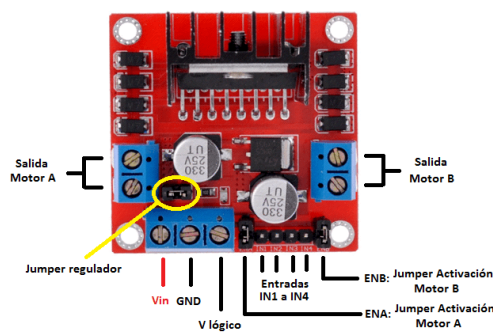


Funcionamiento del motor de corriente continua con el driver L298n

El módulo controlador de motores L298N H-bridge nos permite controlar la velocidad y la dirección de dos motores de corriente continua o un motor paso a paso de una forma muy sencilla, gracias a los 2 los dos H-bridge que monta. Un puente-H o H-bridge es un componente formado por 4 transistores que nos permite invertir el sentido de la corriente, y de esta forma podemos invertir el sentido de giro del motor. El rango de tensiones en el que trabaja este módulo va desde 3V hasta 35V, y una intensidad de hasta 2A. A la hora de alimentarlo hay que tener en cuenta que la electrónica del módulo consume unos 3V, así que los motores reciben 3V menos que la tensión con la que alimentemos el módulo. Además el L298N incluye un regulador de tensión que nos permite obtener del módulo una tensión de 5V, perfecta para alimentar nuestro Arduino. Eso sí, este regulador sólo funciona si alimentamos el módulo con una tensión máxima de 12V. Es un módulo que se utiliza mucho en proyectos de robótica, por su facilidad de uso y su reducido precio.



Conexión eléctrica

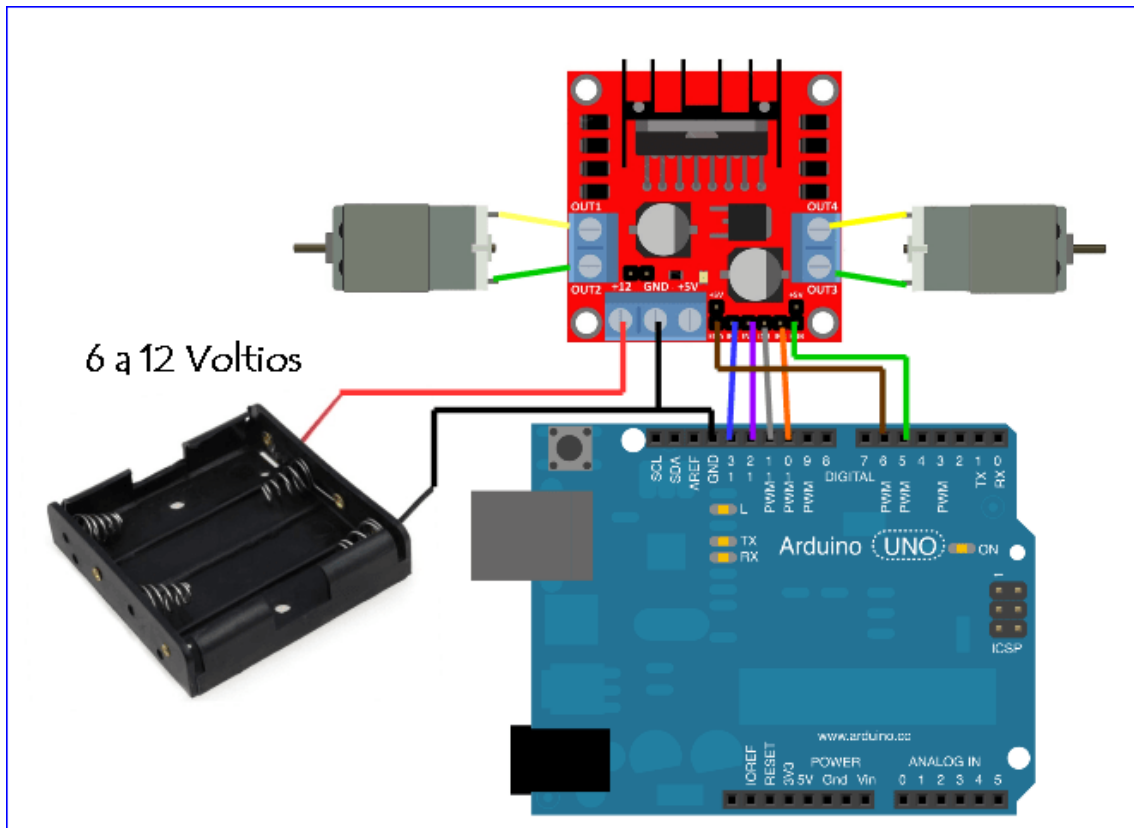
Las salidas para los motores A y B nos darán la energía para mover los motores. Tened en cuenta la polaridad al conectarlos, para que cuando más tarde hagamos que se muevan adelante, funcionen como deberían. Si no fuera así, no tendríamos más que invertir las conexiones.

