

# Control de motores de C.C (driver L293B)

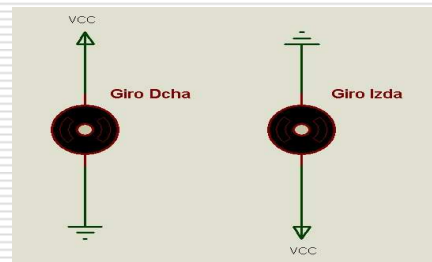
---



# Características de un motor de C.C

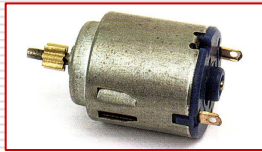
---

- Convierte la energía eléctrica en mecánica (Giro de su eje).
- Está compuesto por una parte fija (estator) y un eje móvil (rotor).
- Fácil control de posición, velocidad y giro.
- Máquina reversible motor/generador. Al mover el eje se produce una tensión en bornes del motor.
- La velocidad de giro es proporcional a la tensión de alimentación.
- Para invertir el sentido basta invertir el sentido de la corriente.



# Tipos de motores de C.C

---



Motor sin reductora de uso general



Motor como ventilador de refrigeración en ordenadores, fuentes de alimentación, etc.

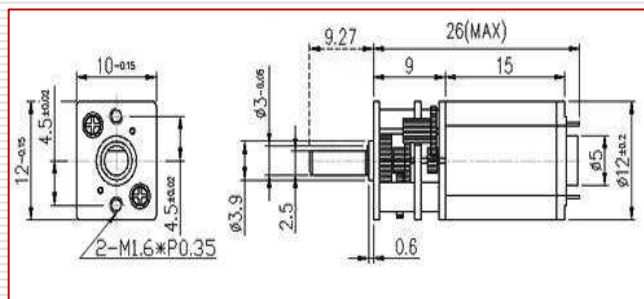


Motor con caja reductora. Disminuye la velocidad en el eje a costa de ganar fuerza (par motoriz).



Motor con caja reductora de doble eje.

# Características del motor “Micro gear motor”



- Voltaje nominal: 6V (\*)
- Velocidad de giro RPM: 140
- Fuerza en el eje: (4.7 kg-cm)
- Corriente con eje bloqueado: 1600mA
- Reductora: 210:1
- Tamaño (mm): 1.42" x 0.39" x 0.47"
- Peso: 10g
- PVP: 16 €

(\*) Aumentar la tensión de alimentación produce un aumento de velocidad de giro, pero no conviene superar en más de un 20% el valor nominal para no provocar sobrecalentamientos.

Fuente:

<http://www.pololu.com/>  
<http://www.microcontroladores.com>

(c) Domingo Llorente 2010

# Driver L293B (I): Características

---

Cuatro canales de salida (habilitados de dos en dos).

- Corriente de salida de hasta 1A por canal.
- Señales de control compatibles TTL (max. 7v). (conexión directa con el PIC)
- Posibilidad de controlar hasta cuatro motores sin inversión de giro o dos motores con control de giro.
- Posibilidad de alimentación externa de motores de hasta 36v.
- El modelo L293D incluye diodos de protección internos.
- La hoja de características original, en inglés, del L293 se puede descargar en: <http://www.megaupload.com/?d=SUMHZNW9>
- La traducción de la hoja por Fernando Remiro se puede descargar en: <http://www.terra.es/personal/fremiro/Archivos/L293b.pdf>

