# ROBÓTICA
# CONTROL DE MOTORES DESDE MICROCONTROLADORES CON EL CIRCUITO INTEGRADO L293

## INTRODUCCIÓN

El presente documento busca cubrir uno de los problemas más comunes que suele encontrar el aficionado en su iniciación en la robótica: el control de motores de corriente continua (DC) cuando se requiere un movimiento.

## PROBLEMA A RESOLVER

Partimos del supuesto que contempla el control de uno, dos o más motores por un circuito integrado programable (un microcontrolador) de acuerdo a una programación preestablecida y respondiendo a una serie de variables que lee en el entorno.

Normalmente, el control de un motor DC irá aparejado a la escritura de un 1 lógico (3,5V-5V) y un 0 lógico (<1,5 V) en cada uno de los bornes del motor.

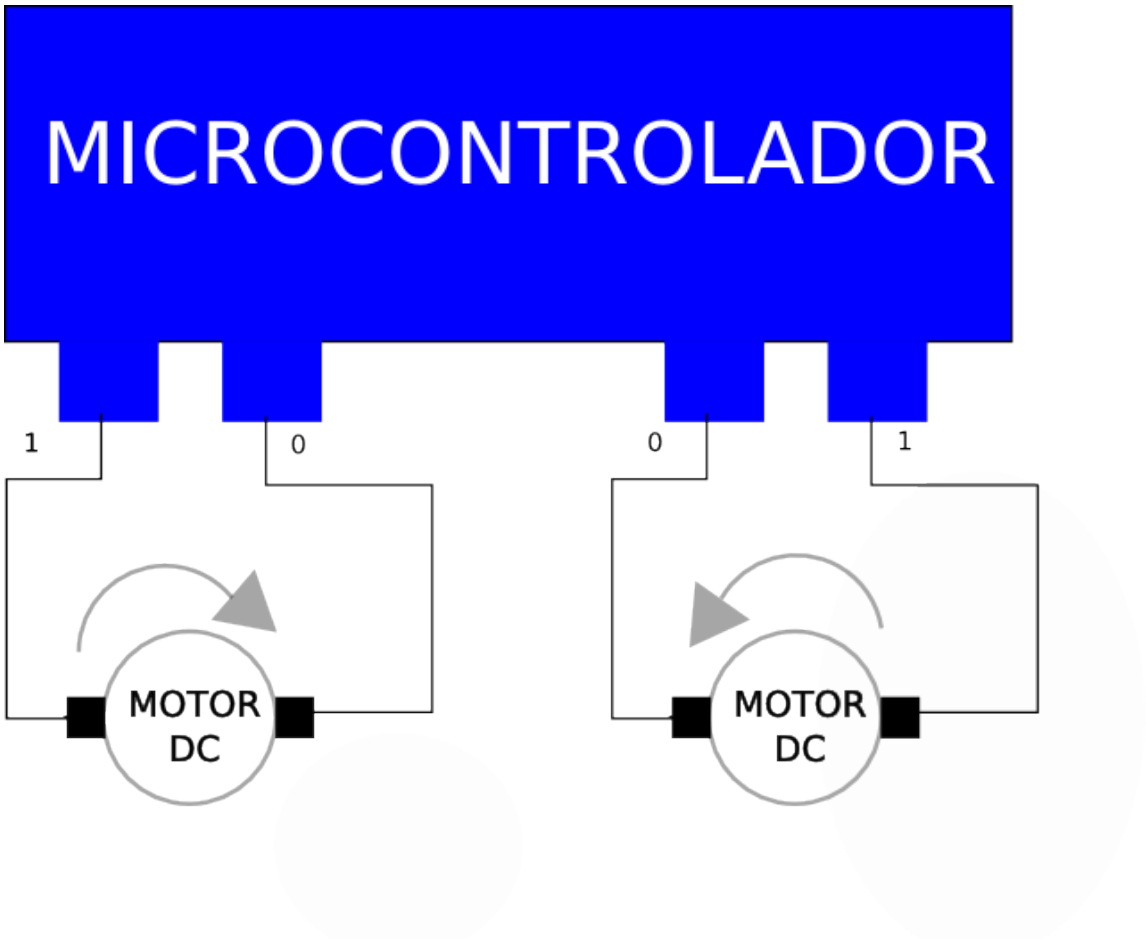


*Ilustración 1: El control directo de un motor DC desde un microcontrolador no es técnicamente posible debido a su carácter inductivo.*