Los pines **IN1** e **IN2** nos sirven para controlar el sentido de giro del **motor A**, y los pines **IN3** e **IN4** el del **motor B**. Funcionan de forma que si IN1 está a HIGH e IN2 a LOW, el motor A gira en un sentido, y si está IN1 a LOW e IN2 a HIGH lo hace en el otro. Y lo mismo con los pines IN3 e IN4 y el motor B.

Para controlar la velocidad de giro de los motores tenemos que quitar los jumpers y usar los pines **ENA** y **ENB**. Los conectaremos a dos salidas **PWM** de Arduino de forma que le enviemos un valor entre 0 y 255 que controle la velocidad de giro. Si tenemos los jumpers colocados, los motores girarán a la siempre a la misma velocidad.

A continuación realizaremos el esquema en el que alimentaremos el driver L298n y los motores con una fuente de alimentación externa (en nuestro caso 4 pilas AA que nos proporcionan un voltaje de 6 V aunque existirían mejores y

más caras opciones ([leer artículo sobre alimentación de la placa Arduino](#)) por lo que nuestro vehículo se podría mover sin tener que estar conectado al ordenador.

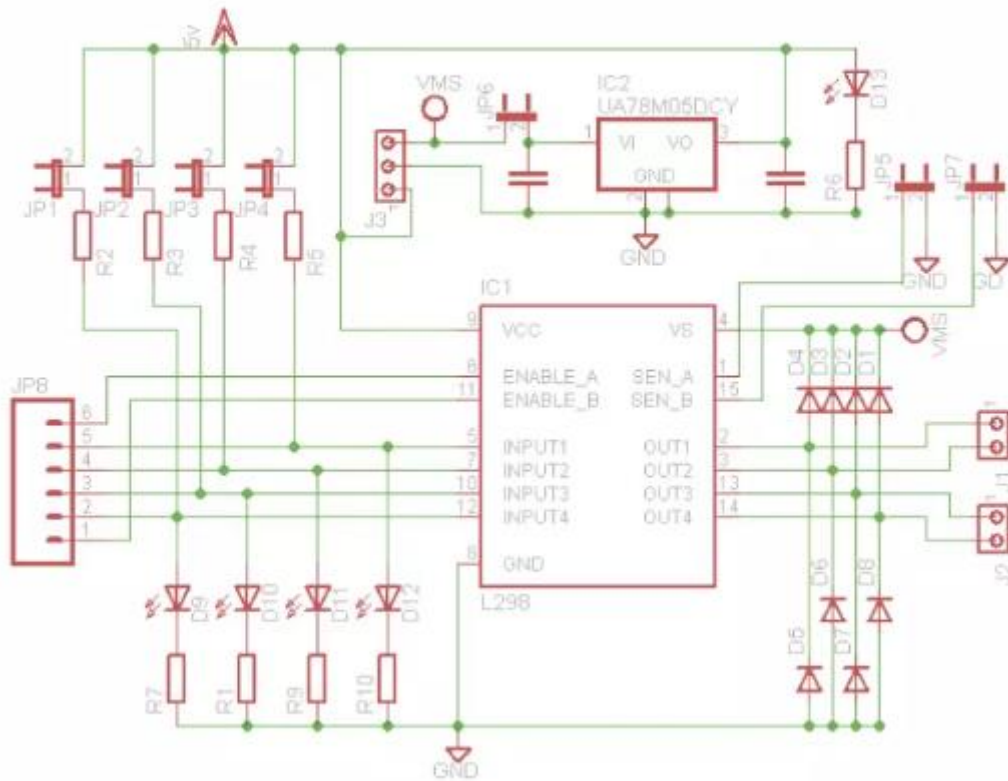


Sensores de infrarrojos TCRT5000

El sensor de infrarrojos nos permitirá detectar la presencia de una línea negra y de esta manera enrutar nuestra trayectoria para poder seguirla