# Driver L293B (II): Diagramas y encapsulado

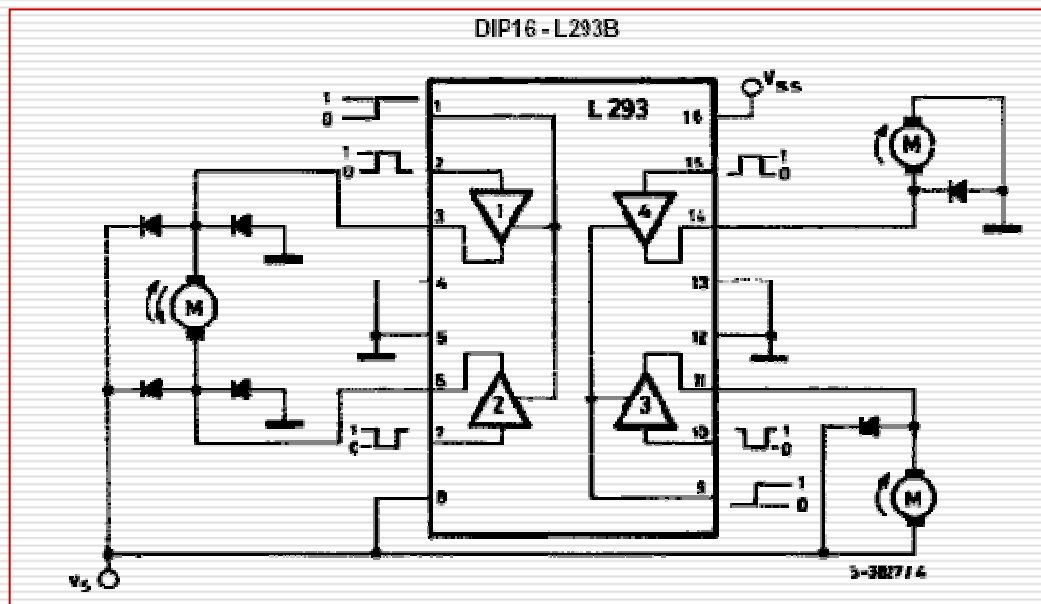
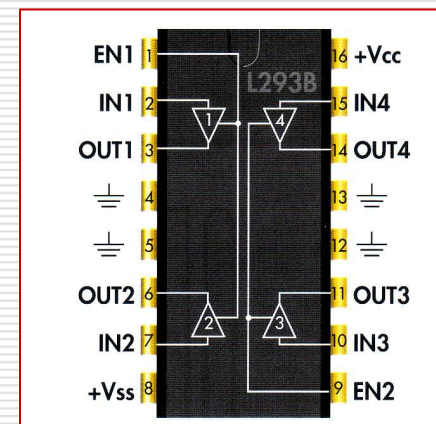


