

INSTRUCCIONES DE MONTAJE ROBOT Z-K2



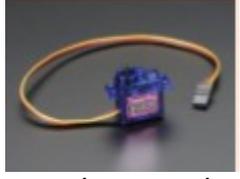
TITULO

LISTADO.....	2
MONTAJE.....	5
1º paso: Conectar los cables a los motores.....	5
2º paso: Montar los motores en el chasis.....	5
3º Montar la rueda loca:.....	7
4º Colocar las ruedas de los motores:.....	8
5º Montar el interruptor:.....	9
6º Atornillar el soporte para pilas:.....	9
CONTROLADOR DE MOTORES L298N.....	10
MONTAJE DEL ARDUINO Y DEL L298N.....	12
MOMTAR SERVO MOTOR (SG90).....	15
CONEXIONES ELÉCTRICAS.....	16
SENSOR SHIELD V5.0 PARA Arduino.....	17

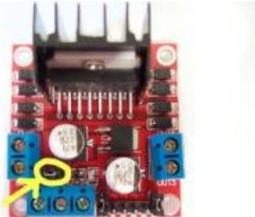
Montaje del sensor de ultrasonidos y programación del vehículo

LISTADO

¡Asegúrese de que las siguientes piezas estén incluidas en su kit!

Arduino Sensor Shield	1		Por favor, vea la nota sobre los puentes debajo.
L298N Dual H-Bridge Motor Driver . Placa de control de motores	1		
Arduino Uno	1		
Cabezales rompibles de 40 pines	1		No se requiere para este kit de regalo gratis
Soporte de montaje servo	1		Ignora la placa superior provista
Base servo	1		Viene en la misma bolsa que los soportes de montaje Servo
Servo Motor (SG90)		 Para poder usar el servo, incorporamos al	Motor CC de precisión tiene en su interior un motor con reductora de velocidad y un pequeño circuito de control que solamente permite giros desde 0°

		principio del programa la instrucción #include <Servo.h>	a 180°. Tiene tres cables que debemos conectar así: uno a +5v, otro a GND y otro a un pin digita
Cable USB	1		El color y la longitud pueden variar
Sensor Ultrasónico HC-SR04	2		
Motor de engranaje	2		
Ruedas de goma. 2.51"	2		<i>Viene envuelto con Motores</i>
Chasis de robot de acrílico	1		<i>Viene envuelto con Motores</i>
Soporte de batería AA	1		

Soportes de montaje del motor	1		
Rueda loca	1		
Interruptor encendido / apagado	1		
<p>Asegúrese de que tanto el tablero del motor como el protector del sensor muestren el puente indicado con un círculo amarillo. Si a los pines les falta el puente, siéntase libre de acortar los pines doblándolos para que se toquen entre sí o utilice un cable de tablero.</p> <div style="display: flex; justify-content: space-around;">   </div>			

MONTAJE

Aunque el set de piezas del coche viene con unas instrucciones de montaje, vamos a intentar desarrollarlo un poquito más para intentar clarificarlo lo más posible. Es muy sencillo y seguro que no tenéis ningún problema.

1º paso: Conectar los cables a los motores

Lo primero que haremos será conectar los cables rojos y negros a los dos motores amarillos. Podríamos también hacerlo después de montar los motores en el chasis del coche, pero al menos a nosotros, nos resulta más sencillo hacerlo antes de colocarlos en el chasis.

Los cables se conectan en los dos bornes metálicos que tienen los motores. La mejor manera es soldarlos, pero si no tenéis soldador podéis pelar un poquito más el cable y después de atarlo lo mejor posible a los bornes ponerle un poquito de cinta aislante para que no se suelten, pero asegurándonos de que el cable no deje de hacer contacto con el borne metálico en ningún momento.

No hace falta preocuparse por el orden en el que conectamos los cables a los bornes, porque luego podemos cambiar el sentido de giro de los motores tanto en la conexión al controlador de motores como en el programa que haremos para controlarlos.

Sí es más importante que los cables salgan hacia el lado del motor, porque luego al conectarlos irán hacia ese lado.