L298 Motor Driver Schemataic Diagram



Lógica del circuito (algoritmo para la programación)

Antes de empezar a programar, vamos a recordar cómo se supone que debería funcionar el robot:

- Si los dos sensores están sobre la línea: sigue recto
- Si se ha salido hacia la izquierda: giramos a la derecha hasta tener los dos sensores sobre la línea
- Si se ha salido hacia la derecha: giramos hacia la izquierda hasta tener los dos sensores sobre la línea
- Si por alguna razón se salieran los dos, haremos que se pare



Código del programa Arduino

A continuación se muestra el código Arduino que implementa la funcionalidad.. Debe tenerse mucho cuidado con la correspondencia izquierda/derecha de los sensores y con el sentido de giro de las ruedas del vehículo...

Presentaremos 2 versiones..

- **En la primera controlaremos la velocidad del motor mediante la activación de los jumpers (ENA, ENB) del driver L298n..**

Arduino Code

```
1.  /*
2.   Nombre: Robot Siguelíneas
3.   Autor: Daniel Pascual Gallegos
4.   Fecha: Febrero 2017
5.   Funcionalidad: este proyecto representa la típica función
6.   del robot siguelíneas que se desplaza siguiendo una línea recta
7.   de color negro.
8.  */
9.  // Definición de variables y constantes relacionadas con el motor
   izquierdo
10.   const int IN1 = 13;  // Pin digital 13 para controlar sentido
   giro motor izquierdo
11.   const int IN2 = 12;  // Pin digital 12 para controlar sentido
   giro motor izquierdo
12.   const int ENA = 6;
13.
14.   // Definición de variables y constantes relacionadas con el
   motor derecho
15.   const int IN3 = 11;  // Pin digital 11 para controlar sentido
   giro motor izquierdo
16.   const int IN4 = 10;  // Pin digital 10 para controlar sentido
   giro motor izquierdo
17.   const int ENB = 5;
18.
```

```

19.     const int vel = 175;
20.
21.     // Definición de variables y constantes relacionadas con los
        sensores IR
22.     int lecturaSensorIzq; // Almacena el valor de la lectura del
        sensor izquierdo
23.     int lecturaSensorDer; // Almacena el valor de la lectura del
        sensor derecho
24.     const int sensorIzqPin = A0; // El sensor izq irá conectado al
        pin analógico A0
25.     const int sensorDerPin = A1; // El sensor derecho irá conectado
        al pin analógico A1
26.
27.     void setup()
28.     {
29.         // Se declaran todos los pines como salidas
30.         // Pines asociados a los motores
31.         pinMode (IN1, OUTPUT);
32.         pinMode (IN2, OUTPUT);
33.         pinMode (IN3, OUTPUT);
34.         pinMode (IN4, OUTPUT);
35.         pinMode (ENA, OUTPUT);
36.         pinMode (ENB, OUTPUT);
37.         pinMode( sensorIzqPin  , INPUT) ;
38.         pinMode( sensorDerPin  , INPUT) ;
39.         Serial.begin(9600); // Se inicia el puerto de comunicaciones
        en serie
40.     }
41.
42.     void loop()
43.     {
44.
45.         lecturaSensorIR(); // Se lee el valor de los sensores IR
46.         // Se analiza el resultado de los sensores para hacer que el
        robot siga la línea negra
47.
48.         // Si el resultado de ambos sensores es 0 (zona blanca) el
        robot sigue se para
49.         if(lecturaSensorIzq == 0 && lecturaSensorDer == 0)
50.         {
51.             robotParar(); // El robot para
52.
53.         }
54.         // Si el izquierdo retorna 0 (zona blanca) y el derecho 1
        (negra) el robot gira derecha
55.         if (lecturaSensorIzq == 0 && lecturaSensorDer == 1)
56.         {
57.             robotDerecha();