Diagrama de conexión general

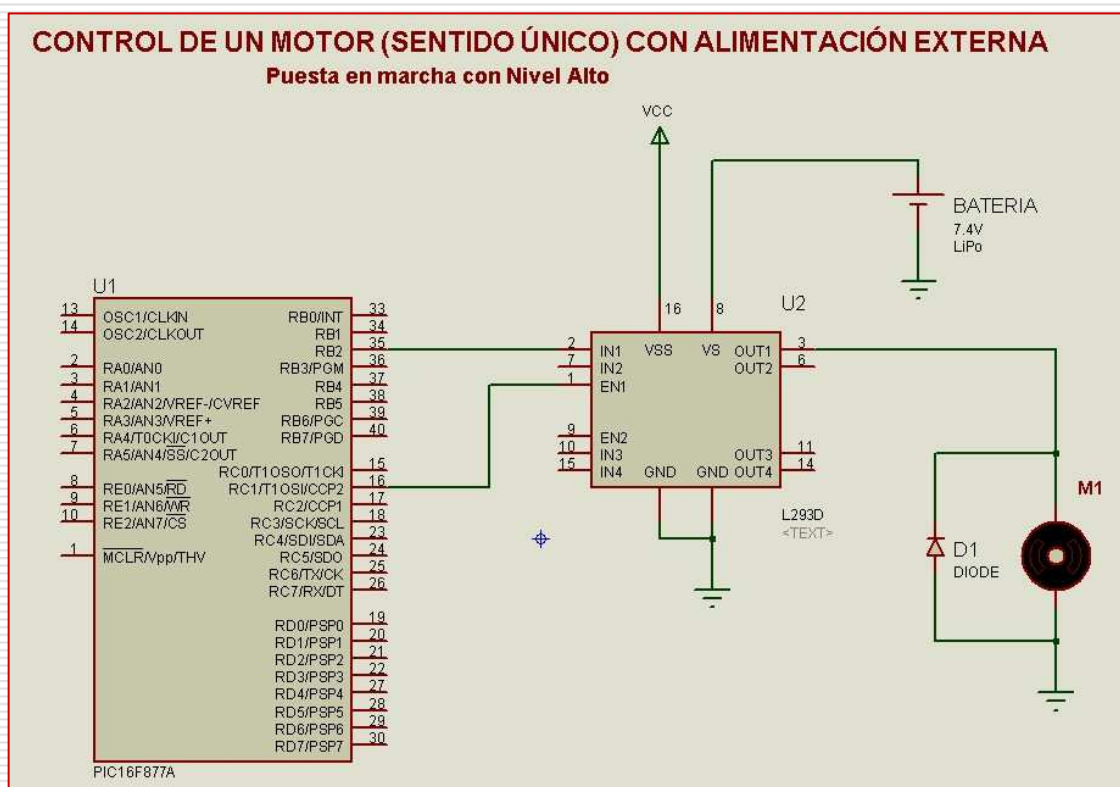


Encapsulado DIL-16



Patillaje L293B

---



## Puesta en marcha:

RB2=1 y RC1=1

## Parada:

RB2=0 ó RC1=0

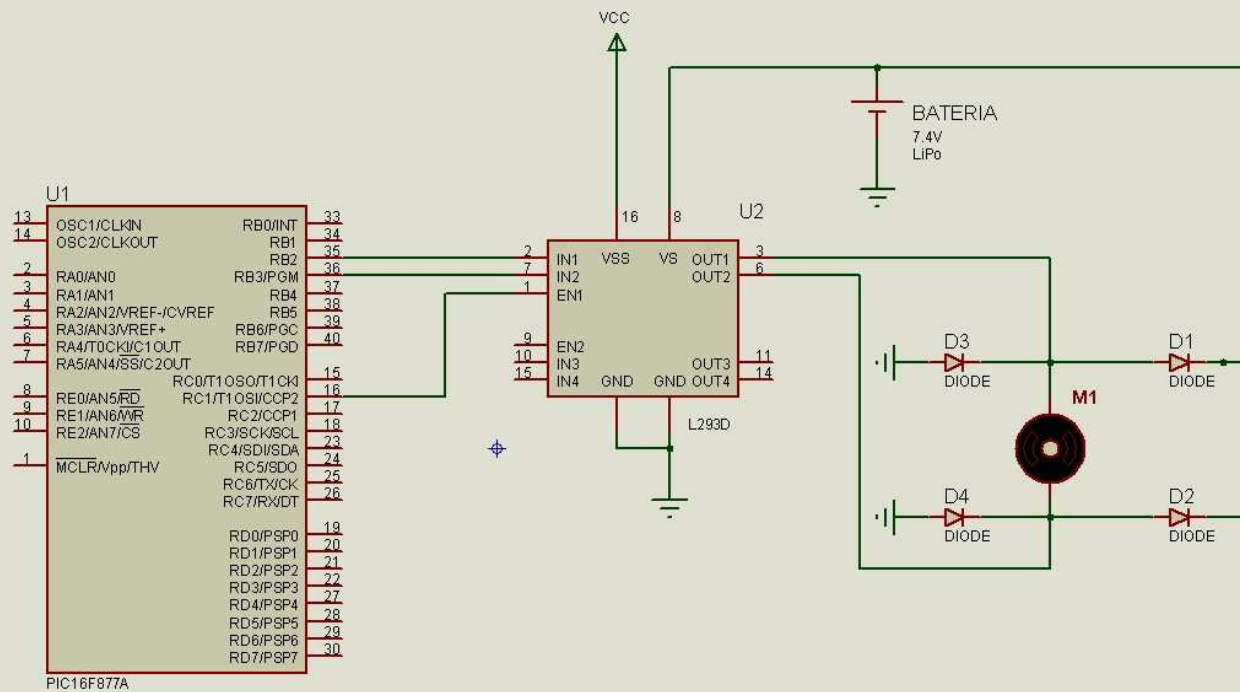
[illegible]

