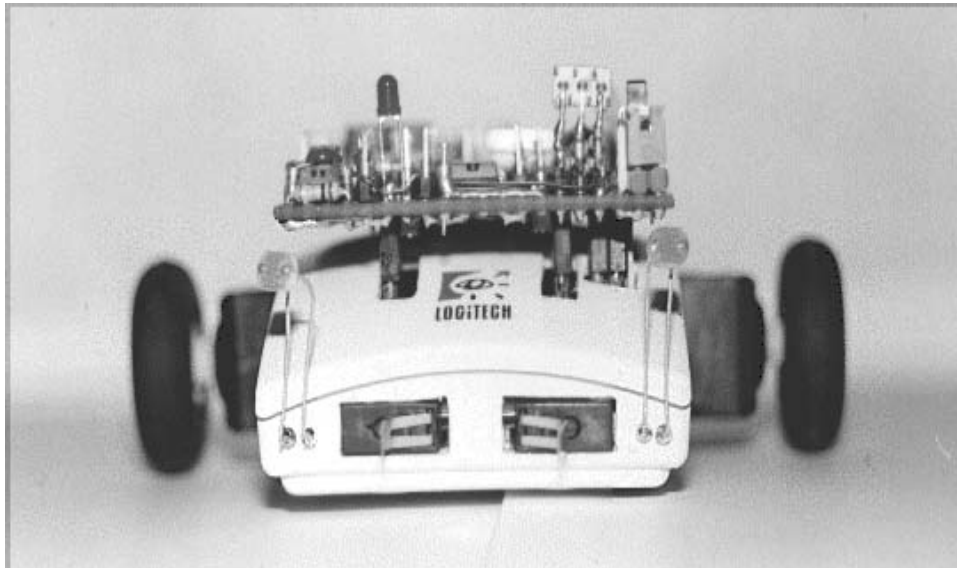

CS-EL

(CRIATURA SIMPLE, ESTIMULO LUMINOSIDAD).



Código: 0000-000-0000

Edición: 1.0

Versión: 1.0, Mayo 1996

Autor: Antonio Arnáiz Lago.

1. ÍNDICE DE CONTENIDOS

2. INTRODUCCIÓN.....	2-1
2.1 DOCUMENTOS RELACIONADOS.....	2-1
2.2 DEFINICIONES Y ACRÓNIMOS.	2-1
3. DEFINICIÓN DEL COMPORTAMIENTO.....	3-2
3.1 ESTADO NOCHE.	3-2
3.2 ESTADO DÍA.....	3-2
3.3 IMPLEMENTACIÓN DEL PROGRAMA.....	3-2
4. DESCRIPCIÓN HARDWARE.....	4-1
4.1.1 Interfaz del conversor A/D.....	4-2
4.1.2 Interfaz de control de los motores.....	4-3
4.1.3 Batería.	4-3
4.1.4 Montaje del circuito.	4-3
5. DESCRIPCIÓN SOFTWARE.....	5-1
5.1 INICIO DEL PROGRAMA.	5-1
5.2 RUTINA DE GENERACIÓN DEL COMPORTAMIENTO.....	5-2
5.3 RUTINA DE RETROCESO.....	5-3
5.4 RUTINA DE COMPROBACIÓN DEL ESTADO NOCHE. (AUSENCIA DE LUZ).	5-4
5.5 RUTINA DE COMPROBACIÓN DEL ESTADO DÍA. (EXISTENCIA DE LUZ).	5-5
5.6 RUTINAS DE ORIENTACIÓN A DERECHA E IZQUIERDA.	5-6
6. DESCRIPCIÓN MECÁNICA.....	6-1
6.1 MECANIZACIÓN DEL CUERPO DEL ROBOT.	6-2
6.1.1 Mecanización de los servos.	6-3
6.1.2 Circuito impreso interior.....	6-5
7. LISTA DE COMPONENTES.....	7-1
7.1 COMPONENTES ELECTRÓNICOS.....	7-1
7.2 COMPONENTES VARIOS.	7-2

2. INTRODUCCIÓN

El presente documento contiene la descripción de un robot que pertenece a una clase que hemos denominado **CS** (**C**riaturas **S**imples). Este tipo de Robots se caracteriza por implementar comportamientos muy básicos, que generalmente sirven de estudio para obtener información destinada a su incorporación en Robots más complejos.

El robot aquí descrito se denominará en adelante **CS-EL** (**C**riatura **S**imple, **E**stimulo **L**uz) y como su nombre indica solamente implementa un estímulo, la luz..

2.1 Documentos Relacionados.

No hay.

2.2 Definiciones y Acrónimos.

- **PC** Personal Computer.
- **CS-EL** Criatura Simple, Estimulo Luz.

3. DEFINICIÓN DEL COMPORTAMIENTO.

El **CS-EL** es un robot móvil cuyo comportamiento es el de desplazarse hacia fuentes de luz.

Como fuentes de luz válidas se entienden todas aquellas que proporcionan luz visible, tales como bombillas de alumbrado, tubos fluorescentes, luz solar y en general todas aquellas fuentes luminosas capaces de hacer variar la resistencia de una célula fotoeléctrica (resistencia LDR) convencional.

Para implementar este comportamiento, se ha equipado al robot con dos [células fotoeléctricas], una en el lado derecho y otra en el lado izquierdo, de esta forma será posible determinar mediante la comparación de las medidas obtenidas de cada una de ellas la dirección en la que se debe mover.

Los *requisitos* que deberá cumplir el comportamiento definido serán los siguientes:

- El único estímulo será la luz visible.
- El robot se desplazará siempre a la zona de mayor luminosidad.
- Una vez que se alcance una zona en la que la luminosidad supere un cierto umbral, el robot permanecerá en dicha zona hasta que se varíen las condiciones de luz en dicha zona.
- Cuando no sea posible encontrar una fuente de luz que supere un cierto valor mínimo, se considerará que se está en un estado que se denominará *noche*, en este estado, el robot deberá permanecer parado, comprobando cada cierto tiempo si ha aparecido alguna fuente de luz que supere el valor de umbral, momento en el cual se volverá a poner en actividad, pasando de este modo al estado *día*.
- Si en su desplazamiento en busca de una fuente de luz, el robot colisiona con algún objeto (este hecho se detectará al activarse alguno de los microinterruptores delanteros), el robot deberá retroceder hacia atrás un cierto espacio para posteriormente volver a ponerse en marcha en el mismo sentido. Este hecho podría hacer que el robot entre en un *bucle* de avance-retroceso, el cual se romperá en el momento que desaparezca el obstáculo o se modifiquen las condiciones de luz en el entorno.

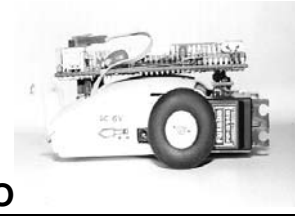
De lo expuesto anteriormente se deduce que en todo momento, el robot se puede encontrar en dos estados bien definidos que se denominarán *noche* y *día*, estos estados se describen más detalladamente en los siguientes apartados.

3.1 ESTADO NOCHE.

Tal y como se ha indicado, cuando el robot no es capaz de detectar ninguna fuente de luz a su alrededor que supere un umbral mínimo, pasará al estado denominado *noche*, en el cual permanecerá parado, activándose cada cierto tiempo, para comprobar si ha aparecido una fuente de luz, con un umbral mínimo que permita su seguimiento. Si esto es así, el robot se activará entrando en el modo de funcionamiento normal (activación del estado día).

Para implementar el estado *noche* se ha seguido la siguiente estrategia.

Si desaparece la fuente luminosa que está siguiendo, el robot girará sobre sí mismo, tratando de detectar alguna otra fuente luminosa de referencia. Si es así, abandonará el giro en dirección a la fuente de luz detectada. Si por el contrario no se detecta luz, seguirá girando sobre sí mismo durante un cierto número de vueltas más y si después de completar dichos giros no se detecta luz, el robot se parará entrando en un bucle de espera. En este bucle el robot permanece parado durante un cierto tiempo, al cabo del cual se activará, girando sobre sí mismo para buscar algún posible punto de luz, si es así abandonará el estado *noche* para pasar al estado *día*, de lo contrario volverá a realizar una pausa para seguir después con otra comprobación y así sucesivamente.



DEFINICIÓN DEL COMPORTAMIENTO

3.2 ESTADO DÍA.

Es el estado de funcionamiento normal del robot. En este estado, el robot tomará muestras de la tensión obtenida a través de las [células fotoeléctricas]. El rango de las muestras es de 0 a 255 (1 byte), siendo 0 máxima luminosidad y 255 oscuridad total. Una vez obtenida una muestra de cada célula, se realiza una operación AND lógica con un valor de máscara con el fin de determinar la sensibilidad, actualmente el valor de la máscara es de 240 (**11110000 Binario**), con lo cual se eliminan los bits de menor peso de las muestras (D0...D3), que son los bits que introducen más *ruido* en la detección. Seguidamente se comparan las dos muestras y del resultado se determina la nueva dirección. Para orientar el robot en la nueva dirección, se realiza el procedimiento siguiente:

- Se detiene el motor que corresponde al lado donde se encuentra la célula de la que se ha obtenido la muestra que indica mayor cantidad de luz. Esta muestra se guarda temporalmente como *muestra de referencia*.
- Al detener el motor anterior, el robot gira en dirección a la zona de más luz. Por ejemplo si se ha detectado más luz a la izquierda, se para el motor izquierdo, con lo cual, debido a que el motor derecho si está funcionando, el robot girará a la izquierda.
- Mientras el robot está girando, se toman muestras de la célula del lado contrario (que es donde se encuentra el motor activo), cuando las muestras tomadas coinciden con la muestra de referencia , se vuelve a activar el motor que permanecía parado, de esta forma el robot se encuentra avanzando en dirección a la nueva fuente de luz.

Este proceso se repite indefinidamente, mientras se permanece en este estado.

Cabe indicar que si durante el proceso de avance, el robot choca con algún objeto (se determina que se ha chocado con un objeto cuando alguno de los micro interruptores delanteros se activa), se procederá a invertir el sentido de la marcha durante un cierto tiempo, al cabo del cual se volverá a activar la rutina anterior.