2º paso: Montar los motores en el chasis

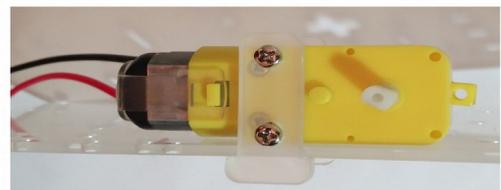
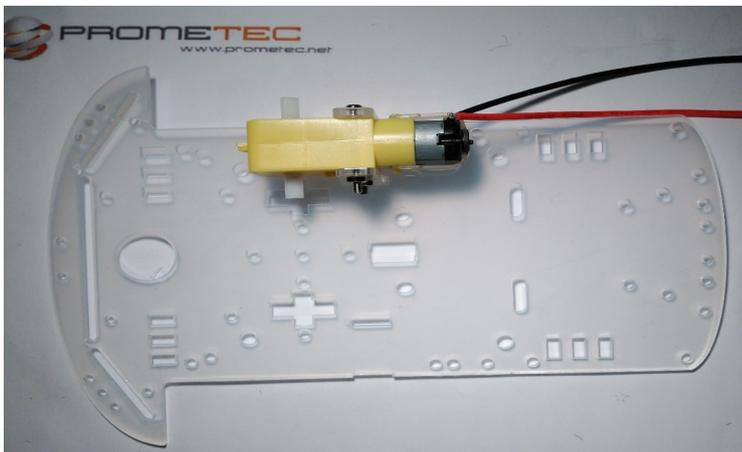
Para sujetar los motores al chasis vamos a necesitar las siguientes piezas:



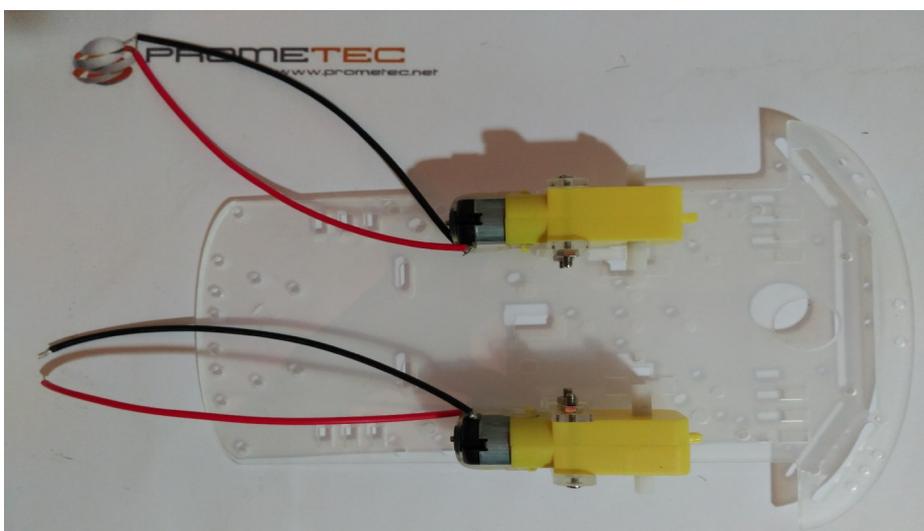
Colocamos una de las piezas de plástico en uno de los dos agujeros rectangulares que hay en la zona central del chasis, y otra en la parte exterior que tiene una hendidura paralela a la que hemos colocado en el interior, con el motor en medio de ellas

Atravesamos con los tornillos los agujeros de las piezas de plástico haciéndolos coincidir con los del motor. Os recomendamos:

- Dejar los cables hacia el interior, y la parte donde están las soldaduras hacia el lado del chasis que no tiene esa especie de parachoques.
 - Las tuercas las colocaremos en el lado interior, porque hacia el exterior rozarían con las ruedas.
 - Colocar primero el tornillo y la tuerca más cercana al chasis antes de meter el otro tornillo, para que nos sea más sencillo poner la tuerca que queda bastante pegada al chasis de plástico.
- Debería quedaros de esta forma:

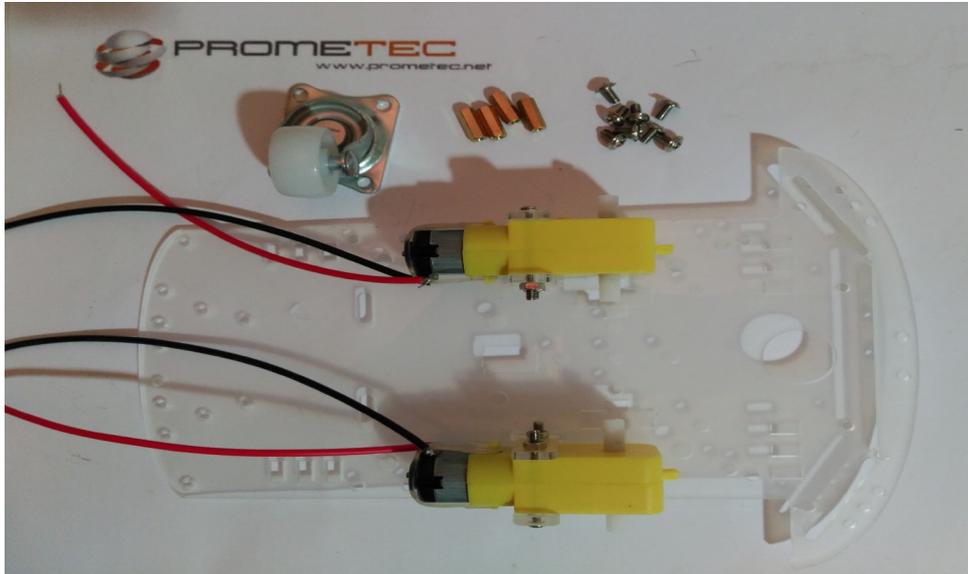


Y hacemos lo mismo con el otro motor.