Parada:



# Driver L293B (IV): Conexión 3

## CONTROL DE UN MOTOR (CONTROL DE GIRO Y PROTECCIÓN DE DIODOS) CON ALIMENTACIÓN EXTERNA



Giro Derecha:

RB2=1 y RB3=0  
RC1=1

Giro Izquierda:

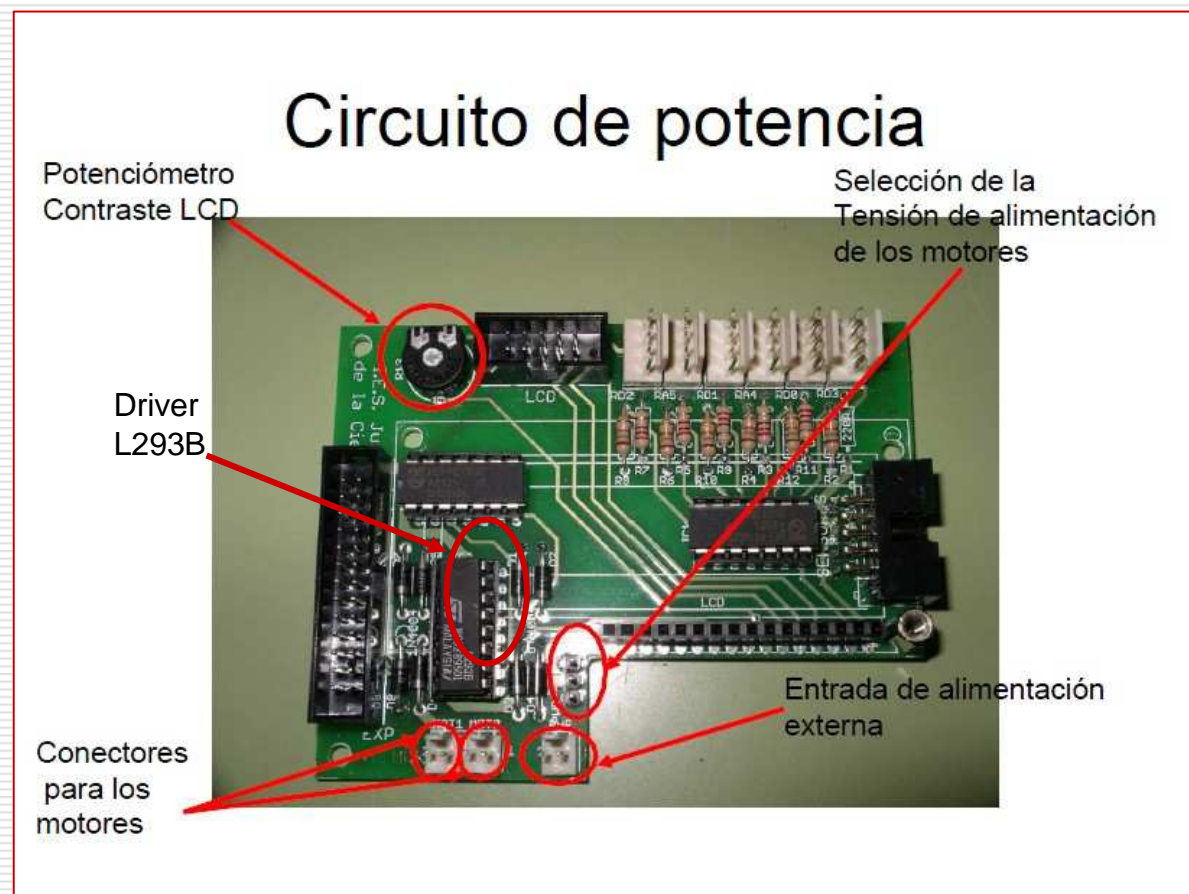
RB2=0 y RB3=1  
RC1=1

Parada:

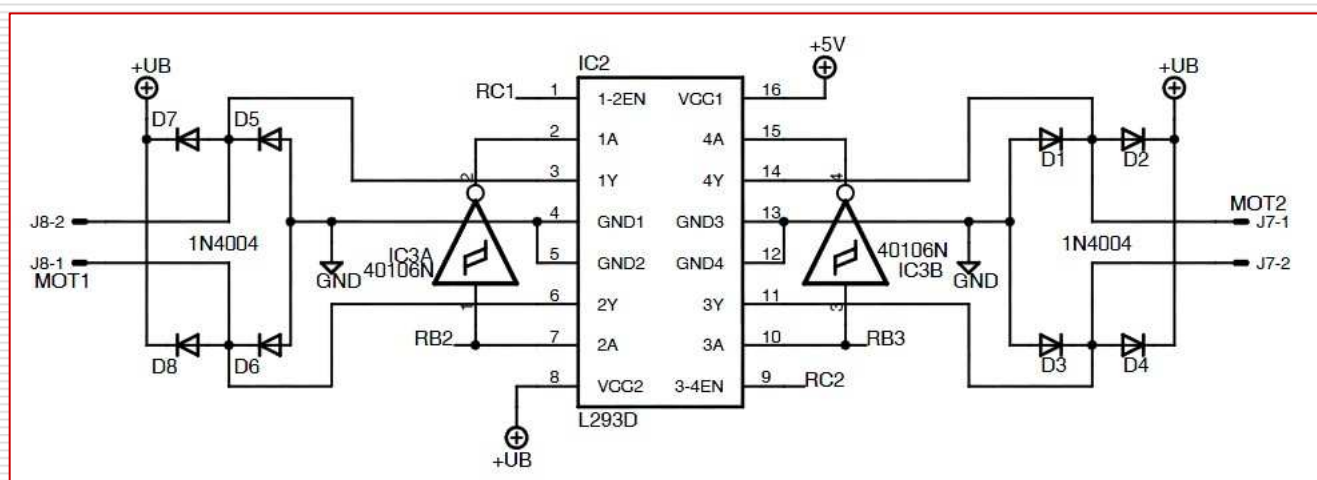
RC1=0

Nota: El sentido de giro no se puede garantizar hay que probar en la práctica e invertir los terminales del motor si fuera necesario.

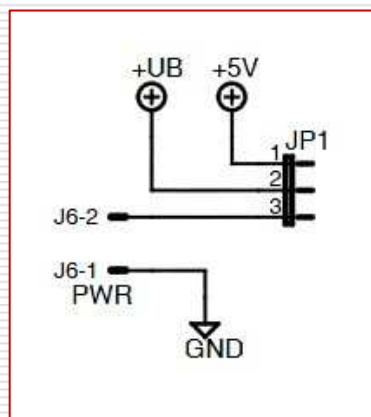
# El driver L293B en la plataforma Monibot (I)



# El driver L293B en la plataforma Monibot (II)



Conexiones  
del driver



Jumper JP1 de selección  
de alimentación de  
los motores

# Funciones de control de motores en CCS (I)

- Para poder controlar los motores vamos a crear un archivo "movimientos.c" con las funciones de todos los movimientos posibles de nuestro robot.

```
movimientos2.c *
1  //////////////////////////////////////
2  // FICHERO CON LAS FUNCIONES DE CONTROL DE MOTORES//
3  // para placa MONIBOT basada en pic-16F877A      //
4  // Fecha: diciembre-2010 by:Domingo Llorente      //
5  // ****
6  // Declaración de funciones                        //
7  // void Init_motores(void)                        //
8  // void Adelante(void)                          //
9  // void Atras(void)                             //
10 // void Paro(void)                               //
11 // void Rota_izda(void)                          //
12 // void Rota_dcha(void)                          //
13 // void Giro_izda(void)                          //
14 // void Giro_dcha(void)                          //
15 // void Rev_dcha(void)                           //
16 // void Rev_izda(void)                           //
17 // ****
18
19 // Declaraciones Monibot_2010, cambiar para otras plataformas o conexiones
20 #byte TRISB=0x86
21 #byte TRISC=0x87
22 #define MOTOR_IZDO PIN_B2
23 #define MOTOR_DCHO PIN_B3
24
```