3.3 IMPLEMENTACIÓN DEL PROGRAMA.

Para la implementación del programa se ha elegido el lenguaje de programación **PLM51**. Este lenguaje es muy similar al **C**, si bien con algunas limitaciones, no obstante es bastante válido para codificar el comportamiento deseado.

Ver el capítulo 5 para más información acerca de este tema.

4. DESCRIPCIÓN HARDWARE.

Este capítulo contiene la descripción hardware del **CS-EL**.

El diagrama que contiene el circuito de control de robot es el mostrado en la siguiente figura.

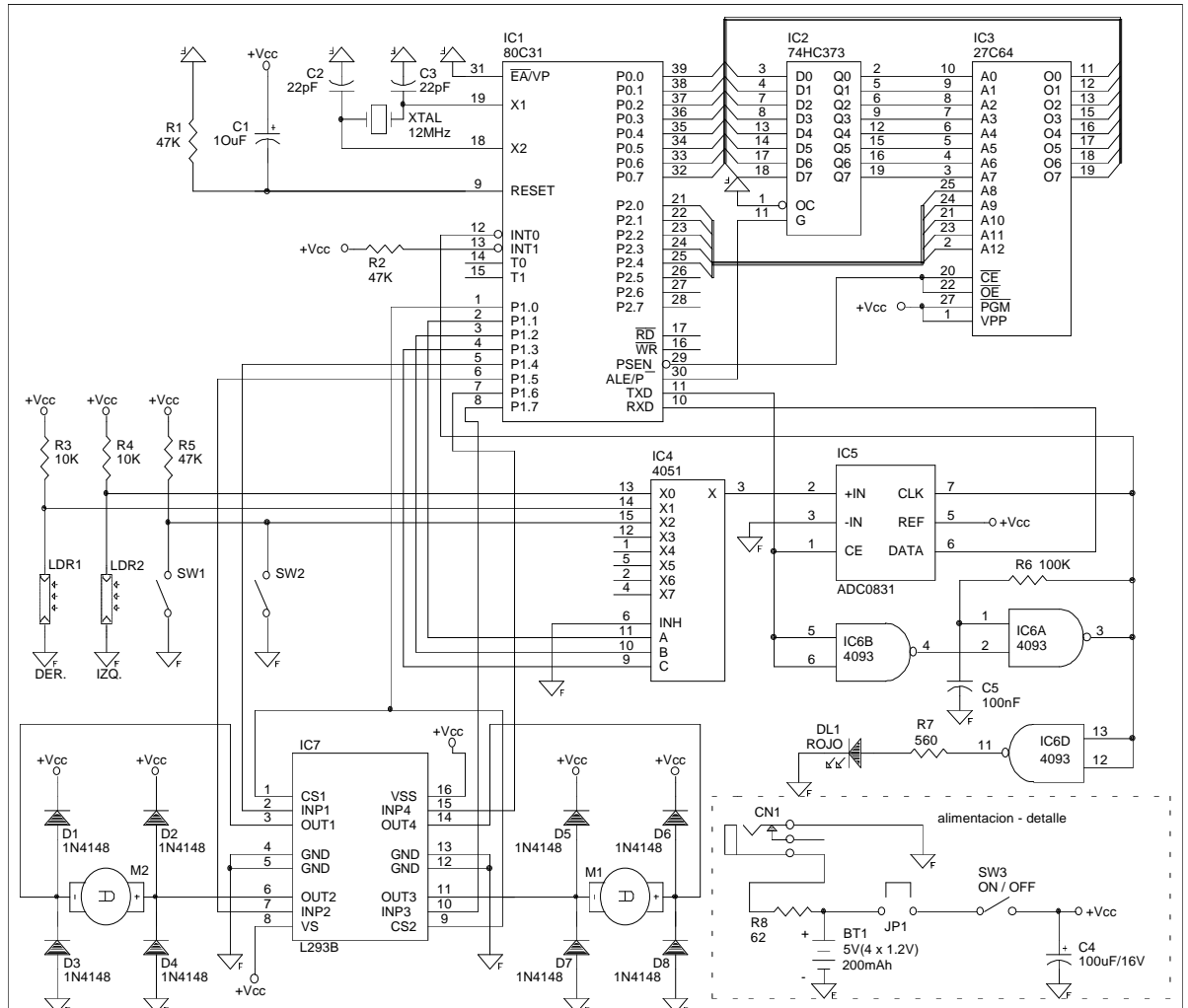
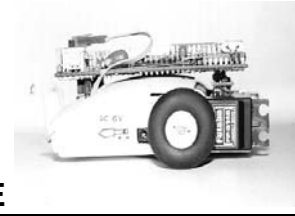


Figura 1 Esquema eléctrico del circuito de control del CS-EL.

Como puede apreciarse, el circuito está implementado entorno a un microprocesador bien conocido, el **80C31**, **IC1**. Este microprocesador proporciona capacidad más que suficiente para implementar el comportamiento que se desea realizar. **IC2** es el *Latch* típico en la arquitectura de sistemas basados en el 80C31 y permite obtener la parte baja del bus de direcciones. **IC3** es la memoria *EPROM* que contiene el programa de control del microprocesador. Se ha elegido una memoria de 8 Kbytes, la **27C64** por ser muy común, si bien el programa tan solo ocupará del orden de 2 Kbytes.

R1 y **C1** forman el circuito que proporciona el pulso de *reset* del microprocesador. Los condensadores **C2** y **C3** junto con el cristal de cuarzo **XTAL** forman el oscilador del microprocesador.

IC4 es un multiplexor analógico de 8 canales, que se utiliza para seleccionar los distintos sensores que incorpora el Robot. El canal sobre el que se desea efectuar una determinada medida, es seleccionado desde el microprocesador utilizando para ello los bits **P1.1**, **P1.2** y **P1.3** del puerto P1 del micro. Como se puede apreciar, este multiplexor tiene 8 canales, lo que permitiría equipar al Robot con 8 sensores, si bien, en la práctica solamente se han incorporado tres: las dos [celulas fotoeléctricas] con las cuales se



DESCRIPCIÓN HARDWARE

guía para buscar la luz y unos pequeños micro interruptores, colocados en la parte delantera del Robot que permiten saber si se ha chocado con algo mientras se desplaza hacia delante.

La salida del multiplexor se envía a la entrada de un conversor analógico-digital, **IC5**, éste es un conversor de 8 bits con salida serie. Se ha elegido este conversor por su facilidad de interfaz con el micro, así como por su reducido tamaño (encapsulado *Dual In Line* 8 pines), es de destacar la alta velocidad de conversión (típicamente 32 μ Seg.), si bien dada la naturaleza de la señal a medir (luz) la velocidad de conversión no es un factor determinante (podría haberse elegido otro conversor más lento).

La salida del conversor se envía al micro y éste la recibe a través de la entrada *RXD* (recepción del puerto serie), así mismo la señal *TXD* (transmisión del puerto serie) se utiliza como *Chip Select* del conversor. Esta utilización totalmente atípica del interfaz de comunicaciones serie del microprocesador es, sin embargo perfectamente válida dado que todos los bits de los puertos del micro se pueden gobernar directamente, tanto en modo lectura como en escritura. Como en este caso no se utiliza el interfaz de comunicaciones y con el fin de no aumentar la tarjeta de control con más circuitos se opta por utilizar estos dos bits como se ha indicado.

No es preciso realizar ningún control adicional sobre el puerto serie debido a que tal y como se ha indicado, el puerto no se utiliza como tal y además su interrupción estará inhibida debido a que el sistema interno del microprocesador inhibe todas las interrupciones después del *reset* (ver información del 8031 para más aclaración).

El generador de reloj del conversor A/D está implementado utilizando dos puertas *NAND* **IC6A** y **IC6B**, la frecuencia del reloj se determina mediante la resistencia **R6** y el condensador **C5**.

La puerta **IC6D** gobierna un diodo led que permite monitorizar el estado del conversor (encendido indica que el microprocesador está obteniendo muestras desde el conversor).

Como se puede apreciar en el esquema, las dos células fotoeléctricas, **LDR1** y **LDR2** están colocadas de la misma manera, ambas forman divisor de tensión con sus correspondientes resistencias **R3** y **R4**. Esto permite obtener en cada una de ellas una tensión que será proporcional a la luz que recibe cada célula, siendo esta tensión **menor** cuanto **más** luz reciba la célula en cuestión.

La tensión obtenida en cada uno de los divisores se envía a los canales de entrada del multiplexor **IC4**, (célula del lado izquierdo por el canal **0** y célula del lado derecho por el canal **1**).

En cuanto a los dos micro interruptores **SW1** y **SW2**, ambos están colocados en paralelo y a través de la resistencia **R5** permiten obtener un señal de 5 voltios cuando **no** están pulsados y 0 voltios cuando alguno de ellos está pulsado (típicamente cuando el Robot choca con algo). Esta tensión se envía al canal **2** del multiplexor.

4.1.1 Interfaz del conversor A/D.

El interfaz implementado para la lectura y control del conversor A/D desde el microprocesador, es el siguiente:

Después de activar el conversor para que dé comienzo una conversión (colocando para ello *TXD* a **0**, puesto que normalmente se encuentra a **1**, inhibiendo de esta forma el conversor), los bits que forman el *byte* resultante de la conversión van apareciendo en la salida del conversor (pin 6), sincronizadamente con los impulsos del generador de reloj del conversor, estos impulsos de reloj se envían además de al conversor para efectúe la conversión, a la interrupción *INT0* del microprocesador. De esta forma cada interrupción producida indica al microprocesador que el correspondiente bit del *byte* que se está obteniendo se encuentra disponible en *RXD*.

Es de destacar que el conversor genera **9** bits por cada muestra (ver datos del fabricante para más información al respecto), debiendo despreciarse el primero de ellos, que actúa a modo de bit de *start*.

Cuando se han obtenido los 9 bits, el microprocesador colocará a **1** el bit *TXD* inhibiendo de esta forma el conversor.