El problema suele ser que, si bien las patillas del microcontrolador responden adecuadamente a las exigencias del programa en cuanto a tensión, a la hora de alimentar de manera directa al motor se producen muchos y varios problemas, relacionados, entre otras cosas, con el carácter eminentemente inductivo y analógico de estos motores, normalmente de juguete. La obediencia de estos dispositivos es, cuando menos, discutible. En ocasiones, se ponen en marcha con un retraso inexplicable respecto al momento en que el microcontrolador ha dado la orden. Otras veces, obedecen pero se paran espontáneamente. Y, desde luego, los cambios de sentido (cambios lógicos 1-0 a 0-1) se convierten en ejercicios de predicción. ¿Obedecerán los motores?. ¿Y cuándo lo harán?.

Tras un exhaustivo trabajo de investigación al respecto (todo lo exhaustivo que puede ser cuando lo realizamos auténticos profanos, eso sí, muy voluntariosos), hemos llegado a la conclusión de que, por un lado, los motores y el circuito de control (microcontrolador) deben tener alimentaciones diferentes, y por otro lado, que los motores, desde su puesta en marcha hasta alcanzar su velocidad normal de funcionamiento, y debido a su carácter inductivo, consumen una intensidad variable que las salidas de un microcontrolador no están en condiciones de suministrar de manera fiable.

Es por ello que se recomienda utilizar el circuito L293, que puede presentar distintas variantes. Concretamente, el modelo que hemos utilizado para nuestra investigación es el L293DNE.

## FUNCIONAMIENTO DEL INTEGRADO L293DNE

Este circuito se concibe como un “puente en H”, en el sentido de que está esperando dos entradas (1 ó 0 lógicos) por un lado, y repite exactamente estas dos entradas (1 ó 0 lógicos) por el otro lado, pero con la posibilidad de cambiar la fuente de alimentación,  (que puede llegar a 36 V), y garantizando el consumo de intensidad del dispositivo a conectar.

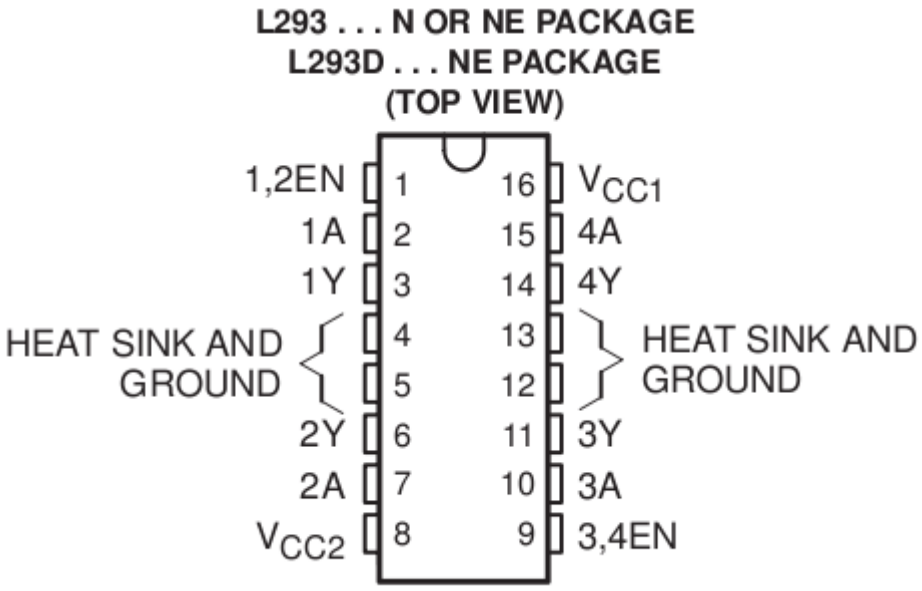


*Ilustración 2: Patillaje típico de cualquier integrado 293*

El patillaje responde a la siguiente relación de funciones:

- VCC1: Tensión de alimentación que supondrá la referencia para las entradas (de 4,5 V a 7 V, por lo común 5 V). Esto es, si está a 5 V, ésta es la tensión que supondría un 1 lógico para las entradas 1A, 2A, 3A y 4A. Por debajo de unos 2,3 V, se entenderá que la entrada está a 0.
- 1A, 2A,3A,4A: Pines de entrada, cuya referencia es VCC1..
- 1Y, 2Y ,3Y, 4Y: Pines de salida. Repetirán el 1 ó 0 lógico que se halle en sus respectivas entradas (1A, 2A, 3A, 4A), pero de acuerdo a la tensión VCC2, que puede llegar a 36 V.
- VCC2: Tensión de alimentación para las salidas 1Y, 2Y, 3Y y 4Y (deVCC1 a 36 V)
- GND (4 pines): Las 4 patillas deberán ir a tierra.
- 1, 2 EN: Patilla que "activa" las entradas 1A y 2A. Si dicho pin está a "0" dichas entradas estarán desactivadas (con el resultado de un "0" lógico inmediato en las correspondientes salidas 1Y, 2Y).
- 3, 4 EN: Patilla que "activa" las entradas 3A y 4A. Si dicho pin está a "0" dichas entradas estarán desactivadas (con el resultado de un "0" lógico inmediato en las correspondientes salidas 1Y, 2Y).

Todo lo arriba expuesto puede traducirse en la siguiente tabla de la verdad:

**TABLA DE LA VERDAD I**

| **EN1,2** | **1A** | **2A** | **1Y** | **2Y** |
|---|---|---|---|---|
| 0 | X | X | Z (ALTA IMPEDANCIA) | Z (ALTA IMPEDANCIA) |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |

**TABLA DE LA VERDAD II**

| **EN2,3** | **3A** | **34** | **3Y** | **4Y** |
|---|---|---|---|---|
| 0 | X | X | Z (ALTA IMPEDANCIA) | Z (ALTA IMPEDANCIA) |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |

En el control de motores, pondremos a "0" las patillas 1, 2 EN y 3, 4 EN para detener de manera inmediata a los correspondientes motores. Otras soluciones podrían ser poner a "0" ambas patillas (motor desactivado) o a "1" ambas patillas (motor bloqueado).

# CONTROL DE DOS MOTORES CON EL INTEGRADO L293DNE

A la luz de lo expuesto, parece lógico pensar en el siguiente paso: los motores no estarán conectados de manera directa a los correspondientes pines de control, sino a través del L293DNE, según esquema adjunto:

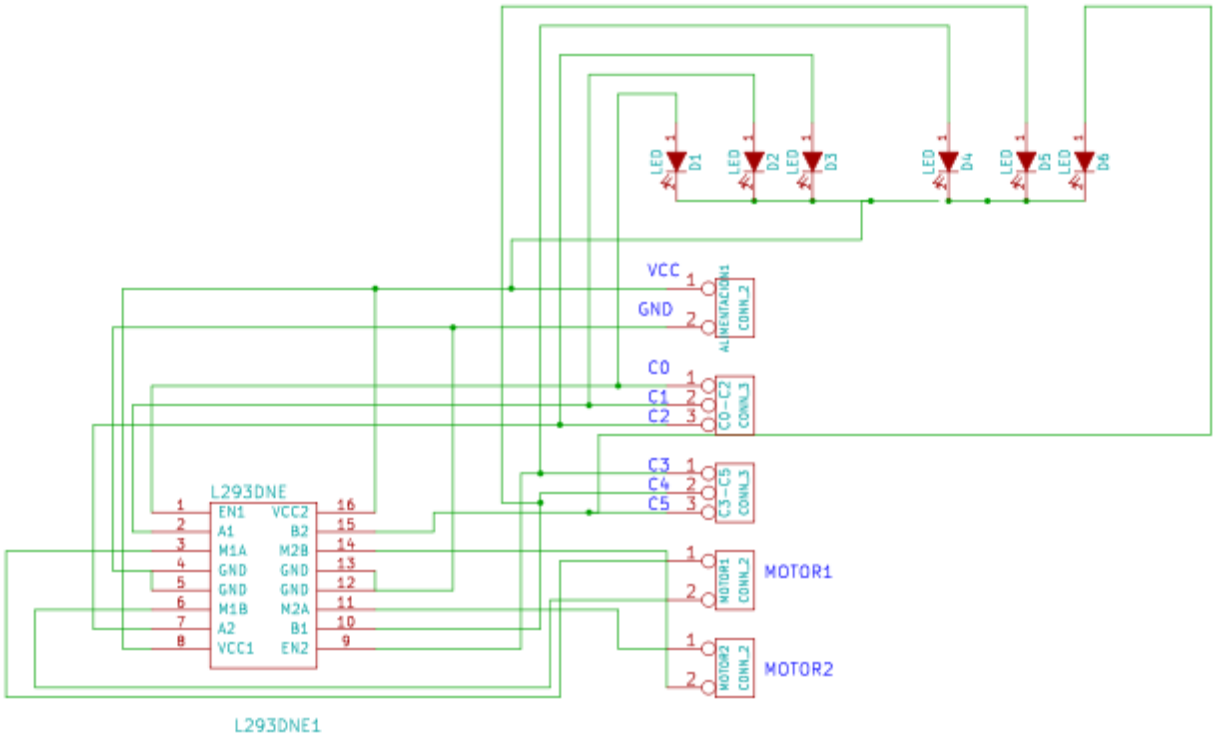


*Ilustración 3: Esquema de conexiones de un circuito de control de dos motores DC desde un L293DNE*

