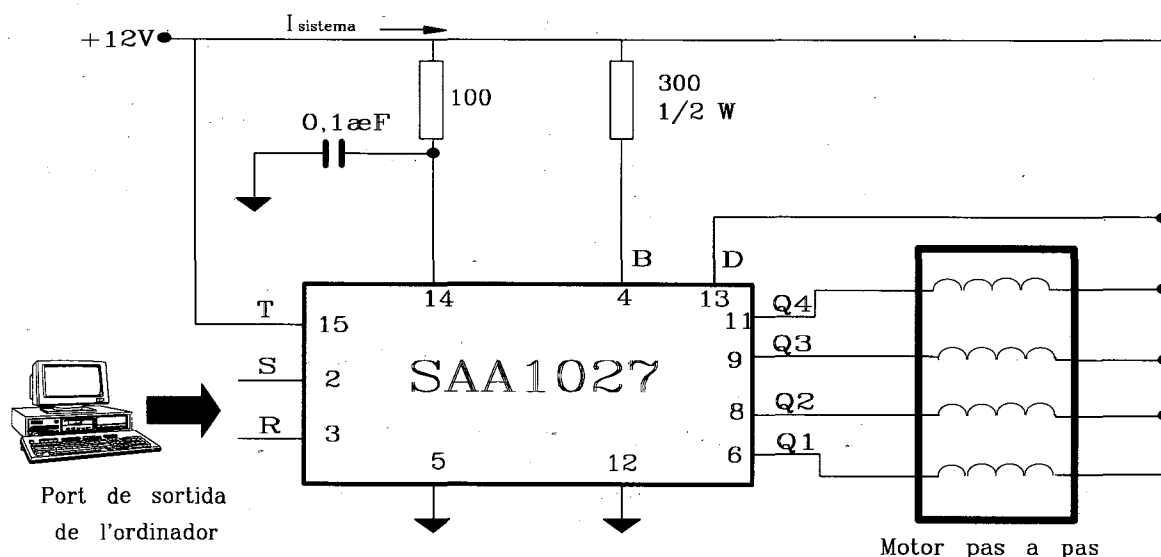


Figura 4.- Esquema de les connexions del xip SAA 1027 amb l'ordinador i amb un motor pas a pas.