DESCRIPCIÓN HARDWARE

4.1.2 Interfaz de control de los motores.

Para el control de los motores se ha elegido un circuito integrado que permite el control de dos motores de corriente continua, de forma totalmente independiente uno del otro y que además no requiere ningún componente adicional. Estas características unidas al hecho de que dicho circuito es fácil de encontrar y barato (¿ hay alguien que dé más ?) lo hacen idóneo para esta aplicación. Se trata del **L293B** , **IC7**, que tal y como se puede apreciar en el esquema, recibe el control desde los bits del puerto **P1** del microprocesador.

El estado de los motores puede resumirse en la siguiente tabla:

<i>Bits (*)</i>					<i>Motor Derecho</i>	<i>Motor Izquierdo</i>
P1.0	P1.4	P1.5	P1.6	P1.7		
0	--	--	--	--	Parado	Parado
1	0	1	--	--	Delante	--
1	1	0	--	--	Detrás	--
1	--	--	0	1	--	Delante
1	--	--	1	0	--	Detrás

Tabla 1. Posibles estados de los bits de control de los motores.

(*) *Solamente se representan los estados válidos, los demás no se permiten desde el microprocesador.*

4.1.3 Batería.

Para la batería se han utilizado 4 baterías de 1.2 voltios y 200mAh en serie. En la lista de componentes se da la referencia del fabricante utilizado, pero al ser un tipo de batería estándar puede utilizarse de cualquier otro fabricante.

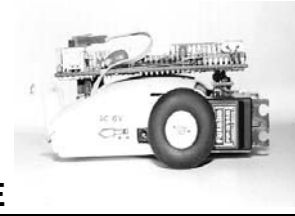
El interruptor SW3 permite conectar o desconectar el robot (ON/OFF), mientras que el puente JP se ha colocado a modo de test para comprobar el consumo del robot. Para realizar esta comprobación bastará quitar momentáneamente dicho puente colocando en su lugar un miliamperímetro. Finalmente, el conector CN1 permitirá efectuar la recarga de la batería desde el exterior, para ello se deberá suministrar una tensión de **6V / 100mA**, de corriente continua respetando siempre la polaridad de la conexión.

4.1.4 Montaje del circuito.

Para el montaje de la tarjeta de control se ha optado por la técnica de *wire wrapping*. Esta técnica se ha elegido teniendo en consideración los siguientes factores:

- Se trata de un prototipo, con lo cual es fácil imaginar que el circuito sufrirá mejoras y modificaciones durante la fase de pruebas, si se procediera a montar sobre un circuito impreso clásico, esto implicaría cortes y modificaciones en las pistas de dicho circuito e incluso la adición de otros componentes , hecho que normalmente es difícil y en algunos casos imposible de llevar a cabo, lo que conllevaría a la realización de nuevos circuitos impresos.
- Dado que sólo se va a construir un Robot, el coste que conllevaría realizar el circuito impreso sería elevado, debido a que al ser de reducido tamaño y gran cantidad de componentes sería preciso un circuito impreso de, como mínimo dos caras, taladro metalizado etc.

Como se puede apreciar en las imagenes, los circuitos integrados no se han colocado sobre zócalos, salvo la memoria EPROM y el driver de los motores, esto es debido a que se ha intentado reducir al mínimo el número de componentes, así como el peso y el espacio adicional de los mismos, aunque en honor a la verdad se puede afirmar sin temor que unos cuantos zócalos no suponen tanto problema.



DESCRIPCIÓN HARDWARE

Por razones funcionales, el circuito que sirve de driver de control de los motores (**L293B**) se ha montado en el circuito impreso de la parte interior del robot, **no** en la tarjeta de control. Por otra parte y tal y como se puede apreciar en la Figura 14. Aspecto interior del robot. existen dos hileras de pines de 4 pines más otra tercera hilera de dos pines, las tres situadas alrededor de dicho circuito integrado. La misión de estas tres hileras es la de conectar las distintas señales de control del microprocesador con el circuito L293B y también conectar los micro interruptores de la tarjeta interior con el microprocesador. La tercera hilera (de dos pines) contiene la alimentación que proviene de la batería y que permite alimentar la tarjeta de control.

Finalmente, se deberán soldar en la parte inferior de la tarjeta unos pines *hembra*, a los cuales se conectarán los pines *macho* que provienen de la tarjeta inferior, situada en el chásis del robot (alojamiento de los motores). De esta forma la tarjeta de control será totalmente enchufable al robot, pudiendo retirarse o sustituirse por otra en cualquier momento.

La siguiente imagen muestra el aspecto final de la tarjeta de control.

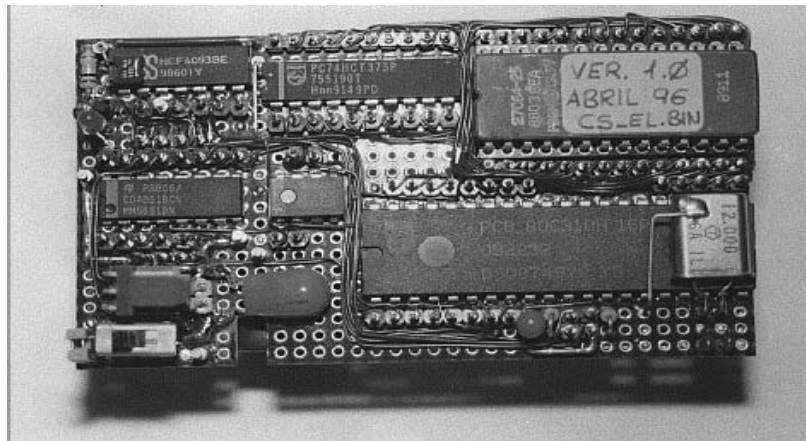
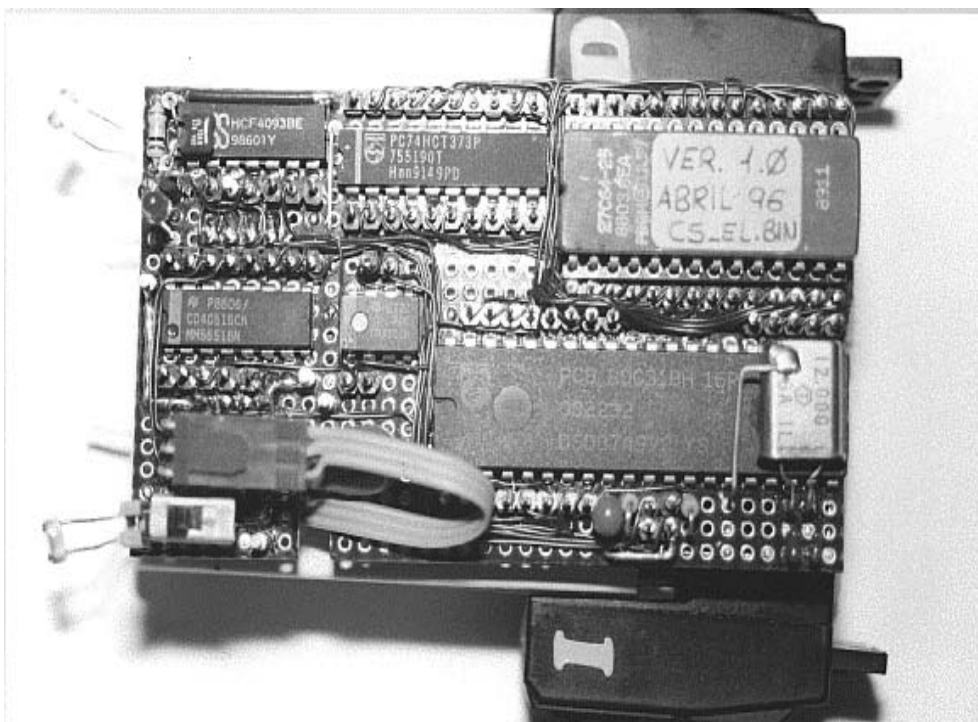
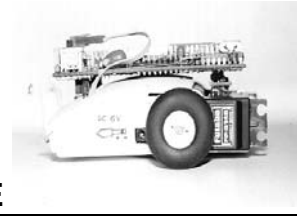


Figura 2. Aspecto de la tarjeta de control.





DESCRIPCIÓN HARDWARE

Figura 3. Aspecto de la tarjeta de control ya montada sobre el chásis.

Tal y como se aprecia en las dos fotografías anteriores, existe un pequeño conector de tres pines que se conecta con la parte inferior mediante un cable plano, este conector contiene las conexiones de las dos células fotoeléctricas, esto es así solamente debido a un requisito de diseño ya que, al estar situadas las dos células en el chásis del robot, las mismas se podían haber conectado a la tarjeta de control añadiendo simplemente algunos pines de control más, pero debido al reducido espacio se ha optado por conectarlas mediante este cable.

5. DESCRIPCIÓN SOFTWARE.

El presente capítulo describe El software de control del robot.

El software se ha realizado en un lenguaje denominado **PLM-51**. **PLM51** es un lenguaje creado por **INTEL** en 1986 y en la actualidad está obsoleto, no obstante, debido a su buena optimización, es aún perfectamente válido para desarrollar aplicaciones que tengan como destino final sistemas basados en **8031/32/51/52** y en **8751/52**, así como en sus respectivas variantes.

Como principal característica del **PLM51**, cabe destacar que es un lenguaje muy similar al lenguaje **C**, pero menos evolucionado.

En este capítulo se describirán mediante diagramas de flujo los principales procedimientos llevados a cabo para efectuar el control del robot. El resto de las rutinas que forma el programa, debido a su simplicidad no se ha considerado pertinente su inclusión en este capítulo, por lo que la utilidad de las mismas puede ser consultada directamente en los ficheros fuente.

Los ficheros, en los que como se ha dicho se encuentra implementado el código descrito aquí, así como el programa resultado son los siguientes:

- **CS_EL1.PL**. Este fichero contiene la inicialización del sistema, así como el bucle de arranque del programa.
- **CS_EL2.PL**. Este fichero contiene todas las rutinas de tipo general utilizadas en el programa.
- **CS_EL3.PL**. Este fichero contiene rutinas de control y la rutina que genera el comportamiento del robot.
- **CS_EL.DCL**. Este es el fichero con todas las declaraciones, definiciones etc del programa, es pues similar a un fichero de tipo *include* de **C**.
- **CS_EL.BIN**. Es el fichero que contiene el programa, resultado de compilar los anteriores ficheros fuentes, es decir, el *ejecutable* que se graba en la memoria EPROM.
Este fichero mide 1280 bytes y se debe grabar en la memoria EPROM a partir de la **primera posición** de la misma (dirección 0).