Para una mayor claridad (al menos para los redactores de este tutorial) hemos realizado las siguientes traducciones: 1A, 2A por A1 y A2, de modo que 1Y y 2 Y se convertirán en M1A y M1B, de cara al control del motor 1. Por otro lado, 3A, 4A son ahora B1 y B2, así que las correspondientes salidas 3Y, 4Y se convierten en M2A y M2B (control del motor 2).

En dicho esquema, puede verse que VCC1 y VCC2 van conectadas al mismo pin, VCC en la clema superior, dado que lo más común es que los motorcillos DC que queremos conectar vayan a tensiones no superiores a 4,5 V, precisamente del orden en que están trabajando los “1” lógicos de los pines de un microcontrolador estándar (un ejemplo muy representativo son las tarjetas ARDUINO). Por tanto, la clema (de dos entradas) de arriba, en la figura, iría a una pila de 4,5 V. Las dos clemas siguientes, (C0-C2) y (C3-C5), conectan, respectivamente, con las entradas 1,2 EN, A1 y A2, la primera, y con 3, 4 EN, B1 y B2, la segunda.. Exteriormente, dichas clemas irán conectadas a 6 pines configurados como salidas en el microcontrolador correspondiente.

Se recomienda fehacientemente, a la luz de los resultados de nuestros experimentos, que las conexiones a tierra de los circuitos de control (microcontrolador) y de fuerza (L293) sean puestas en común.

En nuestras prácticas, hemos trabajado con dos circuitos distintos, a saber: el microcontrolador PIC

16F690, del cual hemos utilizado como salidas las patillas correspondientes al puerto C (salidas 0-5, esto es, C0, C1, C2, C3, C4 y C5), y por otro lado, una tarjeta con base en ARDUINO, concretamente ARDUCLEMA, de las que hemos utilizado los pines digitales 2, 3, 4, 5, 6 y 7, previamente configurados como salidas. (Hay que decir que hemos obviado el uso de las patillas 0 y 1 porque en ocasiones tuvimos algún problema que no sabíamos si atribuír al hecho de que cuando ARDUINO trabaja en modo esclavo con el ordenador pudiera utilizar alguno de dichos pines para la comunicación SERIAL).

Terminando con el análisis de la figura, quedan dos clemas dobles, que van, precisamente, a las entradas de los dos motores que queremos gobernar. Así pues, en dicha figura, puede apreciarse que dichas clemas van conectadas a las correspondientes salidas M1A, M1B, en el caso de la clema superior, y M2A, M2B, en el caso de la clema inferior.

Los diodos LED tienen una función eminentemente informativa (el programador tiene información visual sobre la evolución del programa, viendo qué patillas están activadas a cada momento), además de facilitar la eliminación de las corrientes parásitas que se pueden producir en los cambios de sentido de los motores DC.

# CONSTRUCCIÓN DE UN ROBOT BÁSICO. PROYECTO MASAYLO.

Como ya hemos expresado en anteriores apartados, el objetivo es dotar a un robot básico (que hemos bautizado como MASAYLO) de rutinas de movimiento fácilmente incluibles en programas más complejos. Dicho robot estaría equipado de dos motores DC con mecanismos reductores, de los que se pueden encontrar fácilmente en cualquier taller de Tecnología de un instituto.