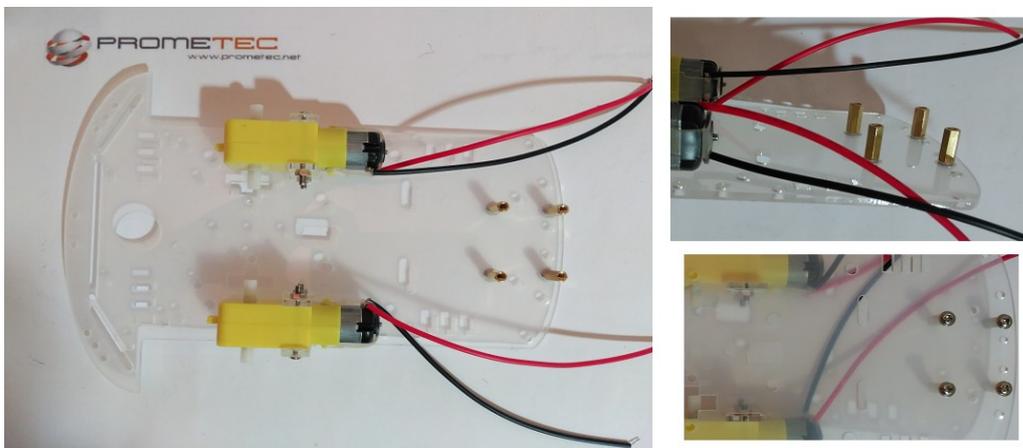


3º Montar la rueda loca:

A continuación vamos a montar lo que se llama rueda loca. Es una rueda sin tracción que puede girar libremente. Para este paso vamos a necesitar los 4 hexágonos dorados, 8 de los tornillos más cortos y la parte del chasis que ya tenemos montada.



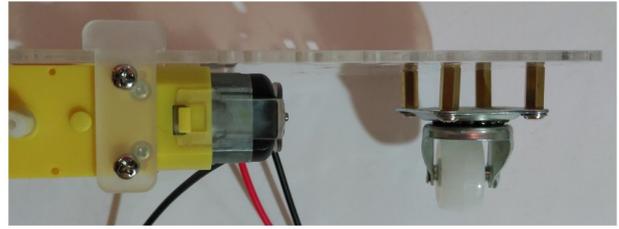
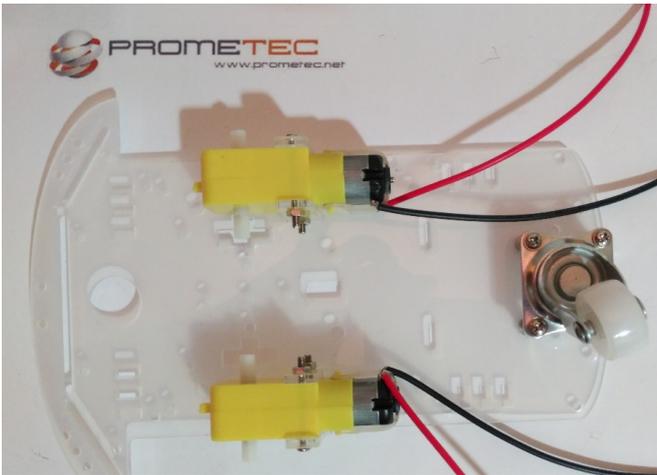
Empezamos atornillando los hexágonos dorados al chasis con cuatro de los tornillos, de forma que los hexágonos queden hacia el lado de los motores.



Ahora colocamos la rueda loca en los hexágonos y la atornillamos con cuatro tornillos.