5.1 Inicio del Programa.

Inmediatamente después de activar el robot (*power ON*), se produce el impulso de reset, el cual inicializa el microprocesador. Al desaparecer el pulso de reset, el microprocesador comienza a ejecutar el código contenido en la memoria EPROM, desde la posición **0** de la misma.

La primera operación que se lleva a cabo es la inicializar todas las variables de control, así como los distintos elementos hardware existentes. Tal y como se puede apreciar en el diagrama eléctrico, todos los elementos hardware que actúan como *periféricos*, se controlan mediante los bits del puerto **P1**, por lo cual la primera tarea será la de inicializar los bits de este puerto. Seguidamente se efectúa una pequeña pausa con el fin de estabilizar todo el sistema (eléctricamente) y seguidamente se activa el movimiento del robot en sentido *hacia* delante. Finalmente se pasa a ejecutar un bucle indefinido (sin fin) que es el que ejecuta el comportamiento del robot mientras está activo, en este bucle también se actualizan 8 variables que contienen el valor que aparece en todo momento en los canales de entrada del multiplexor analógico que está conectado a la entrada del conversor A/D. Estas variables son globales a todo el programa por lo que pueden ser consultadas desde cualquier punto del mismo. Se han implementado 8 variables debido a que es el número máximo de sensores que podría tener el robot (uno por cada canal del multiplexor), si bien sólo se utilizan las tres que corresponden a los tres sensores implementados (las dos células y los micro switches delanteros). Independientemente de estas tres variables, los canales del multiplexor a través de los cuales se lee el valor de las células fotoeléctricas y los micro switches delanteros son consultados directamente desde diversas rutinas del programa, cuando es preciso.

El siguiente diagrama muestra el procedimiento de arranque:



DESCRIPCIÓN SOFTWARE

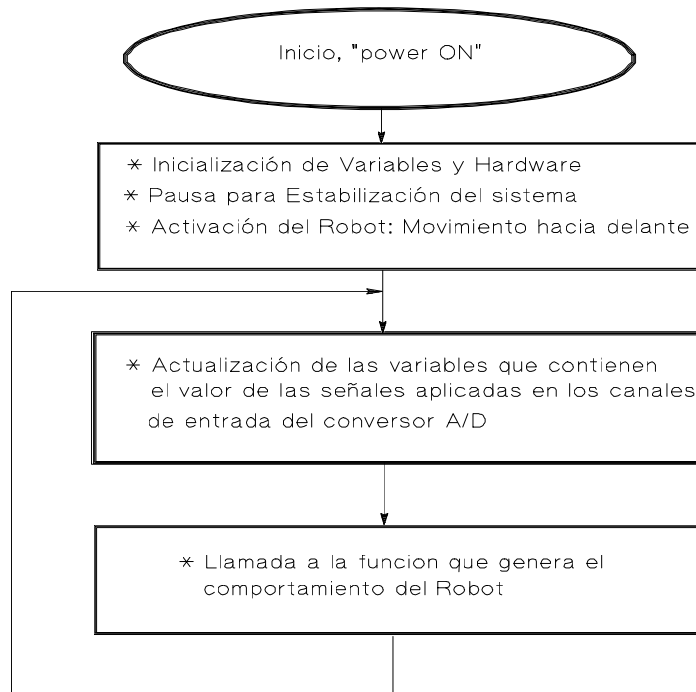


Figura 4. Diagrama de flujo del proceso de inicialización y ejecución del programa.

Todo este proceso se lleva acabo íntegramente en el código contenido en el fichero **CS_EL1.PLM**.

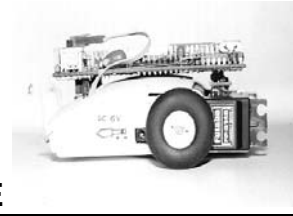
5.2 Rutina de generación del Comportamiento.

Esta es la rutina más importante del programa, puesto que es la encargada de generar el comportamiento que sigue el robot.

La rutina es llamada desde el bucle general de ejecución (que es el que mantiene activo el programa) y efectúa las siguientes operaciones:

- Comprueba si hay luz suficiente, en cuyo caso evalúa la zona donde hay mayor luminosidad y orienta el avance del robot hacia dicha zona.
- Si no es posible encontrar luz suficiente, pasa a modo *NOCHE*. En este modo, el robot permanece parado efectuando cada cierto tiempo una comprobación para ver si ha aparecido de nuevo luz.
- Si el robot se encuentra en modo *NOCHE* y aparece de nuevo luz en alguna zona, se conmuta al estado *DÍA*, volviendo a efectuarse la tarea de evaluación y orientación.

El diagrama de flujo que ilustra este comportamiento es el mostrado seguidamente:



DESCRIPCIÓN SOFTWARE



Figura 5. Diagrama de flujo de la rutina de generación de comportamiento.

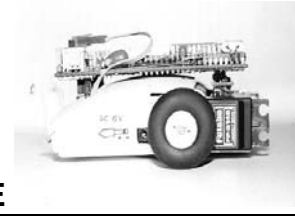
La rutina que genera el comportamiento se llama *genera_comportamiento* y se encuentra en el fichero CS_EL3.PLM.

5.3 Rutina de retroceso.

Esta rutina es llamada cuando se activa alguno de los dos micro switches delanteros y se utiliza durante el desplazamiento, en los giros etc del robot, para prevenir que durante los mismos, se choque frontalmente con algún objeto.

Cuando el programa detecta que alguno de los micro switches delanteros ha sido pulsado llama a esta rutina, la cual coloca los motores en movimiento hacia atrás durante un cierto tiempo para después volver a poner el movimiento hacia delante. Finalmente devuelve el control a la rutina que lo llamó.

Con esta forma de proceder, es posible que el robot vuelva a chocar de nuevo con el objeto (sobre todo en el caso de objetos fijos, que no se mueven), pero debido a numerosos factores que también interaccionan simultáneamente, tales como pequeños desvíos del robot al mover hacia detrás, ligeras modificaciones en la cantidad de luz detectada debido a la nueva posición del robot etc, es probable que el robot termine por modificar su curso, en cualquier caso se puede comprender que este robot no tiene



DESCRIPCIÓN SOFTWARE

suficientes recursos como para efectuar un buen esquivamiento de obstáculos, por lo que se ha decidido que este comportamiento puede ser satisfactorio, en este caso

El diagrama de flujo correspondiente a esta rutina es el siguiente:

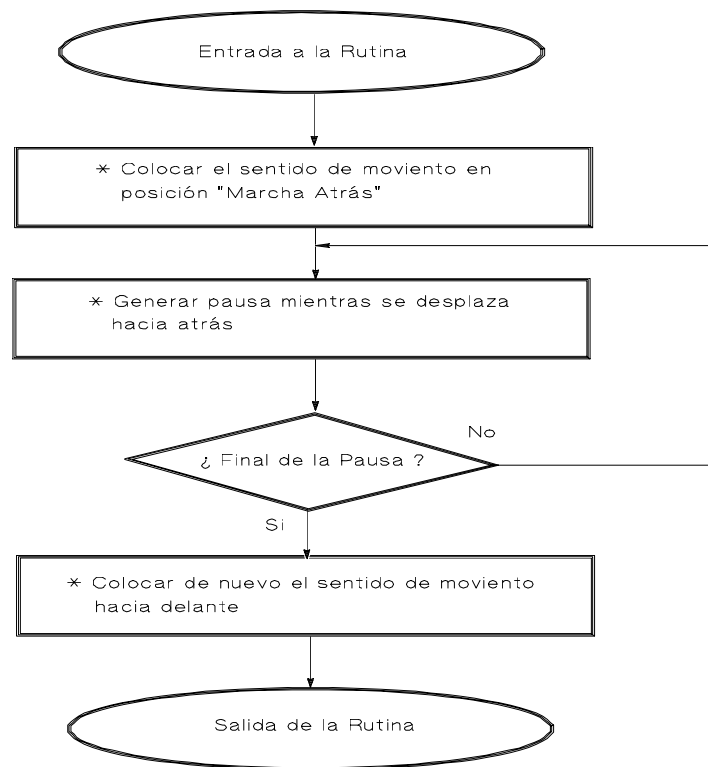


Figura 6. Diagrama de flujo de la rutina de retroceso.

La rutina que efectúa esta operación se llama *objeto_delante* y se encuentra en el fichero **CS_EL3.PLM**.

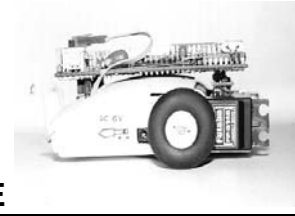
5.4 Rutina de comprobación del estado NOCHE. (Ausencia de Luz).

Esta rutina se utiliza para detectar la **ausencia** de luz alrededor del robot.

La rutina hace que el robot gire sobre sí mismo durante un cierto tiempo, durante el cual, se efectúan continuas mediciones. Si en una de las medidas se detecta que **alguna** de las células recibe una intensidad luminosa por encima del umbral considerado como oscuridad, esto indica que en dicha dirección hay luz. En este caso se abandonará la función con el resultado de error, que indica existencia de luz. Al salir de la función, el robot estará moviéndose hacia la zona donde se detectó luminosidad.

El correspondiente diagrama de flujo es el mostrado seguidamente.

Esta rutina se llama *es_noche* y se encuentra implementada en el fichero **CS_EL3.PLM**.