# Funciones de control de motores en CCS (II)

```
29 //////////////////////////////////////////////////
30 // Función: Init_motores()
31 // Descripción: Configura los pines de salida y para los motores
32 //////////////////////////////////////////////////
33 void Init_motores(void)
34 {
35     bit_clear(TRISB,2);      // RB2 como salida (MOT1=Motor izquierdo)
36     bit_clear(TRISB,3);      // RB3 como salida (MOT2=Motor derecho)
37     bit_clear(TRISC,1);      // RC1 como salida (Habilitación del MOT1)
38     bit_clear(TRISC,2);      // RC2 como salida (Habilitación del MOT2)
39     output_low(PIN_C1);      // Deshabilitamos salida Motor Izdo
40     output_low(PIN_C2);      // Deshabilitamos salida Motor Dcho
41     return;
42 }
43
44
45 //////////////////////////////////////////////////
46 // Función: Adelante()
47 // Descripción: Avance de ambos motores
48 //////////////////////////////////////////////////
49 void Adelante(void)
50 {
51     output_high(PIN_C1);      // Habilitamos salida Motor Izdo
52     output_high(PIN_C2);      // Habilitamos salida Motor Dcho
53     output_high(MOTOR_IZDO);  // Motor Izdo avanza
54     output_high(MOTOR_DCHO);  // Motor Dcho avanza
55     return;
56 }
```

# Funciones de control de motores en CCS (III)

```
58 ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
59 // Función: Atras()
60 // Descripción: Retroceso de los dos motores
61 ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
62 void Atras(void)
63 {
64     output_high(PIN_C1);      // Habilitamos salida Motor Izdo
65     output_high(PIN_C2);      // Habilitamos salida Motor Dcho
66     output_low(MOTOR_IZDO);    // Motor Izdo avanza
67     output_low(MOTOR_DCHO);    // Motor Dcho avanza
68     return;
69 }
70
71 ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
72 // Función: Paro()
73 // Descripción: Parada de ambos motores
74 ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
75 void Paro(void)
76 {
77     output_low(PIN_C1);        // Deshabilitamos salida Motor Izdo
78     output_low(PIN_C2);        // Deshabilitamos salida Motor Dcho
79     return;
80 }
81
```



# Funciones de control de motores en CCS (IV)

```
82 //////////////////////////////////////////////////
83 // Función: Rota_izda()
84 // Descripción: Rotación a la izquierda sobre su propio eje
85 //////////////////////////////////////////////////
86 void Rota_izda(void)
87 {
88     output_high(PIN_C1);      // Habilitamos salida Motor Izdo
89     output_high(PIN_C2);      // Habilitamos salida Motor Dcho
90     output_low(MOTOR_IZDO);    // Motor Izdo retrocede
91     output_high(MOTOR_DCHO);   // Motor Dcho avanza
92     return;
93 }
94
95 //////////////////////////////////////////////////
96 // Función: Rota_dcha()
97 // Descripción: Rotación a la derecha sobre su propio eje
98 //////////////////////////////////////////////////
99 void Rota_dcha(void)
100 {
101     output_high(PIN_C1);      // Habilitamos salida Motor Izdo
102     output_high(PIN_C2);      // Habilitamos salida Motor Dcho
103     output_high(MOTOR_IZDO);   // Motor Izdo avanza
104     output_low(MOTOR_DCHO);    // Motor Dcho retrocede
105     return;
106 }
```