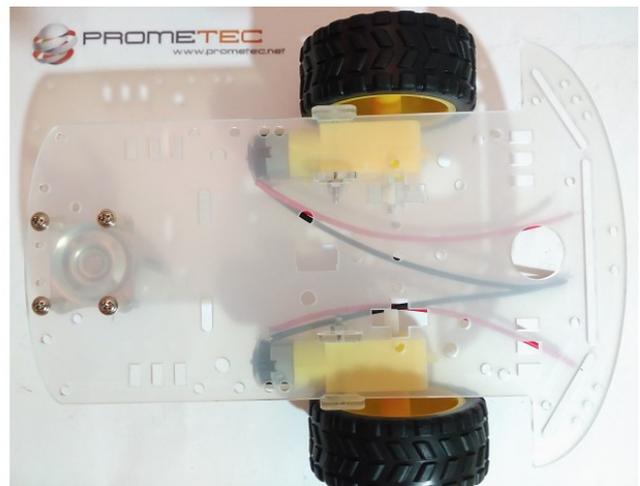
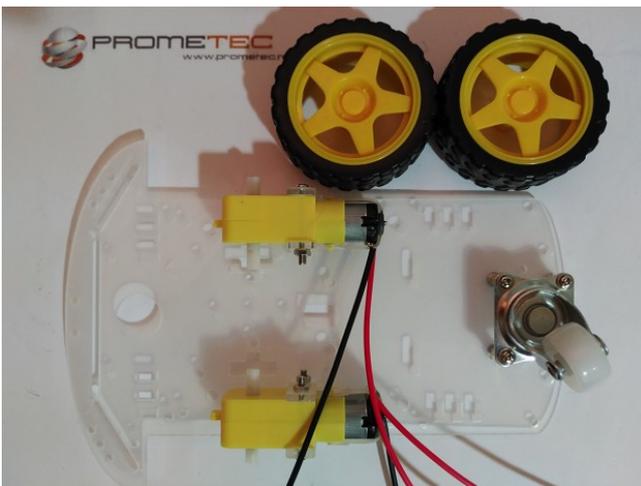
La base de la rueda no es cuadrada, sino rectangular así que sólo encaja en un sentido.

A la hora de colocar los tornillos, es mejor no apretarlos del todo hasta que estén los cuatro bien puestos, para que nos sea más sencillo hacer coincidir los tornillos con los hexágonos a través de la base de la rueda.



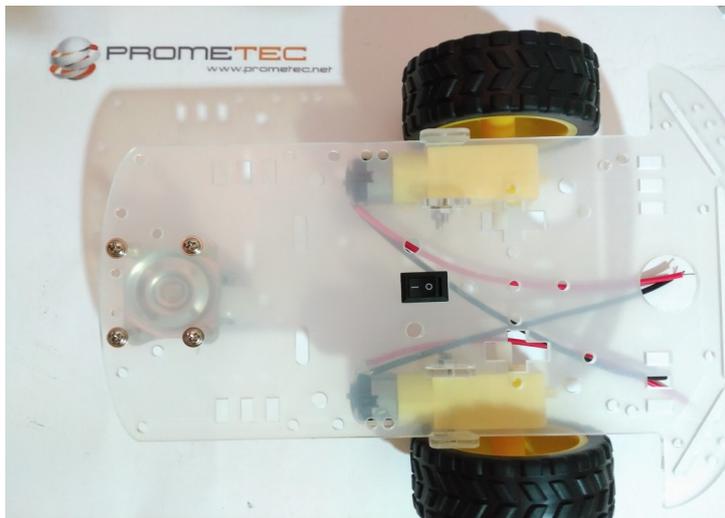
4º Colocar las ruedas de los motores:

Simplemente tenemos que colocar las ruedas en la parte exterior de los motores, haciendo coincidir la forma del agujero de las ruedas con las de la parte blanca rotativa del motor.



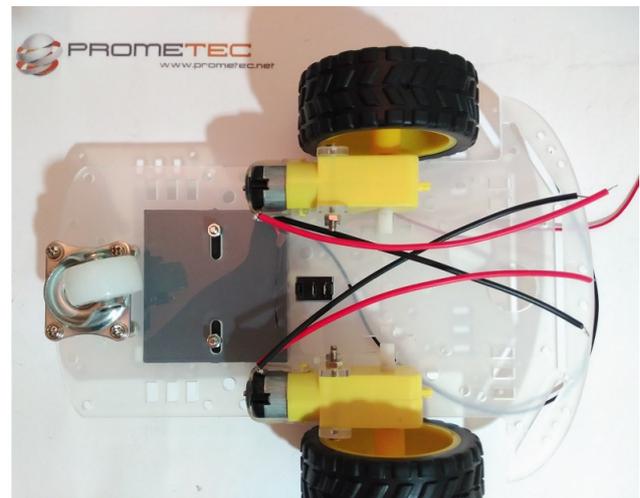
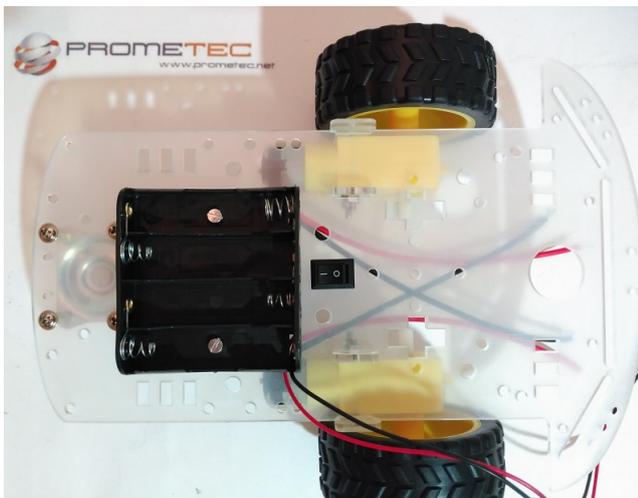
5º Montar el interruptor:

Vamos a utilizar un interruptor para poder encender y apagar el coche de la forma más sencilla posible. Sólo tenemos que colocarlo en el agujero que hay en el centro del coche, forzándolo un poquito a que entre y dejando el interruptor hacia la parte de arriba.