DESCRIPCIÓN SOFTWARE

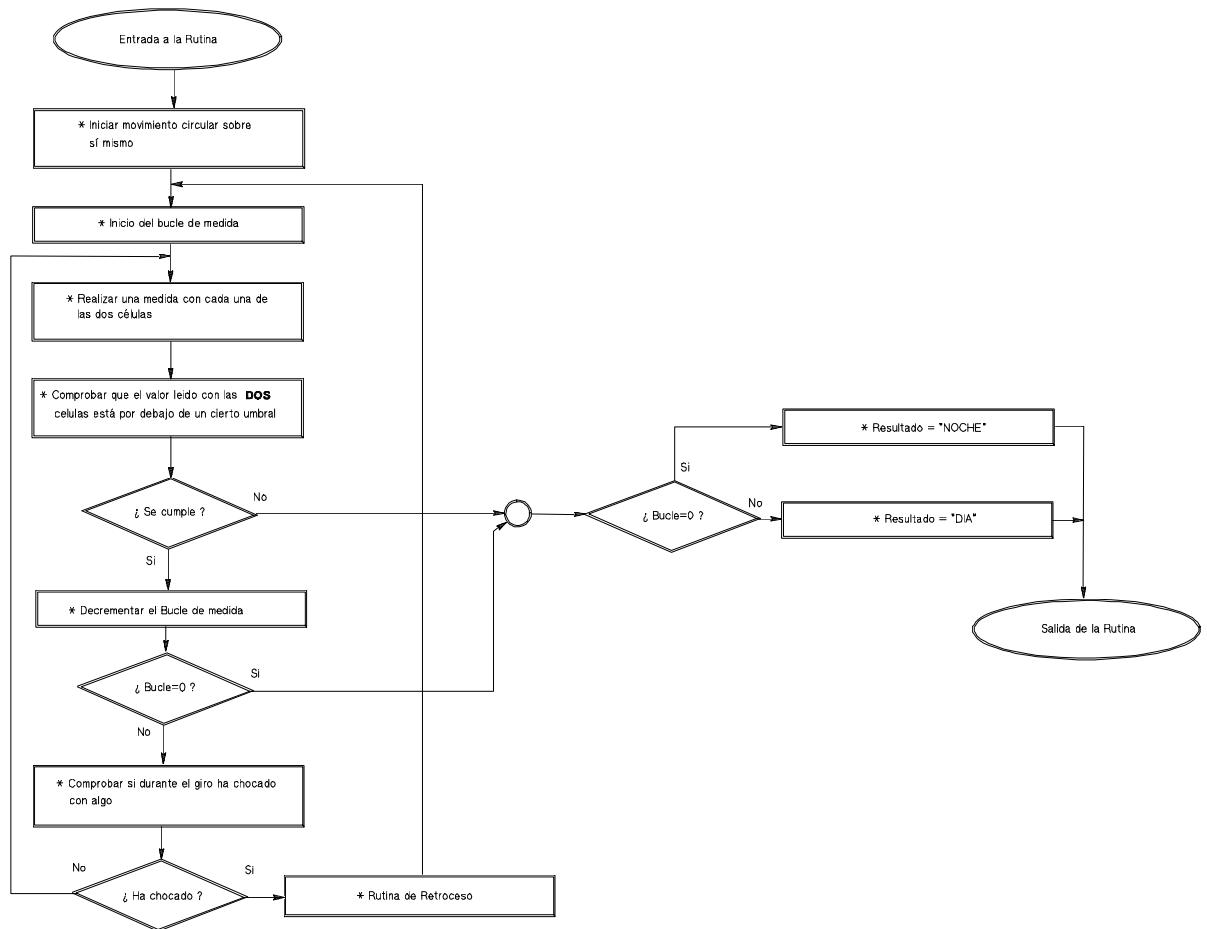
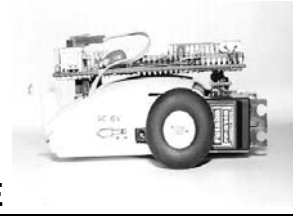


Figura 7. Diagrama de flujo de la rutina de detección de ausencia de Luz.

5.5 Rutina de comprobación del estado DÍA. (Existencia de Luz).

Esta rutina comprueba si existe luz.. Es utilizada para pasar del estado NOCHE al estado DÍA.Cuando el robot se encuentra en el estado NOCHE (es decir, parado y realizando pausas intermitentes) se llama a esta rutina para comprobar si ha aparecido luz en alguna parte y de este modo volver a la *actividad*. Al entrar en la rutina el robot se pone en movimiento, en giro circular sobre sí mismo, al tiempo que se realizan continuas mediciones de luz. Al mismo tiempo se arranca un timer de control general de todo el proceso. Si en algún momento durante los giros que se están efectuando, se detecta luz en alguna dirección, el robot se detendrá inmediatamente comprobando durante un cierto tiempo si la fuente de luz que se está detectando permanece constante, si después de transcurrido un pequeño espacio de tiempo dicha fuente de luz es constante, el robot abandonará la rutina con el resultado de *OK* y con movimiento en dirección a la fuente de luz detectada, de lo contrario el robot volverá a reanudar los giros circulares, buscando otra fuente de luz. .Si finalmente no se encuentra luz y se llega al final del temporizador de control, se abandonará la rutina con el resultado de *error* y el robot permanecerá parado.



DESCRIPCIÓN SOFTWARE

Es de destacar que, si mientras el robot está describiendo los giros, choca con algo (los micro switches delanteros están a 0) se efectuará una llamada a la rutina de retroceso, volviéndose a inicializar el timer de control de la rutina.

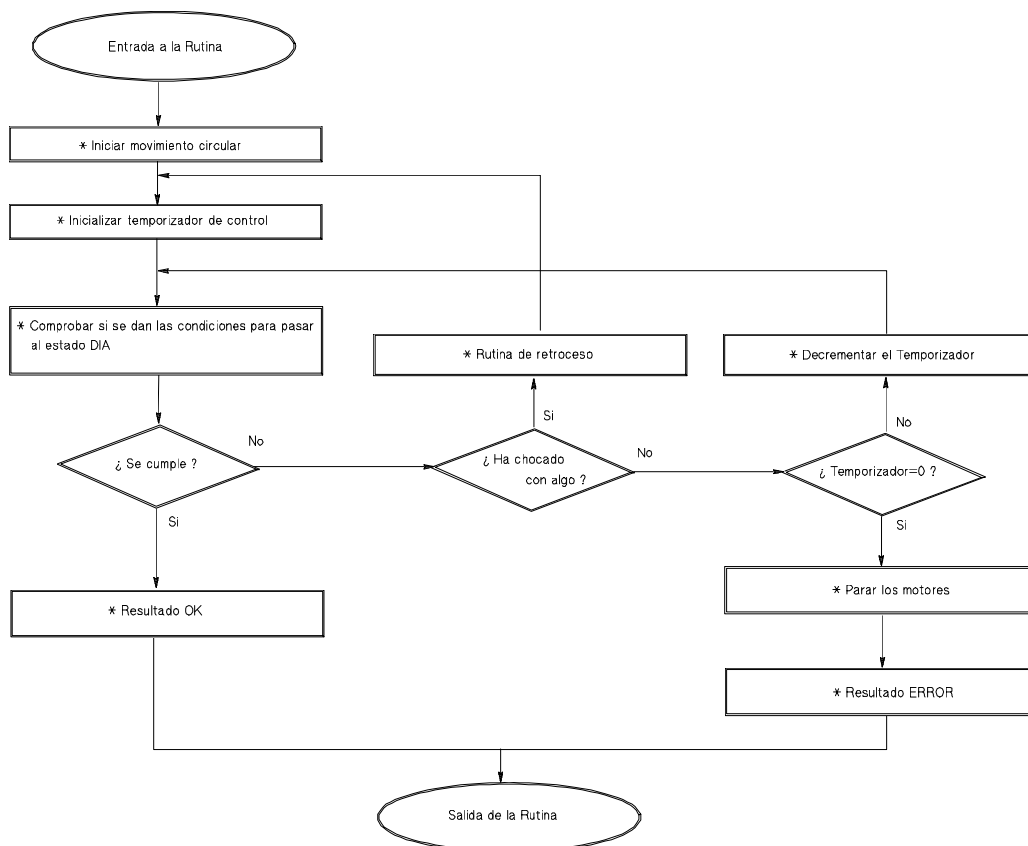


Figura 8. Diagrama de flujo de la rutina de reactivación (paso al estado DIA).

Esta rutina se llama *es_dia* y se encuentra implementada en el fichero **CS_EL3.PLM**

5.6 Rutinas de orientación a Derecha e Izquierda.

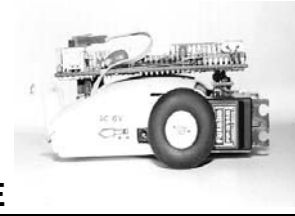
Estas rutinas son utilizadas para modificar la dirección en el desplazamiento del robot dependiendo de la zona luminosa a la que se desea dirigir el mismo.

Previamente a la llamada a alguna de estas rutinas, el programa habrá determinado la zona donde se encuentra la mayor cantidad de luz, seguidamente se llamará a una de estas rutinas para modificar la dirección del movimiento del robot.

Ambas rutinas son iguales por lo que se describirá solamente la de orientación a la izquierda.

Al entrar a la rutina, se efectúan las siguientes operaciones:

- Se modifica el sentido de avance del robot, dejándolo en situación de giro permanente a la izquierda.
- Se inicializa un temporizador de protección.
- Se arranca un bucle de medida y control general.



DESCRIPCIÓN SOFTWARE

En el bucle medición, se compara el valor de las muestras que se toman constantemente con la célula izquierda, con la **última muestra** tomada desde la célula derecha al entrar a la rutina.

Mientras el valor leído con la célula izquierda está por debajo, el bucle continúa, al llegar a ser igual o ser mayor el de la célula izquierda, se abandona el bucle, modificando de nuevo el sentido del movimiento del robot colocandolo en sentido de avance hacia delante.

Paralelamente a estas acciones y dentro del mismo bucle se decrementa el temporizador de protección, si dicho temporizador llega a 0 y sin embargo todavía no se ha conseguido que las muestras tomadas desde la célula izquierda superen o igualen el valor de la célula derecha, se considerará que existe algún tipo de problema. En este caso se abandonará el bucle, dejando también el sentido de avance hacia delante, pero indicando que se ha producido error.

El valor del temporizador permitirá no obstante que el robot ejecute al menos un par de vueltas sobre sí mismo cuando está tratando de orientarse.

El diagrama de flujo correspondiente a esta rutina es el mostrado a continuación.