```

```

58.          // El robot gira a la derecha
59.
60.      }
61.      // Si el izquierdo retorna 1 (zona negra) y el derecho 0
        (blanca) el robot gira izquierda
62.      if (lecturaSensorIzq == 1 && lecturaSensorDer == 0)
63.      {
64.          robotIzquierda();
65.
66.      }
67.      // Si ambos sensores retornan 0 (zona negra) el robot sigue
        recto
68.      if (lecturaSensorIzq == 1 && lecturaSensorDer == 1)
69.      {
70.          robotAvance(); // El robot avanza
71.          Serial.println("robot avanza");
72.      }
73.
74.  }
75.  /*
76.      Función lecturaSensorIR: leerá el valor del sensor de
        infrarrojos TCRT5000
77.      y lo almacena en una variable. Dicho sensor retornará el
        valor 0 (LOW) si
78.      el sensor está en zona blanca y el valor 1 (HIGH) si el
        sensor está en zona
79.      negra.
80.  */
81.  void lecturaSensorIR()
82.  {
83.      lecturaSensorIzq = digitalRead(sensorIzqPin); // Almacena la
        lectura del sensor izquierdo
84.      lecturaSensorDer = digitalRead(sensorDerPin); // Almacena la
        lectura del sensor derecho
85.
86.      Serial.println("El valor del sensor izquierdo es ");
87.      Serial.println(lecturaSensorIzq);
88.
89.      Serial.println("El valor del sensor derecho es ");
90.      Serial.println(lecturaSensorDer);
91.
92.  }
93.  /*
94.      Función robotAvance: esta función hará que ambos motores se
        activen a máxima potencia
95.      por lo que el robot avanzará hacia delante
96.  */
97.  void robotAvance()

```

```

98.     {
99.         // Motor izquierdo
100.        // Al mantener un pin HIGH y el otro LOW el motor gira en un
            sentido
101.        digitalWrite (IN1, HIGH);
102.        digitalWrite (IN2, LOW);
103.        analogWrite (ENA, vel); //Velocidad motor A
104.
105.        // Motor derecho
106.        // Al mantener un pin HIGH y el otro LOW el motor gira en un
            sentido
107.        digitalWrite (IN3, HIGH);
108.        digitalWrite (IN4, LOW);
109.        analogWrite (ENB, vel); //Velocidad motor B
110.    }
111.    /*
112.        Función robotRetroceso: esta función hará que ambos motores
            se activen a máxima potencia
113.        en sentido contrario al anterior por lo que el robot avanzará
            hacia atrás
114.    */
115.    void robotRetroceso()
116.    {
117.        // Motor izquierdo
118.        // Al mantener un pin LOW y el otro HIGH el motor gira en
            sentido contrario al anterior
119.        digitalWrite (IN1, LOW);
120.        digitalWrite (IN2, HIGH);
121.        analogWrite (ENA, vel); //Velocidad motor A
122.
123.        // Motor derecho
124.        // Al mantener un pin LOW y el otro HIGH el motor gira en
            sentido contrario al anterior
125.        digitalWrite (IN3, LOW);
126.        digitalWrite (IN4, HIGH);
127.        analogWrite (ENB, vel); //Velocidad motor B
128.    }
129.
130.    /*
131.        Función robotDerecha: esta función accionará el motor
            izquierdo y parará el derecho
132.        por lo que el coche girará hacia la derecha (sentido horario)
133.    */
134.    void robotDerecha()
135.    {
136.        // Motor izquierdo
137.        // Se activa el motor izquierdo
138.        digitalWrite (IN1, HIGH);