6º Atornillar el soporte para pilas:

Para el soporte necesitamos dos tornillos un poquito más largos, para que puedan atravesar el chasis. Los colocaremos en los dos agujeros alargados que hay entre el interruptor y la rueda loca. Sólo tenemos que meterlos por los agujeros del propio soporte y poner las tuercas para sujetarlo al chasis.



Y con esto ya hemos terminado el montaje del chasis que nos servirá de soporte para ir colocando los componentes electrónicos que usemos en la construcción del coche.

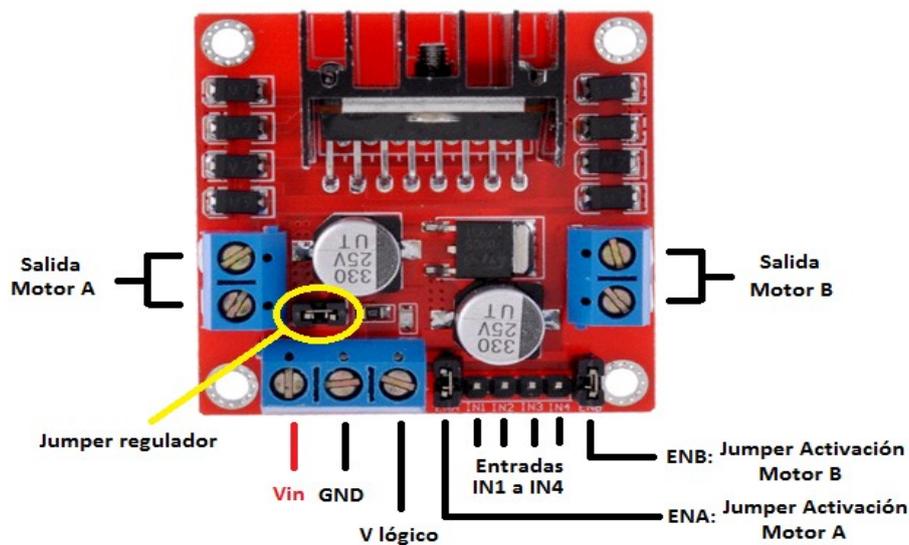
CONTROLADOR DE MOTORES L298N

Cuando queramos trabajar con motores un poquito potentes vamos a tener que utilizar unos componentes a los que llamamos **controladores de motores** o drivers. Estos controladores nos permitirán suministrar la potencia necesaria a los motores y los controlaremos utilizando nuestros Arduinos.

Hay muchísimos controladores diferentes, pero nosotros nos hemos decidido por el **L298N** por su facilidad de uso y su reducido precio, y porque nos permite elegir los pines del Arduino que queremos usar para controlar los motores. El **L298N** puede trabajar con tensiones hasta 35V y una intensidad de hasta 2A por canal por canal, con lo que puede manejar hasta 4A en total y unos 25W.

Este módulo tiene además otra característica muy importante, y es que nos permite controlar la **dirección** de giro de los motores, es decir podremos hacer que giren en ambos sentidos, algo totalmente necesario para nuestro coche.

Además nos permite obtener del módulo una tensión de 5V, perfecta para alimentar nuestro Arduino sin tener que utilizar otra fuente adicional. Eso sí, este regulador sólo funciona si alimentamos el módulo con una tensión máxima de 12V. En nuestro caso son 4 pilas de 1.5V, o sea, 6V.



*Un **jumper** es un componente que nos sirve para conectar dos pines*

La entrada de tensión **Vin** admite tensiones entre 3V y 35V, y justo a su derecha en la imagen tenemos el pin que debemos conectar a **GND**.

La tercera conexión de ese grupo **V lógico** puede funcionar de dos maneras:

Si el **jumper del regulador** está **cerrado** activaremos el regulador de tensión del **L298N**, y en **V lógico** tendremos una salida de 5V, que podremos usar para lo que queramos, por ejemplo para alimentar una placa Arduino.

Si el **quitamos** el **jumper** desactivaremos el regulador, necesitaremos alimentar la parte lógica del módulo, así que tendremos que meter una tensión de 5V por la conexión **V lógico** para que el módulo funcione.

¡Cuidado! Si introducimos corriente por V lógico con el jumper de regulación puesto podríamos dañar el módulo.

Además el **regulador** sólo funciona con tensiones hasta **12V** en **Vin**, por encima de este valor tendremos que quitar el jumper y alimentar la parte lógica del módulo desde otra fuente.