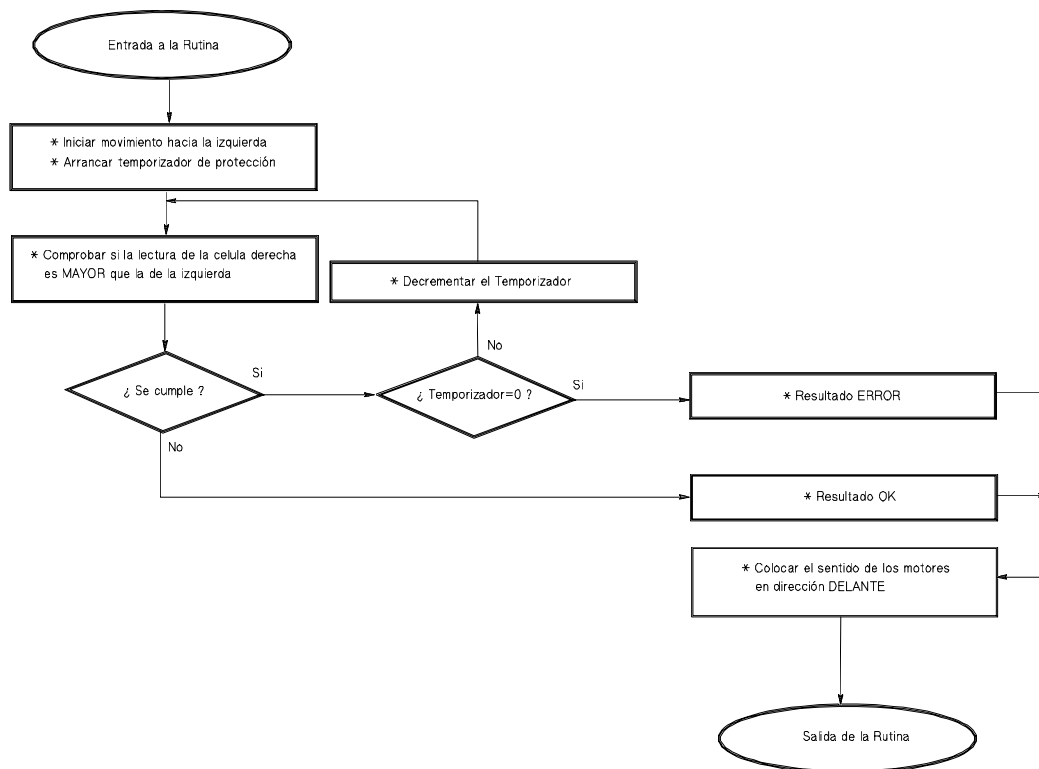


Figura 9. Diagrama de flujo correspondiente a la rutina de orientación a la izquierda.

6. DESCRIPCIÓN MECÁNICA.

Antes de comenzar a entrar en detalles descriptivos, es conveniente destacar los siguiente:

Se asume que el lector está familiarizado con los distintos procedimientos de mecanizado y que se cuenta con una cierta experiencia en este tipo de cuestiones de montaje. El autor ha querido en este capítulo dar a conocer al lector mediante el contenido del mismo y con las fotografías aportadas, datos acerca del *aspecto* final del mismo , así como de ciertos detalles de la mecanización mismo, entendiendo que entrar en detalles puntuales de su construcción sería largo y tedioso. Con este capítulo se pretende pues, que el lector pueda extraer sus propias conclusiones, conclusiones e ideas que tal vez puedan serle útiles en sus propios desarrollos.

Para el diseño mecánico (*físico*) del robot se ha partido como base principal de un elemento fácil de conseguir a la vez que curioso: un ratón (*mouse*) de PC, concretamente el modelo denominado **Basic Mouse**, de la firma LOGITECH, cuyo aspecto se puede apreciar en la siguiente imagen:

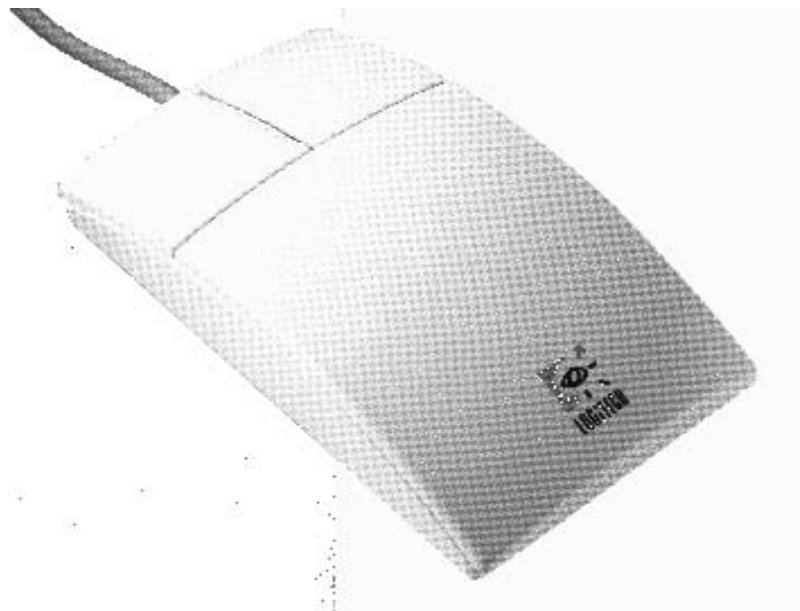


Figura 10. Ratón utilizado para la base del Robot.

Esta base se ha utilizado debido a que presenta las siguientes características:

- Su estructura se adapta correctamente a los motores y baterías que se van a utilizar.
- Es de pequeño tamaño.
- Fácil de mecanizar.
- El contenido electrónico del mismo puede ser *reutilizado*, parte en el propio Robot (micro interruptores) y parte en futuros diseños (como por ejemplo la mecánica de los optoacopladores que contiene).
- Aspecto agradable.

Los motores utilizados son, en realidad dos *servos* de los utilizados comúnmente en modelismo, concretamente el modelo **FP-S148** de la firma FUTABA. Estos servos presentan ventajas sobre los motorcillos eléctricos convencionales:

- Consumo muy reducido (entorno a los 50mA).
- Incorporan una mecánica muy optimizada, que permite reducir la velocidad del motor , incrementando notablemente la fuerza (par) en el eje de salida, que es el que gobierna la rueda de tracción correspondiente.



DESCRIPCIÓN MECÁNICA

Como principal desventaja sobre los motores clásicos destacan:

- Elevado precio (el modelo de servo utilizado se encuentra entorno a las 5000 pesetas).
- Requieren una mecanización adicional.

En los siguientes apartados se describen los diferentes pasos seguidos en la construcción. Estos pasos se detallan someramente, entendiéndose que el lector es capaz de llevar a cabo por sí mismo los diversos procesos y aspectos de la mecanización que, por razón de espacio no se detallan en el documento.

6.1 Mecanización del cuerpo del robot.

Tal y como se comentó al principio, se ha utilizado para el cuerpo del robot, un ratón de PC. El primer paso será pues, mecanizar dicho ratón para adaptarlo a los componentes que deberá alojar.

Para ello se deberán realizar las siguientes operaciones:

- Eliminar el circuito impreso que contiene el ratón.
De este circuito se deberán desoldar los dos micro interruptores que están asociados a los pulsadores del mismo. Dichos microinterruptores se colocarán posteriormente en la parte delantera para detectar los posibles obstáculos.
- Eliminar la tapa inferior que permite acceder a la bola sobre la que se desplaza el ratón. Esta tapa ya no será necesaria (ni tampoco la bola), puesto que en su lugar irá colcada la rueda que permite girar y dar estabilidad al robot.
- Seguidamente se deberá mecanizar la parte trasera del ratón que es donde se alojarán los dos servos que van a permitir desplazarse al ratón. Una idea de como deberá quedar esta zona se puede obtener viendo la **Figura 14. Aspecto interior del robot.**
- Es preciso mecanizar unas pequeñas ranuras en la tapa del ratón por las cuales se sacarán los pines de control que provienen de la tarjeta interior y se conectarán con la tarjeta de control, el aspecto de las aberturas que se realizarán se puede ver en la siguiente fotografía.

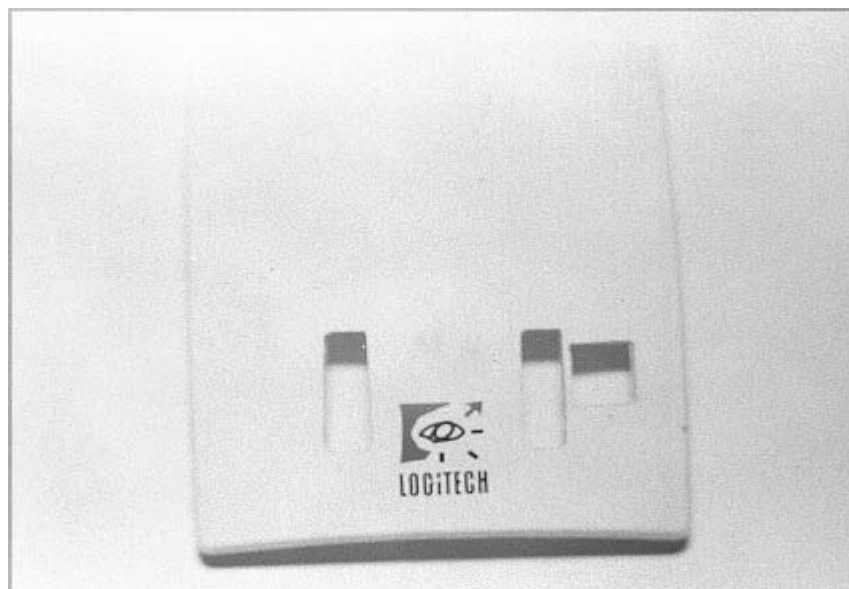
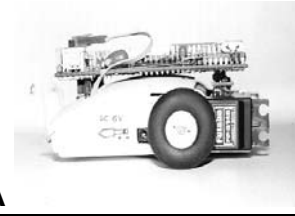


Figura 11. Mecanización de la tapa conteniendo las ranuras de acceso a los pines de control.