```

```

139.     digitalWrite (IN2, LOW);
140.     analogWrite (ENA, vel); //Velocidad motor A
141.
142.     // Motor derecho
143.     // Se para el motor derecho
144.     digitalWrite (IN3, LOW);
145.     digitalWrite (IN4, HIGH);
146.     analogWrite (ENB, vel); //Velocidad motor A
147.
148. }
149. /*
150.     Función robotIzquierda: esta función accionará el motor
        derecho y parará el izquierdo
151.     por lo que el coche girará hacia la izquierda (sentido
        antihorario)
152. */
153. void robotIzquierda ()
154. {
155.     // Motor izquierdo
156.     // Se para el motor izquierdo
157.     digitalWrite (IN1, LOW);
158.     digitalWrite (IN2, HIGH);
159.     analogWrite (ENA, vel); //Velocidad motor A
160.
161.     // Motor derecho
162.     // Se activa el motor derecho
163.     digitalWrite (IN3, HIGH);
164.     digitalWrite (IN4, LOW);
165.     analogWrite (ENB, vel); //Velocidad motor A
166. }
167. /*
168.     Función robotParar: esta función parará ambos motores
169.     por lo que el robot se parará.
170. */
171. void robotParar()
172. {
173.     // Motor izquierdo
174.     // Se para el motor izquierdo
175.     digitalWrite (IN1, LOW);
176.     digitalWrite (IN2, LOW);
177.
178.     // Motor derecho
179.     // Se para el motor derecho
180.     digitalWrite (IN3, LOW);
181.     digitalWrite (IN4, LOW);
182.
183. }

```


184.

- **En la segunda no controlaremos la velocidad del motor y no estarán activados los jumpers (ENA, ENB) del driver L298n..**

Arduino Code

```
1.
2.  /*
3.   Nombre: Robot Siguelíneas
4.   Autor: Daniel Pascual Gallegos
5.   Fecha: Febrero 2017
6.   Funcionalidad: este proyecto representa la típica función
7.   del robot siguelíneas que se desplaza siguiendo una línea recta
8.   de color negro.
9.  */
10. // Definición de variables y constantes relacionadas con el
    motor izquierdo
11.   const int IN1 = 13; // Pin digital 13 para controlar sentido
    giro motor izquierdo
12.   const int IN2 = 12; // Pin digital 12 para controlar sentido
    giro motor izquierdo
13.
14. // Definición de variables y constantes relacionadas con el
    motor derecho
15.   const int IN3 = 11; // Pin digital 11 para controlar sentido
    giro motor izquierdo
16.   const int IN4 = 10; // Pin digital 10 para controlar sentido
    giro motor izquierdo
17.
18.
19. // Definición de variables y constantes relacionadas con los
    sensores IR
20.   int lecturaSensorIzq; // Almacena el valor de la lectura del
    sensor izquierdo
21.   int lecturaSensorDer; // Almacena el valor de la lectura del
    sensor derecho
22.   const int sensorIzqPin = A1; // El sensor izq irá conectado al
    pin analógico A0
23.   const int sensorDerPin = A0 // El sensor derecho
    irá conectado al pin analógico A1
24.
25.   void setup()
26.   {
27.       // Se declaran todos los pines como salidas
28.       // Pines asociados a los motores
```

```

29.     pinMode (IN1, OUTPUT);
30.     pinMode (IN2, OUTPUT);
31.     pinMode (IN3, OUTPUT);
32.     pinMode (IN4, OUTPUT);
33.     pinMode( sensorIzqPin  , INPUT) ;
34.     pinMode( sensorDerPin  , INPUT) ;
35.     Serial.begin(9600); // Se inicia el puerto de comunicaciones
    en serie
36. }
37.
38. void loop()
39. {
40.
41.     lecturaSensorIR(); // Se lee el valor de los sensores IR
42.     // Se analiza el resultado de los sensores para hacer que el
    robot siga la línea negra
43.
44.     // Si el resultado de ambos sensores es 0 (zona blanca) el
    robot sigue se para
45.     if(lecturaSensorIzq == 0 && lecturaSensorDer == 0)
46.     {
47.         robotParar(); // El robot para
48.
49.     }
50.     // Si el izquierdo retorna 0 (zona blanca) y el derecho 1
    (negra) el robot gira derecha
51.     if (lecturaSensorIzq == 0 && lecturaSensorDer == 1)
52.     {
53.         robotDerecha();
54.         // El robot gira a la derecha
55.
56.     }
57.     // Si el izquierdo retorna 1 (zona negra) y el derecho 0
    (blanca) el robot gira izquierda
58.     if (lecturaSensorIzq == 1 && lecturaSensorDer == 0)
59.     {
60.         robotIzquierda();
61.
62.     }
63.     // Si ambos sensores retornan 0 (zona negra) el robot sigue
    recto
64.     if (lecturaSensorIzq == 1 && lecturaSensorDer == 1)
65.     {
66.         robotAvance(); // El robot avanza
67.         Serial.println("robot avanza");
68.     }
69.
70. }