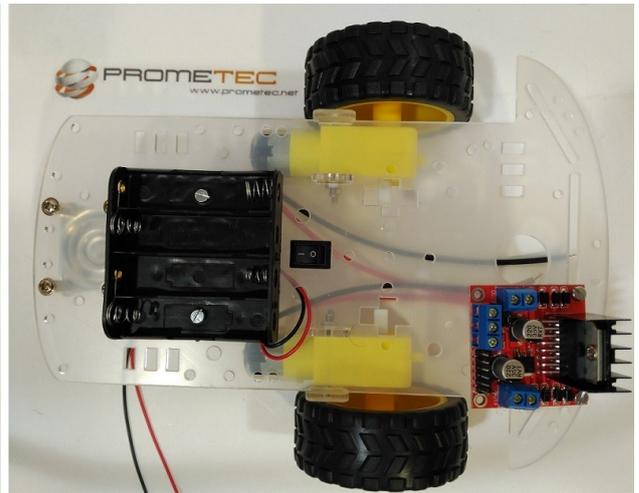
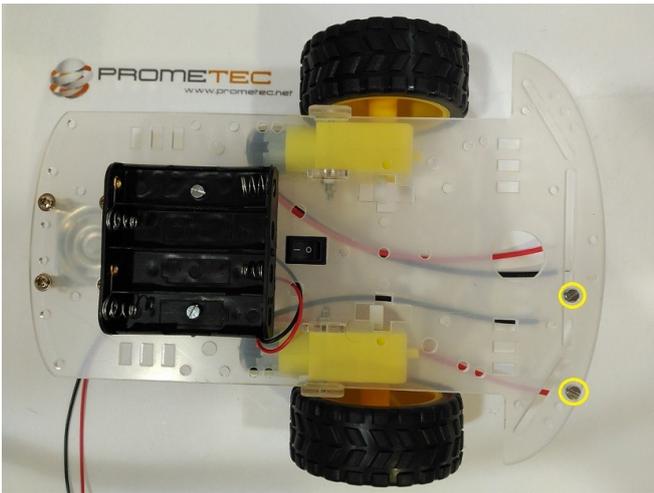
Los pines **IN1** e **IN2** nos sirven para controlar el sentido de giro del **motor A**, y los pines **IN3** e **IN4** el del **motor B**. Funcionan de forma que si IN1 está a HIGH e IN2 a LOW, el motor A gira en un sentido, y si está IN1 a LOW e IN2 a HIGH lo hace en el otro. Y lo mismo con los pines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

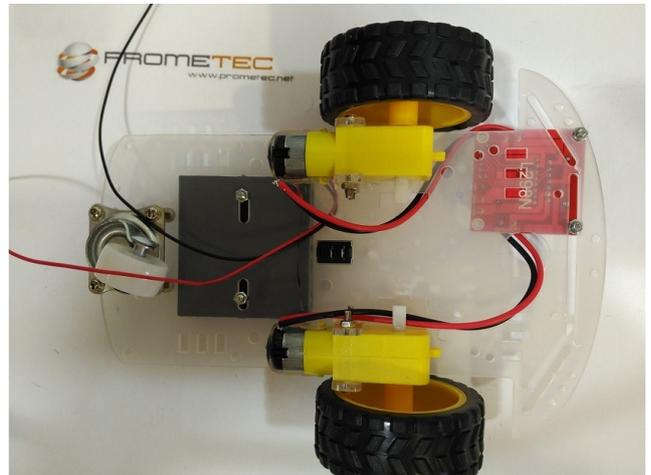
MONTAJE DEL ARDUINO Y DEL L298N

Con el chasis montado vamos a colocar el Arduino y el controlador de motores **L298N**. No tenéis por qué hacerlo igual, pero es recomendable el lugar y el orden que se dicta a continuación para montarlo.

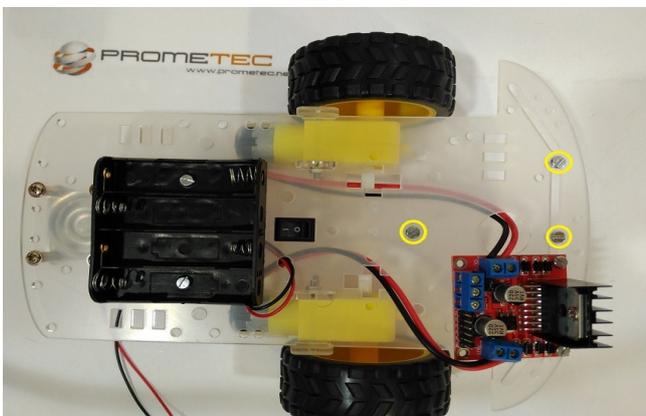
Con eso en mente os recomendamos montar primero el L298N, para poder conectarle los cables de los motores sin que nos moleste el **Arduino**. Lo colocaremos en esta posición mediante 2 tornillos (*de los dos tipos que nos quedan, los más largos*):



Y a continuación conectaremos los cables de los **motores** a los bornes de alimentación de los motores. No os preocupéis demasiado por la posición en la que conectéis el cable negro y el blanco. De ello dependerá hacia donde gire el motor, pero si gira en el sentido que no esperamos hay varias maneras de corregirlo, y así lo haremos más adelante.