DESCRIPCIÓN MECÁNICA

- Ahora se deberán mecanizar los dos servos que se van a utilizar para propulsar el robot. Este es un proceso delicado y es recomendable seguir los pasos descritos en el apartado **6.1.1**, aunque el lector siempre puede realizar esta mecanización siguiendo los dictados de su propia experiencia y habilidad:

6.1.1 Mecanización de los servos.

Los pasos a seguir para la mecanización de los servos son los siguientes:

- Eliminar la tapa inferior del servo retirando para ello los cuatro tornillos inferiores.
- Desoldar el circuito impreso que contiene el servo. Este circuito actúa de controlador del motor. Los puntos que unen el circuito con los elementos del servo se detallan en el siguiente diagrama:

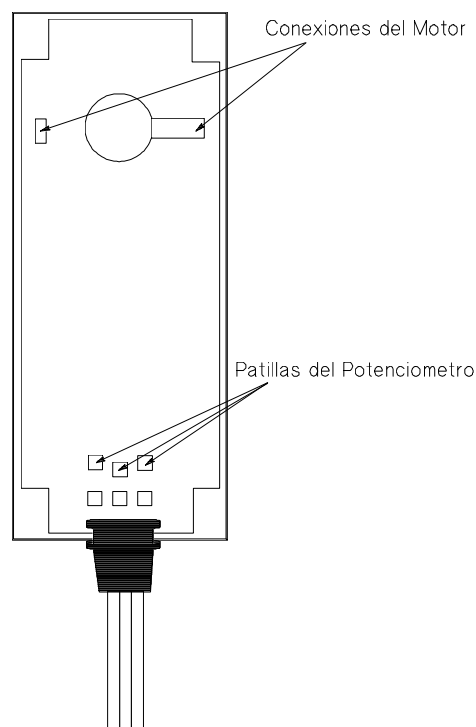
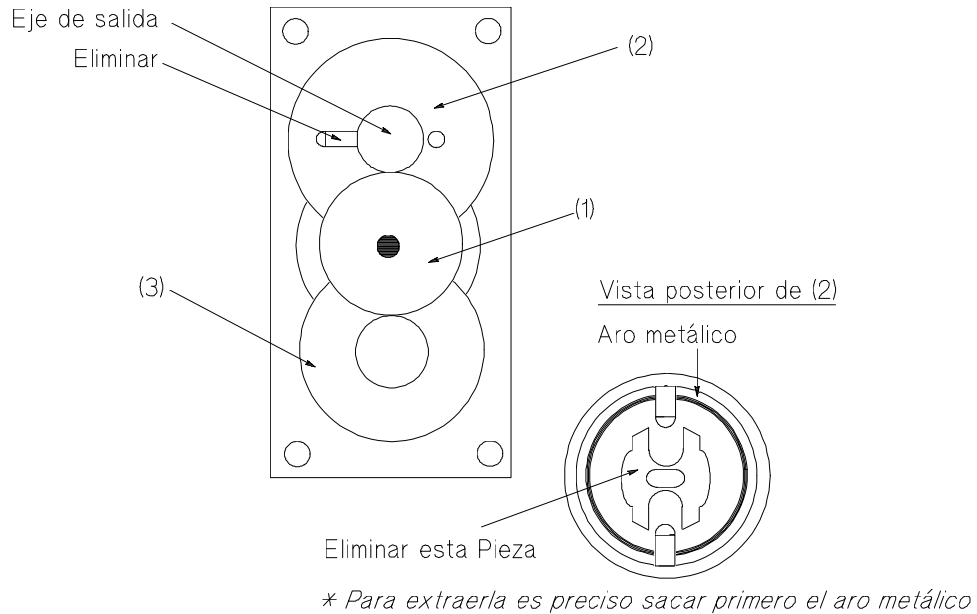
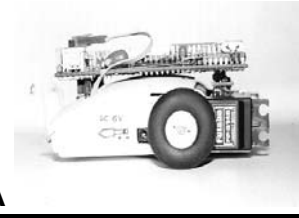


Figura 12. Puntos de unión del circuito de control con los elementos del servo.

- Una vez eliminado el circuito de control se procederá a modificar la parte mecánica, para ello se deberá separar la parte superior del servo que contiene las ruedas dentadas que efectúan la reducción de velocidad. Una vez que se tiene acceso a esta parte del servo se deberá eliminar un pequeño saliente situado en la rueda dentada que tiene asociado el eje de salida del servo. Este saliente es el que impide al eje del servo un giro de 360 grados, que es útil en modo servo, pero que en esta aplicación es imprescindible eliminar para que el eje pueda girar libremente. En la siguiente figura se detalla esta operación. Así mismo se deberá eliminar el pequeño potenciómetro que se encuentra en el interior del servo, y la pieza que une el eje de este potenciómetro con las ruedas dentadas superiores. Esta pieza está también indicada en la figura.

DESCRIPCIÓN MECÁNICA



- (2) Engranaje que lleva asociado el eje de salida del Servo
- (1) Engranaje auxiliar que prolonga el movimiento, desde el engranaje motor (3) al engranaje de salida (2)
- (3) Engranaje que proviene del motor

Figura 13. Modificación de la parte mecánica de los servos.

Una vez realizadas estas operaciones satisfactoriamente, sólo queda *serrar* la parte de la carcasa del servo que ocupaba el potenciómetro para de esta manera poder acoplarlo a la zona trasera de la carcasa del ratón que se habrá mecanizado anteriormente.

Finalmente el servo se atornillará a la carcasa del ratón utilizando para ello la parte que contiene los espacios para tornillos del mismo. Como es lógico se deberán realizar dos pequeños taladros en la carcasa del ratón para poder pasar los tornillos desde el servo a la carcasa.

El resultado final se puede apreciar en la siguiente fotografía.

DESCRIPCIÓN MECÁNICA

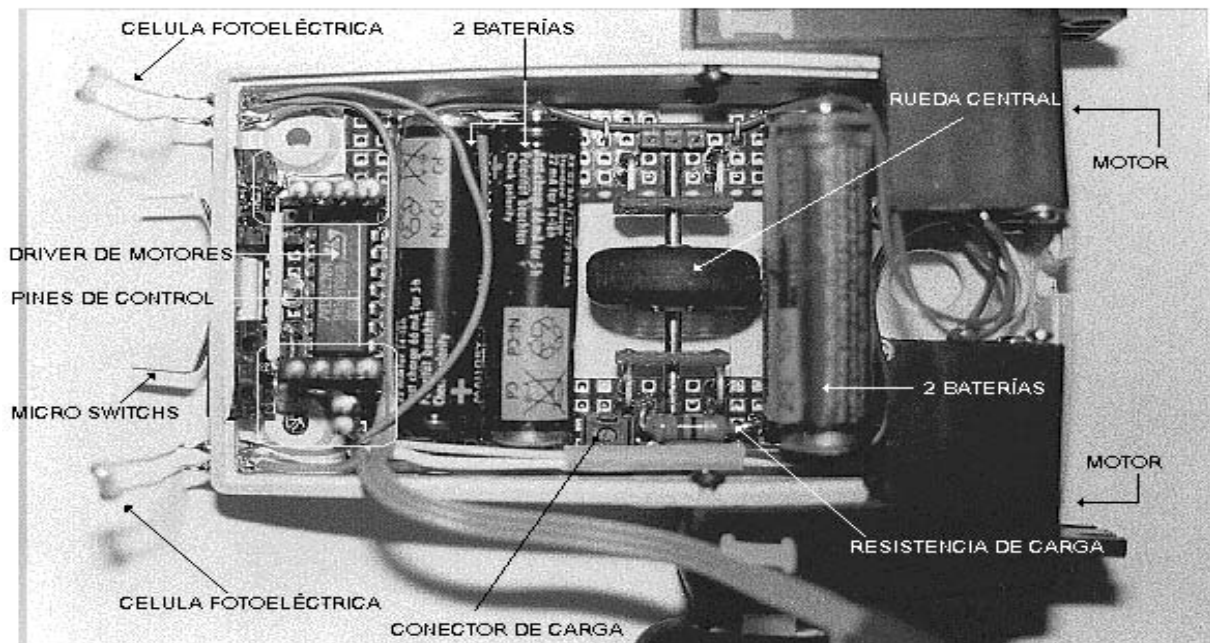
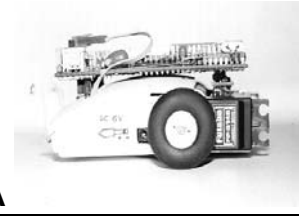


Figura 14. Aspecto interior del robot.

6.1.2 Circuito impreso interior.

El circuito impreso interior del robot contiene los siguientes elementos:

- **Baterías.**
El conjunto de baterías (4 de 1.2 voltios) genera una tensión de 5 voltios aproximadamente (a plena carga). Las cuatro se encuentran soldadas directamente al circuito y conectadas entre sí en serie. En la **Figura 14. Aspecto interior del robot.** se puede apreciar la ubicación de las mismas, en dicha figura se puede ver que dos de las baterías se encuentran una junto a la otra; mientras que las otras dos se encuentran una encima de la otra por lo que en dicha fotografía sólo se aprecia la superior.
- **Driver de los motores.**
Tal y como se comentó anteriormente, este circuito se encuentra montado sobre un zócalo y permite el control de los motores.
- **Rueda de giro.**
Esta rueda permite dar estabilidad al robot y está fijada al mismo mediante un eje que es soportado por otros dos pequeños trozos de placa, los cuales se encuentran soldados mediante hilo rígido al circuito impreso.
Esta rueda permite girar sobre sí mismo al robot
- **Conector de carga de la batería.**
El conector de carga de la batería está soldado a esta placa. Así mismo, la resistencia de carga que limita la corriente que la batería absorbe del exterior durante el proceso de carga se encuentra también en este circuito.



DESCRIPCIÓN MECÁNICA

Las siguientes fotografías muestran distintos aspectos del robot.