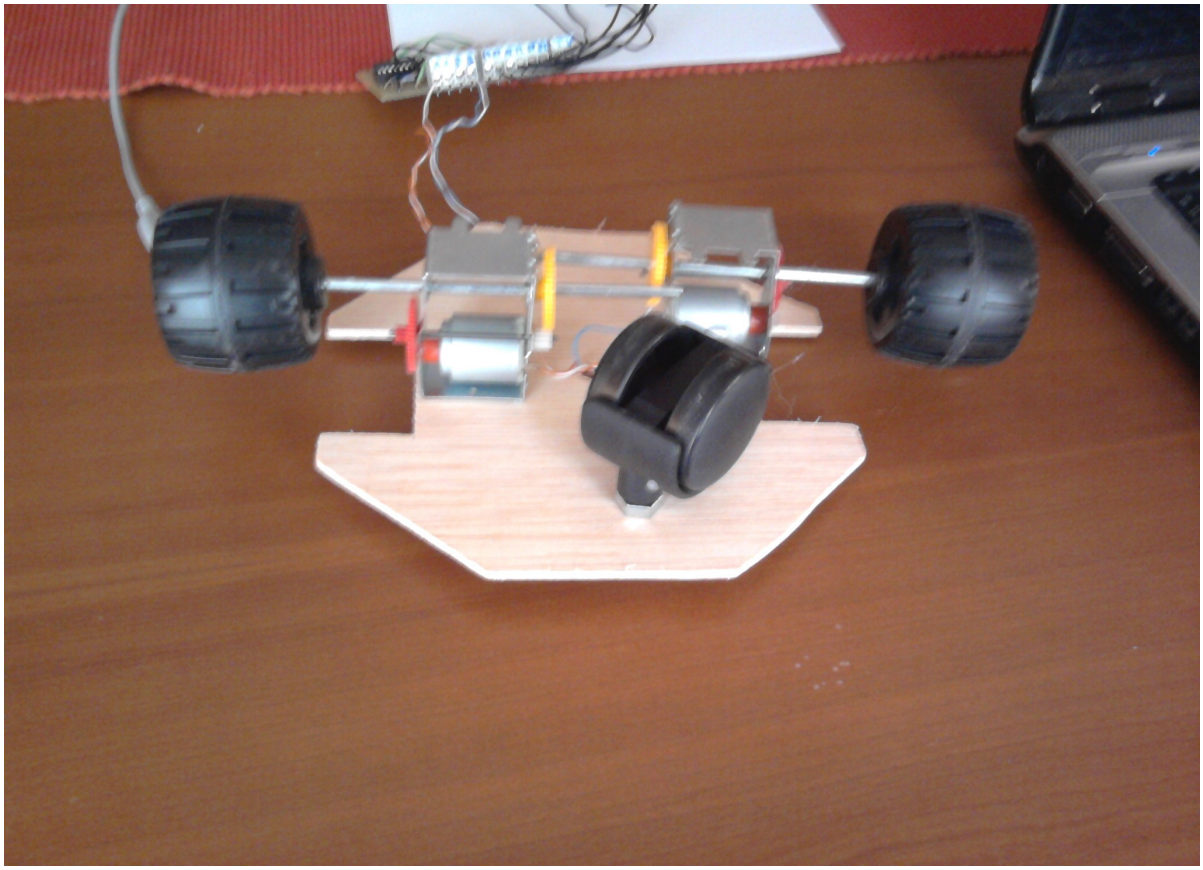

*Ilustración 4: MASAYLO cuenta con dos motores DC controlados por nuestro circuito basado en L293DNE. La rueda loca proporciona el equilibrio del centro de gravedad del robot.*

La rueda loca completa la infraestructura de tan sencillo robot, al objeto, sobre todo, de equilibrar el centro de gravedad del conjunto.

En la siguiente ilustración, puede apreciarse la apariencia final de un circuito implementado de acuerdo al diseño de la figura 3. El circuito L293DNE irá implementado en un zócalo, al objeto de poder reponer el integrado en caso de avería. Las clemas con tornillos posibilitan el fácil cambio de conexiones, dado que habrá que realizar un proceso de ensayo y error en la conexión de los motores a las salidas 1Y, 2Y, 3Y y 4Y (si un motor va hacia adelante cuando queremos que vaya hacia atrás, o viceversa, habrá que intercambiar la conexión de los cables de dicho motor; por otro lado, es posible que el sentido de ambos motores esté bien, pero gire a la izquierda cuando deseamos un giro a la derecha, o al revés, lo que exigirá intercambiar 1Y con 3Y y 2Y con 4Y)