```

```

71.    /*
72.        Función lecturaSensorIR: leerá el valor del sensor de
           infrarrojos TCRT5000
73.        y lo almacena en una variable. Dicho sensor retornará el
           valor 0 (LOW) si
74.        el sensor está en zona blanca y el valor 1 (HIGH) si el
           sensor está en zona
75.        negra.
76.    */
77.    void lecturaSensorIR()
78.    {
79.        lecturaSensorIzq = digitalRead(sensorIzqPin); // Almacena la
           lectura del sensor izquierdo
80.        lecturaSensorDer = digitalRead(sensorDerPin); // Almacena la
           lectura del sensor derecho
81.
82.        Serial.println("El valor del sensor izquierdo es ");
83.        Serial.println(lecturaSensorIzq);
84.
85.        Serial.println("El valor del sensor derecho es ");
86.        Serial.println(lecturaSensorDer);
87.
88.    }
89.    /*
90.        Función robotAvance: esta función hará que ambos motores se
           activen a máxima potencia
91.        por lo que el robot avanzará hacia delante
92.    */
93.    void robotAvance()
94.    {
95.        // Motor izquierdo
96.        // Al mantener un pin HIGH y el otro LOW el motor gira en un
           sentido
97.        digitalWrite (IN1, HIGH);
98.        digitalWrite (IN2, LOW);
99.
100.        // Motor derecho
101.        // Al mantener un pin HIGH y el otro LOW el motor gira en un
           sentido
102.        digitalWrite (IN3, HIGH);
103.        digitalWrite (IN4, LOW);
104.    }
105.    /*
106.        Función robotRetroceso: esta función hará que ambos motores
           se activen a máxima potencia
107.        en sentido contrario al anterior por lo que el robot avanzará
           hacia atrás
108.    */

```

```

109. void robotRetroceso()
110. {
111.     // Motor izquierdo
112.     // Al mantener un pin LOW y el otro HIGH el motor gira en
        sentido contrario al anterior
113.     digitalWrite (IN1, LOW);
114.     digitalWrite (IN2, HIGH);
115.
116.     // Motor derecho
117.     // Al mantener un pin LOW y el otro HIGH el motor gira en
        sentido contrario al anterior
118.     digitalWrite (IN3, LOW);
119.     digitalWrite (IN4, HIGH);
120. }
121.
122. /*
123.     Función robotDerecha: esta función accionará el motor
        izquierdo y parará el derecho
124.     por lo que el coche girará hacia la derecha (sentido horario)
125. */
126. void robotDerecha()
127. {
128.     // Motor izquierdo
129.     // Se activa el motor izquierdo
130.     digitalWrite (IN1, HIGH);
131.     digitalWrite (IN2, LOW);
132.
133.     // Motor derecho
134.     // Se para el motor derecho
135.     digitalWrite (IN3, LOW);
136.     digitalWrite (IN4, HIGH);
137.
138. }
139. /*
140.     Función robotIzquierda: esta función accionará el motor
        derecho y parará el izquierdo
141.     por lo que el coche girará hacia la izquierda (sentido
        antihorario)
142. */
143. void robotIzquierda ()
144. {
145.     // Motor izquierdo
146.     // Se para el motor izquierdo
147.     digitalWrite (IN1, LOW);
148.     digitalWrite (IN2, HIGH);
149.
150.     // Motor derecho
151.     // Se activa el motor derecho

```

```

152.     digitalWrite (IN3, HIGH);
153.     digitalWrite (IN4, LOW);
154. }
155. /*
156.     Función robotParar: esta función parará ambos motores
157.     por lo que el robot se parará.
158. */
159. void robotParar()
160. {
161.     // Motor izquierdo
162.     // Se para el motor izquierdo
163.     digitalWrite (IN1, LOW);
164.     digitalWrite (IN2, LOW);
165.
166.     // Motor derecho
167.     // Se para el motor derecho
168.     digitalWrite (IN3, LOW);
169.     digitalWrite (IN4, LOW);
170.
171. }
172.
173.

```