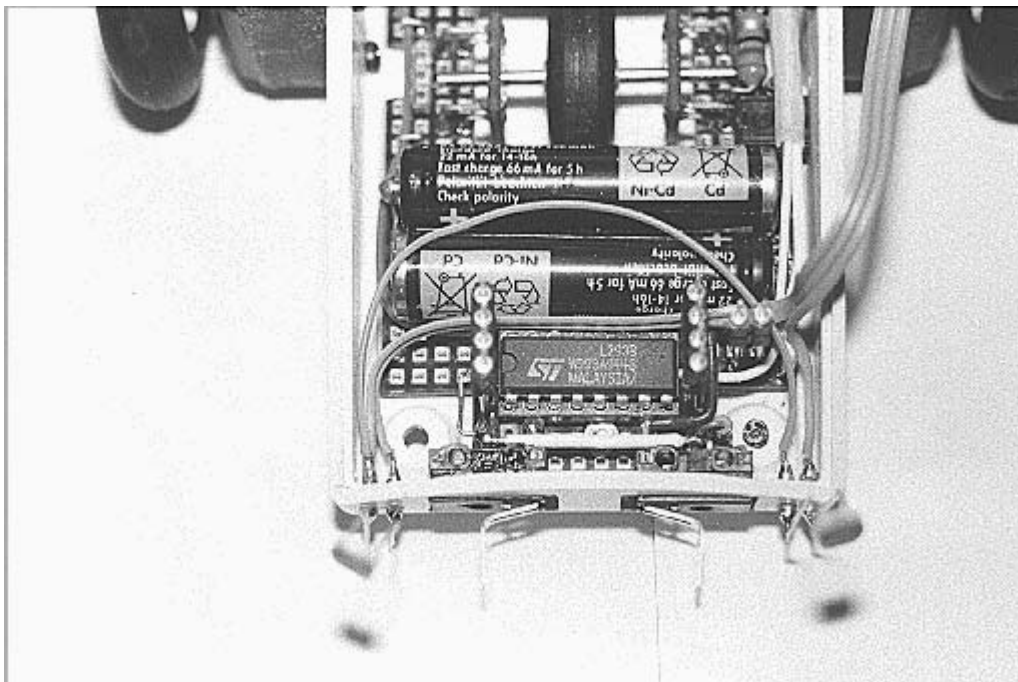


Figura 15. Aspecto delantero.

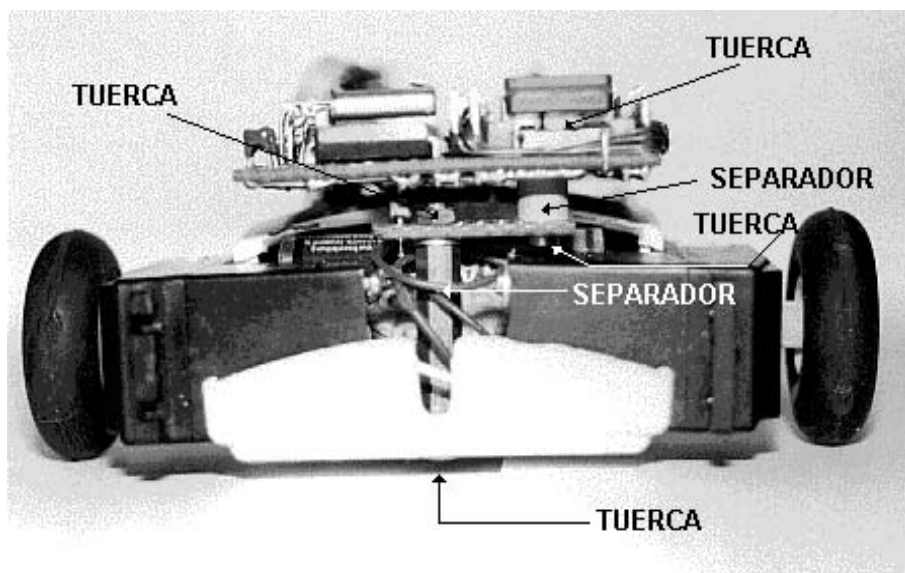


Figura 16. Parte posterior. Cabe destacar el separador metálico (entre los dos motores) que sujeta la tarjeta de control con el robot en sí.

DESCRIPCIÓN MECÁNICA

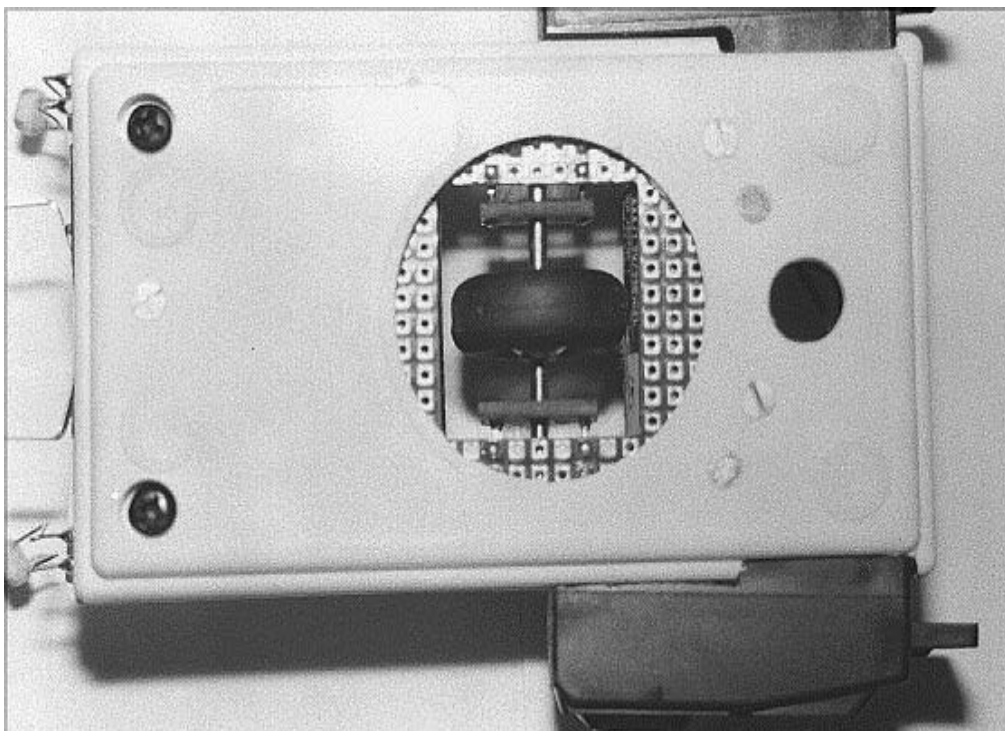
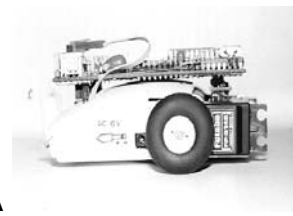
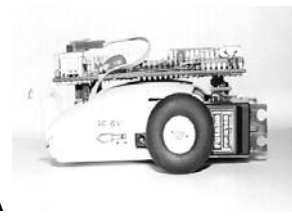


Figura 17. Aspecto inferior, mostrando la rueda estabilizadora.



DESCRIPCIÓN MECÁNICA

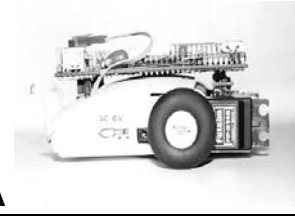
7. LISTA DE COMPONENTES

A continuación se detalla la lista de componentes que forman el CS-EL.

7.1 COMPONENTES ELECTRÓNICOS.

<i>Referencia</i>	<i>Cantidad</i>	<i>Valor</i>	<i>Descripción</i>
R1,R5	2	47K, 1/4W	Resistencia
R3,R4	2	10K, 1/4W	Resistencia
R6	1	100K, 1/4W	Resistencia
R7	1	560Ω, 1/4W	Resistencia
R8	1	62Ω, 1/2W	Resistencia
C1	1	10μF/16V	Condensador de Tántalo
C2,C3	2	22pF	Condensador cerámico
C4	1	100μF, 16V	Condensador Electrolítico
C5	1	100nF	Condensador cerámico
D1,D2,D3,D4,D5,D6,D7,D8	8	1N4148	Diodo
DL1	1	Led Rojo	Diodo Led
LDR1,LDR2	2	Célula fotoeléctrica	LDRs
IC1	1	80C31BH 16P	Microprocesador, versión para reloj de hasta 16MHz
IC2	1	74HCT373	Latch 8 Bits
IC3	1	27C64	Memoria EPROM 8Kbytes x 8
IC4	1	CD4051	Multiplexor analógico CMOS
IC5	1	ADC0831CCN	Conversor Analógico / Digital , National Semiconductor
IC6	1	CD4093	Cuadruple puerta NAND, CMOS
IC7	1	L293B	Driver de motores
XTAL	1	12 MHz	Cristal de cuarzo

Tabla 2. Relación de componentes.



DESCRIPCIÓN MECÁNICA

7.2 COMPONENTES VARIOS.

<i>Referencia</i>	<i>Cantidad</i>	<i>Valor</i>	<i>Descripción</i>
--	1	Basic Mouse	<i>Ratón, marca LOGITECH</i>
M1,M2	2	FP-S148	<i>Servos, FUTABA</i>
--	1	Zocálo 28 Pines	<i>Zócalo para la memoria EPROM</i>
--	1	Zocálo 14 Pines	<i>Zócalo para el L293B</i>
SW1, SW2	2	Micro Switch	<i>Obtenidos del ratón</i>
SW3	1	Micro interruptor	<i>ON/OFF</i>
--	2	Ruedas goma	<i>Ruedas de tracción</i>
--	1	Rueda goma	<i>Rueda central</i>
--	1	Varilla metálica	<i>Eje para la rueda central, alrededor de 4 cm en este prototipo</i>
--	1	Conector de alimentación	
BT	4	Baterías Mod. RX 003 AAA, ACCU1000 o similar. (cuatro en serie)	<i>1.2V / 220mAh, carga rápida en 5h a 66mA</i>
--	1	Placa circuito Impreso multitaladrada	
--	10	Pines de Wire Wrapping, macho	<i>Altura pequeña</i>
--	10	Pines de soldar Hembra	<i>Para su conexión con los anteriores</i>
--	1	Interruptor	<i>Interruptor ON/OFF</i>
JP	1	Micropuente	<i>medida de corriente</i>
CN1	1	Conector hembra minatura para alimentación.	<i>Conector</i>

Tabla 3. Relación de componente varios.