# Funciones de control de motores en CCS (V)

---

```
108 ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
109 // Función: Giro_Dcha()
110 // Descripción: Giro a la derecha
111 ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
112 void Giro_dcha(void)
113 {
114     output_high(PIN_C1);      // Habilitamos salida Motor Izdo
115     output_low(PIN_C2);      // Deshabilitamos salida Motor Dcho
116     output_high(MOTOR_IZDO);  // Motor Izdo avanza
117     return;
118 }
119
120 ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
121 // Función: Giro_izda()
122 // Descripción: Giro a la izquierda
123 ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
124 void Giro_izda(void)
125 {
126     output_low(PIN_C1);      // Deshabilitamos salida Motor Izdo
127     output_high(PIN_C2);     // Habilitamos salida Motor Dcho
128     output_high(MOTOR_DCHO); // Motor Dchoo avanza
129     return;
130 }
```

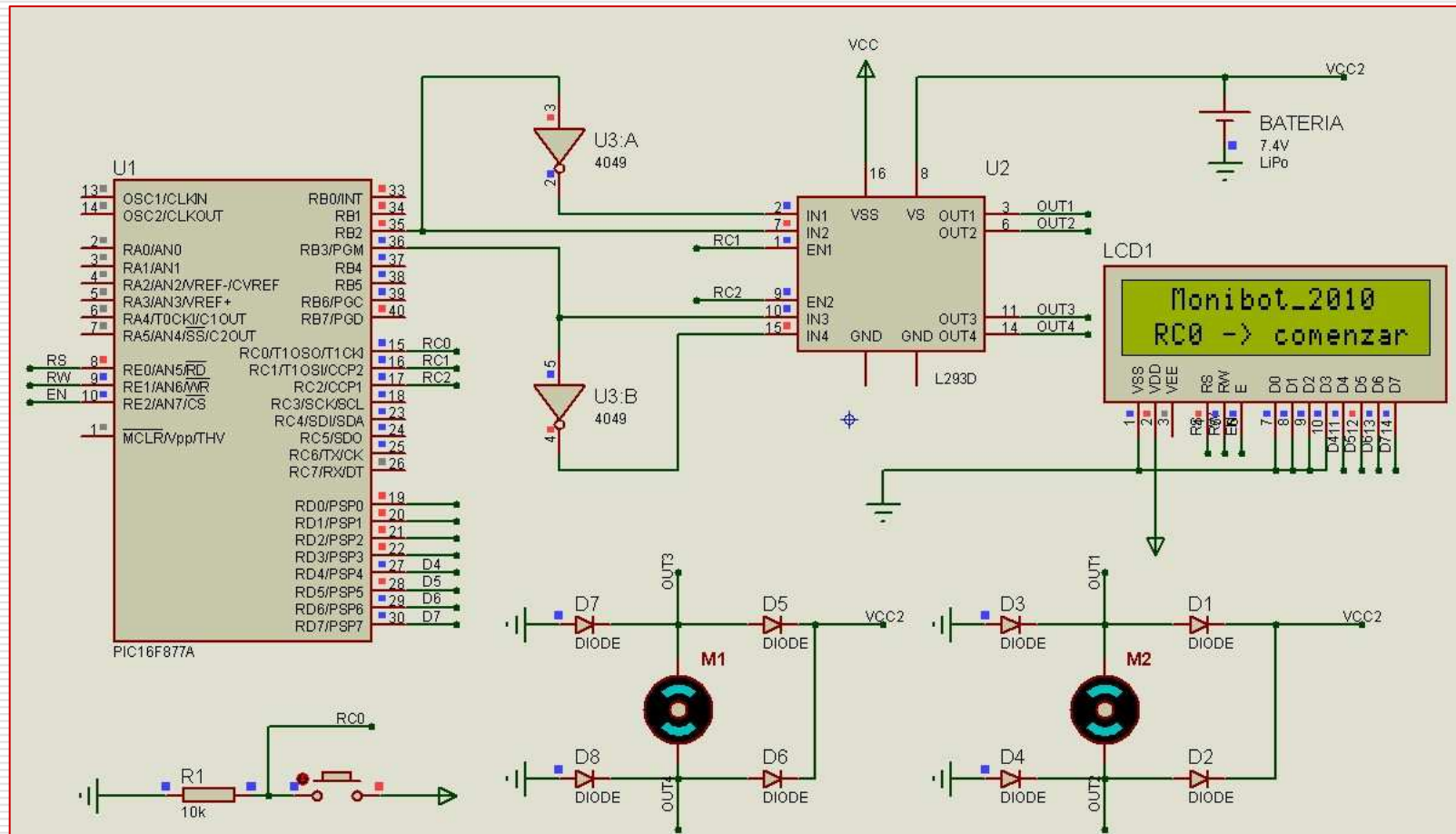


# Funciones de control de motores en CCS (VI)

```
132 ///////////////////////////////////////////////////  
133 // Función: Rev_izda()  
134 // Descripción: Giro a la izquierda marcha atrás  
135 ///////////////////////////////////////////////////  
136 void Rev_izda(void)  
137 {  
138     output_low(PIN_C1);        // Deshabilitamos salida Motor Izdo  
139     output_high(PIN_C2);       // Habilitamos salida Motor Dcho  
140     output_low(MOTOR_DCHO);    // Motor Dcho retrocede  
141     return;  
142 }  
143  
144 ///////////////////////////////////////////////////  
145 // Función: Rev_dcha()  
146 // Descripción: Giro a la derecha marcha atrás  
147 ///////////////////////////////////////////////////  
148 void Rev_dcha(void)  
149 {  
150     output_high(PIN_C1);       // Habilitamos salida Motor Izdo  
151     output_low(PIN_C2);        // Deshabilitamos salida Motor Dcho  
152     output_high(MOTOR_IZDO);   // Motor Izdo retrocede  
153     return;  
154 }
```