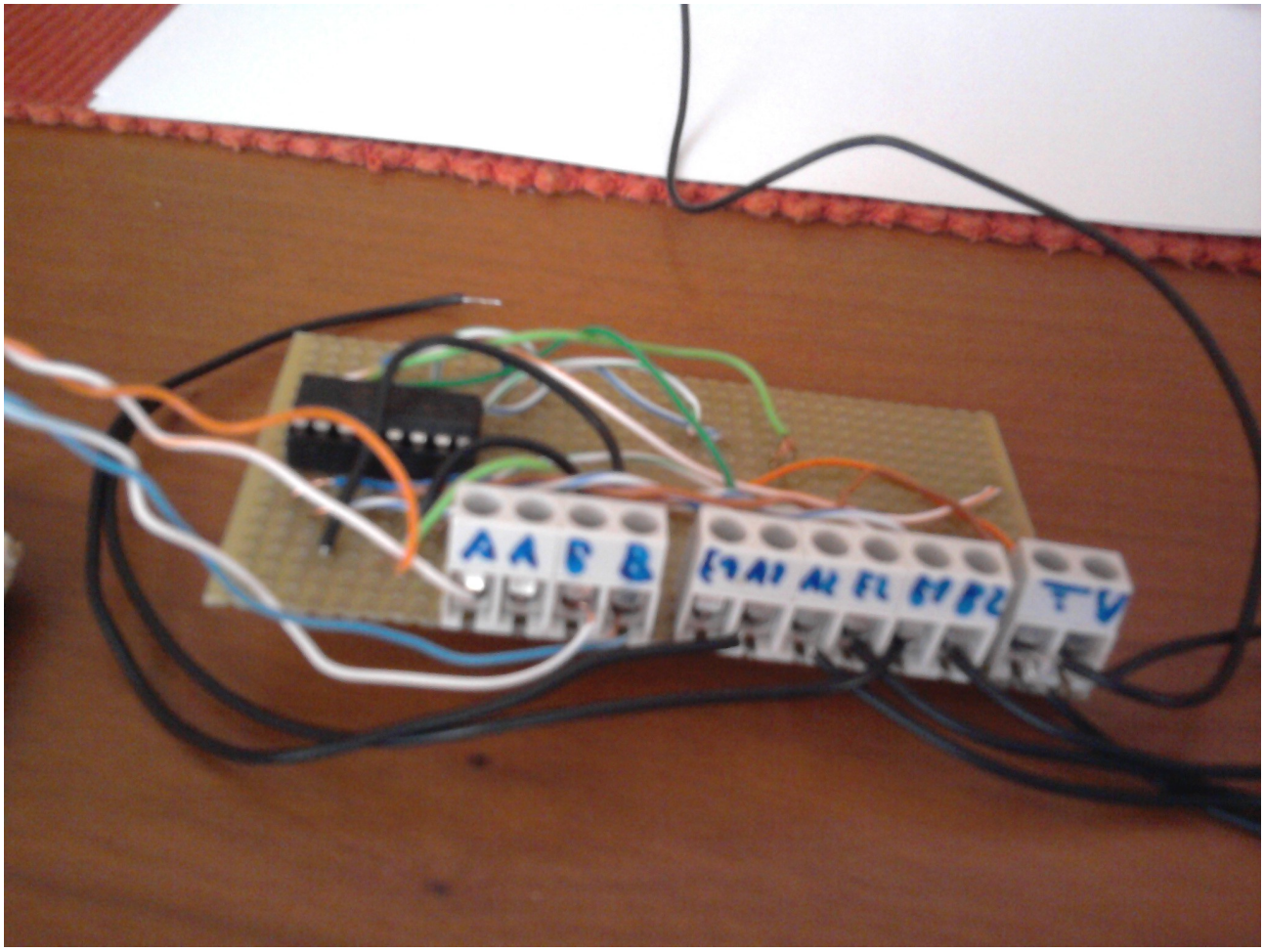

*Ilustración 5: Circuito de fuerza con base en L293DNE. Se recomienda que la conexión a tierra sea puesta en común con la del microcontrolador que lo gobierna.*

## CREACIÓN DE LAS RUTINAS BÁSICAS DE MOVIMIENTO EN C

En el transcurso de nuestros experimentos hemos trabajado con dos tipos de integrados de control: los microcontroladores PIC, concretamente el modelo 16F690, y la tarjeta ARDUINO, basada en el microcontrolador ATMEL168.