Una vez conectados vamos a colocar el Arduino justo encima del controlador utilizando 3 tornillos (de los largos también):



MONTAR SERVO MOTOR (SG90)

En el quinto paso montaremos el servo y el sensor ultrasónico. Primero colocamos y atornillamos la pieza blanca en forma de cruz dentro de la parte redonda de esta pieza:



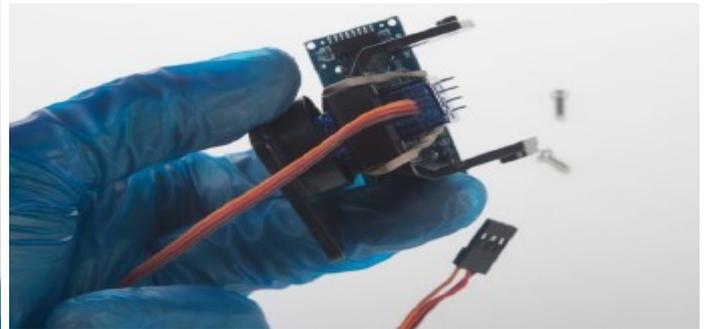
Ponemos por el lado opuesto dos tornillos con dos tuercas, e insertamos después el servo en la parte donde está colocada la pieza blanca en forma de cruz



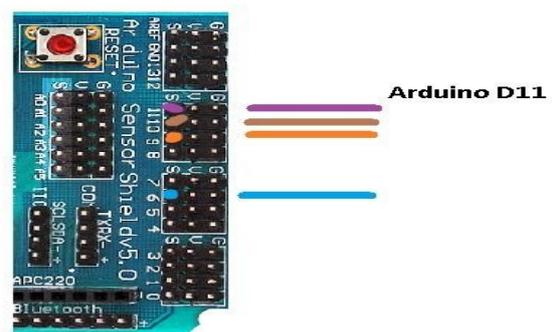
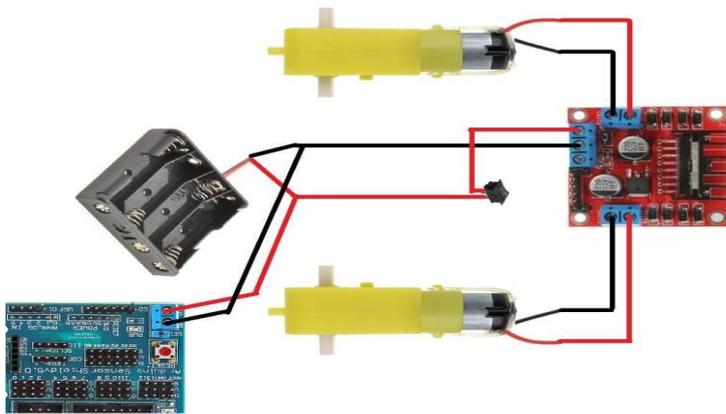
Después encajamos en los dos laterales las piezas que faltan y las atornillamos con dos tornillos:

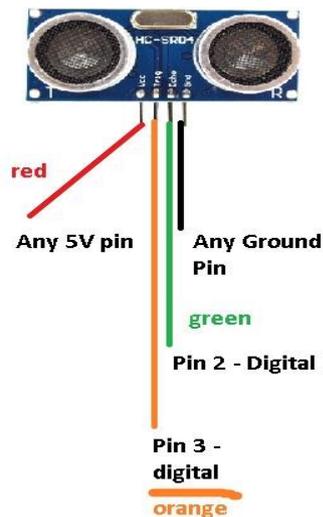


Seguidamente sujetamos el sensor ultrasónico con una goma elástica. El sistema de la goma elástica que contiene las instrucciones nos parece chapucero y endeble, la verdad. Nos preguntamos si eso es habitual en los productos de robótica para principiantes.



CONEXIONES ELÉCTRICAS





SENSOR SHIELD V5.0 PARA Arduino

Usando este shield podemos conectar rápidamente sensores, servos, pulsadores, potenciómetros, módulos I2C, etc

Descripción del producto.

Usando este shield podemos conectar rápidamente sensores, servos, pulsadores, potenciómetros, módulos I2C, etc.

De esta forma ahorramos tiempo en verificar conexiones y depurar errores, tiempo que se aprovecha en mejorar el proyecto.

El sensor shield posee conectores con 3 pines: Alimentación, Tierra y Señal. Existen conectores para entradas analógicas y conectores para I/O digitales.