# Ejemplo de programa

Diseñar un programa que realice todos los movimientos del robot durante 3 segundos de forma ininterrumpida al pulsar RC0.



(c) Domingo Llorente 2010

## Solución propuesta (parte 1)

```
1  //////////////////////////////////////
2  // Programa.....: movimientos1.c                                     //
3  // Plataforma hw: Placa monibot 16F877A + motores DC en MOT1 y MOT2   //
4  // Fecha.....: Enero-2010                                           //
5  // Programador...: Domingo Llorente                                   //
6  // Descripción...: Prueba de las funciones de control de motores incluidas en //
7  // el archivo "movimientos.c" (Adelante,Atras,Giro_dcha,Giro_izda,Rota_dcha, //
8  // Rota_izda,Rev_dcha,Rev_izda y Paro).                               //
9  //////////////////////////////////////
10 #include <16F877A.h>
11 #fuses XT,NOWDT,NOPROTECT,NOLVP
12 #use delay(clock=4000000)
13 #include <lcd4.c>
14 #include <movimientos.c>
15
16 //////////////////////////////////  MENSAJES DE TEXTO  //////////////////////////////////
17 char msg1[]= "  Movimientos ";
18 char msg2[]= "  Monibot_2010";
19
20 void main()
21 {
22     Init_Motores();           // Configuración de E/S motores
23     LCD_Init4();
24     LCD_Escribecadena(msg1);  // Escribe la cadena msg1 en el lcd
25     LCD_Cambiolinea();        // Cambio a la segunda línea
26     LCD_Escribecadena(msg2);  // Escribe la cadena msg2 en el lcd
27 }
```

## Solución propuesta (parte 2)

```
28 while(true)
29 {
30   while(!input(PIN_C0)) {}; // Espera a pulsar RC0 para comenzar
31   Adelante();
32   delay_ms(3000);
33   Paro();
34   Atras();
35   delay_ms(3000);
36   Paro();
37   Giro_dcha();
38   delay_ms(3000);
39   Paro();
40   Giro_izda();
41   delay_ms(3000);
42   Paro();
43   Rota_dcha();
44   delay_ms(3000);
45   Paro();
46   Rota_izda();
47   delay_ms(3000);
48   Paro();
49   Rev_izda();
50   delay_ms(3000);
51   Paro();
52   Rev_dcha();
53   delay_ms(3000);
54   Paro();
55 }
56 }
```