En el primer caso, hemos programado el 16F690 desde CODE:BLOCKS (software libre) para compilar el programa en C (compilador SDCC), para luego grabar el programa utilizando el grabador PICKIT2, sea a través del software de la empresa matriz, MICROCHIP, sea desde PIKLAB, en Linux (Ubuntu, concretamente). En ambos casos, el código base que implementaría el movimiento de MASAYLO (adjunto en los anexos de este documento) incluye las siguientes funciones, a las que se puede llamar en cualquier momento:

**adelante();** //EN1,2==1; 1A==1; 2A==0; EN2,3==1; 3A==1; 4A==1;

**atras();** //EN1,2==1; 1A==0; 2A==1; EN2,3==1; 3A==0; 4A==0;

**izquierda();** //EN1,2==0; 1A==1; 2A==0; EN2,3==1; 3A==1; 4A==1;

**derecha();** //EN1,2==1; 1A==1; 2A==0; EN2,3==0; 3A==1; 4A==1;

**alto();** EN1,2==0; 1A==1; 2A==0; EN2,3==0; 3A==1; 4A==1;

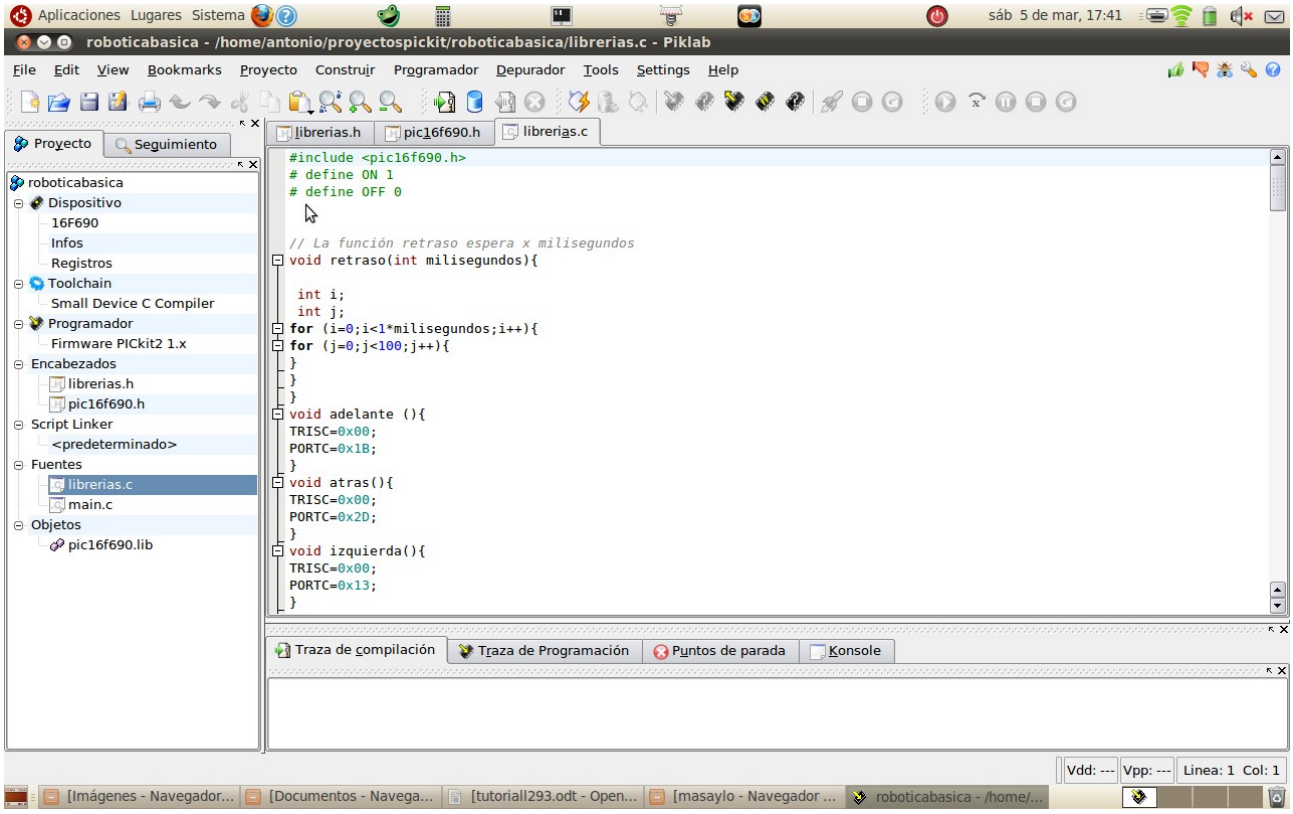


*Ilustración 6: En Ubuntu, hemos utilizado PIKLAB, equipado con el compilador SDCC, para programar el microcontrolador PIC16F690*

En PIC16F690, hemos utilizado las patillas C0, C1, C2, C3, C4 y C5 para conectar a las entradas del L293 (PUERTO C que será previamente configurado como de salida en su totalidad, TRISC=0x00). En el caso de ARDUINO, obnviaremos la conexión de los dos primeros pines digitales, 0 y 1, puesto que en algunos programas nos hemos encontrado con dificultades, que creemos (perdónesenos la ignorancia si erramos) están relacionadas con el uso de uno o de ambos pines por parte de la tarjeta en la comunicación con el ordenador en su uso como esclavo. Así, los pines a conectar a las entradas de nuestro circuito de fuerza serían los digitales 2, 3, 4, 5, 6 y 7.

# CONCLUSIÓN

A la hora de poner en marcha la construcción de un robot desde cero, como ha sido nuestra intención, hay muchos frentes que cubrir, y por supuesto, mucha más documentación (aparte de estas humildes líneas) que redactar, máxime cuando tenemos por objeto utilizar herramientas de software libre, así como iniciar una línea de investigación que pueda ser seguida y ampliada por otros docentes en un mundo tan apasionante como el de la robótica, que creemos firmemente abarcará varios aspectos de la evolución de la enseñanza de las Tecnologías en educación secundaria. Somos asímismo conscientes de los numerosos vacíos e inexactitudes de que este documento adolece. Sirva por lo menos como base para que otros investigadores lo utilicen como

punto de partida y mejoren su contenido.

Robótica. Control de motores desde microcontroladores con el circuito integrado L293por Antonio Gómez y Mª Dolores Nogueras se encuentra bajo una Licencia Creative Commons Reconocimiento-NoComercial-CompartirIgual 3.0 Unported.

# ANEXO I: LIBRERÍA DE CONTROL DEL L293 con SDCC para PIC 16F690

```
#include <pic16f690.h>
// La función retraso espera x milisegundos
void retraso(int milisegundos){
 int i;
 int j;
for (i=0;i<1*milisegundos;i++){
for (j=0;j<100;j++){
}
}
}
void adelante (){
TRISC=0x00;
PORTC=0x1B;
}
void atras(){
TRISC=0x00;
PORTC=0x2D;
}
void izquierda(){
TRISC=0x00;
PORTC=0x13;
}
void derecha(){
TRISC=0x00;
PORTC=0x1A;
}
void alto(){
TRISC=0x00;
PORTC=0x0A;
}
```

# ANEXO II: LIBRERÍA DE CONTROL DEL L293 DESDE ARDUINO

```
/*
 masaylobase: libreria de control del robot desde ARDUINO
 a traves del circuito de fuerza basado en el integrado
 L293DNE
 */

void setup() {
  // Inicializamos los pines digitales 2, 3, 4, 5, 6 y  7
  //como pines de salida
  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);



}

void loop() {
  /*En arduino, esta es la rutina principal;
  desde aqui se haran las llamadas a las rutinas base de
  movimiento del robot, separadas por la rutina de retraso
  delay(numeromilisegundos);
  */
}




}
//rutina de movimiento de MASAYLO hacia adelante
```

```
void adelante(){
  digitalWrite(2, HIGH);
  digitalWrite(3, HIGH);
  digitalWrite(4, LOW);
  digitalWrite(5, HIGH);
  digitalWrite(6, HIGH);
  digitalWrite(7, LOW);
}
//rutina de movimiento de MASAYLO hacia atras
void atras(){
  digitalWrite(2, HIGH);
  digitalWrite(3, LOW);
  digitalWrite(4, HIGH);
  digitalWrite(5, HIGH);
  digitalWrite(6, LOW);
  digitalWrite(7, HIGH);
}
//rutina de movimiento de MASAYLO hacia la derecha
void derecha(){
  digitalWrite(2, LOW);
  digitalWrite(3, HIGH);
  digitalWrite(4, LOW);
  digitalWrite(5, HIGH);
  digitalWrite(6, HIGH);
  digitalWrite(7, LOW);
}
//rutina de movimiento de MASAYLO hacia la izquierda
void izquierda(){
  digitalWrite(2, HIGH);
  digitalWrite(3, HIGH);
  digitalWrite(4, LOW);
  digitalWrite(5, LOW);
  digitalWrite(6, HIGH);
  digitalWrite(7, LOW);
```

```
}
//rutina de detencion inmediata del movimiento de MASAYLO
void alto(){
  digitalWrite(2, LOW);
  digitalWrite(3, HIGH);
  digitalWrite(4, LOW);
  digitalWrite(5, LOW);
  digitalWrite(6, HIGH);
  digitalWrite(7, LOW);
}
```