Destacamos...

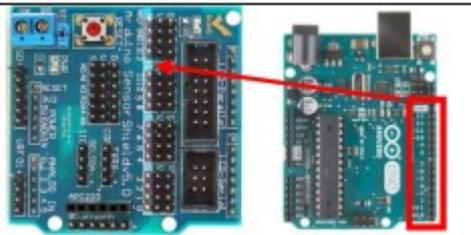
- * Listo para Plug & Play.
- * Se puede conectar a varios modulos.
- * Cada módulo funcional tiene el puerto de plegado con VCC, GND y salida.

El puente de selección permite que la tensión del bloque de terminales entre y se regule a 5 VCC para el microcontrolador Arduino. Si el puente no está presente, debe conectar un conector de cd al puerto de cd de Arduino.



La columna resaltada se conoce como el bus 5v. En lugar de tener múltiples servos o sensores conectados a un solo puerto de 5 VCC, este bus reduce la congestión y hace que las conexiones sean convenientes.

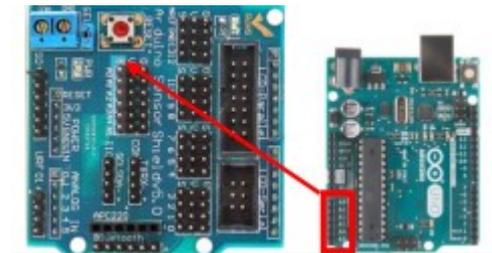
El mismo principio que el bus de 5V, se llama el bus de tierra.



El pinout en la imagen de la izquierda resaltada es equivalente a la imagen de la derecha con el cuadro rojo. Esto se utiliza para sensores digitales.

highlighted section is for analog signals and

La sección resaltada es para señales analógicas y es equivalente a la imagen de la derecha con el cuadro rojo.



El servo dispone de tres cables, dos de alimentación (GND y Vcc) y uno de señal (Sig).

El color de estos cables suele tener dos combinaciones:

- Marrón (GND), Rojo (Vcc) y Naranja (Sig)
- Negro (GND), Rojo (Vcc) y Blanco (Sig)

En general, la alimentación a los servos se realizará desde una fuente de tensión externa (una batería o fuente de alimentación) a una tensión de 5V-6.5V, siendo 6V la tensión idónea.

Arduino puede llegar a proporcionar corriente suficiente para encender un servo pequeño (SG90) .

Por otro lado, finalmente, para el control conectamos el cable de señal (naranja / blanco) a cualquier pin digital de Arduino.

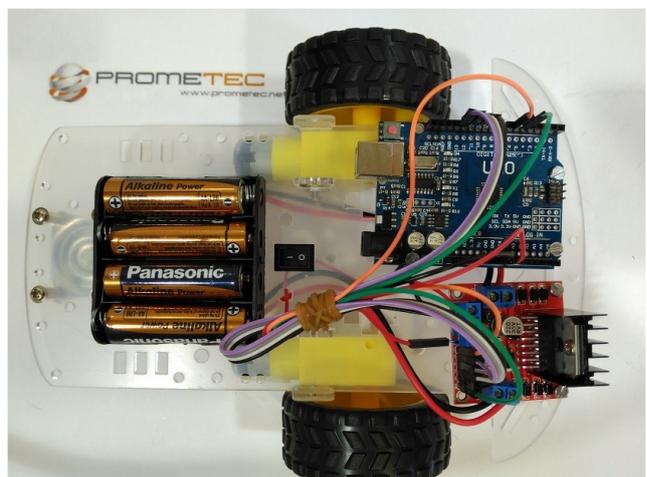
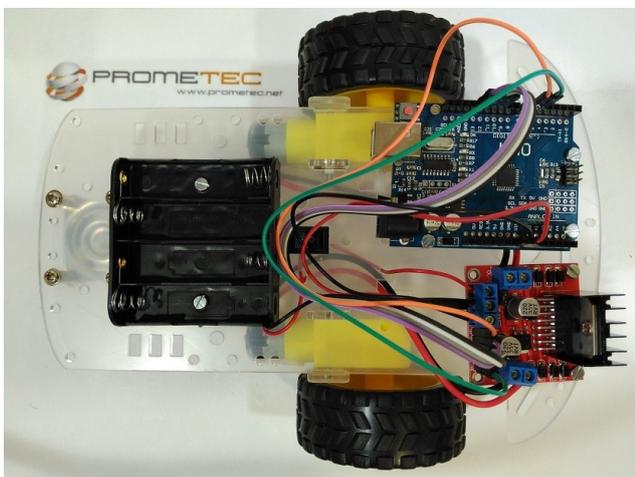
Para ello incrementa el ángulo de 0 a 180° a razón de 1° cada 15ms, posteriormente realiza la operación contraria de 180° a 0°, para finalmente reiniciar el bucle.

```

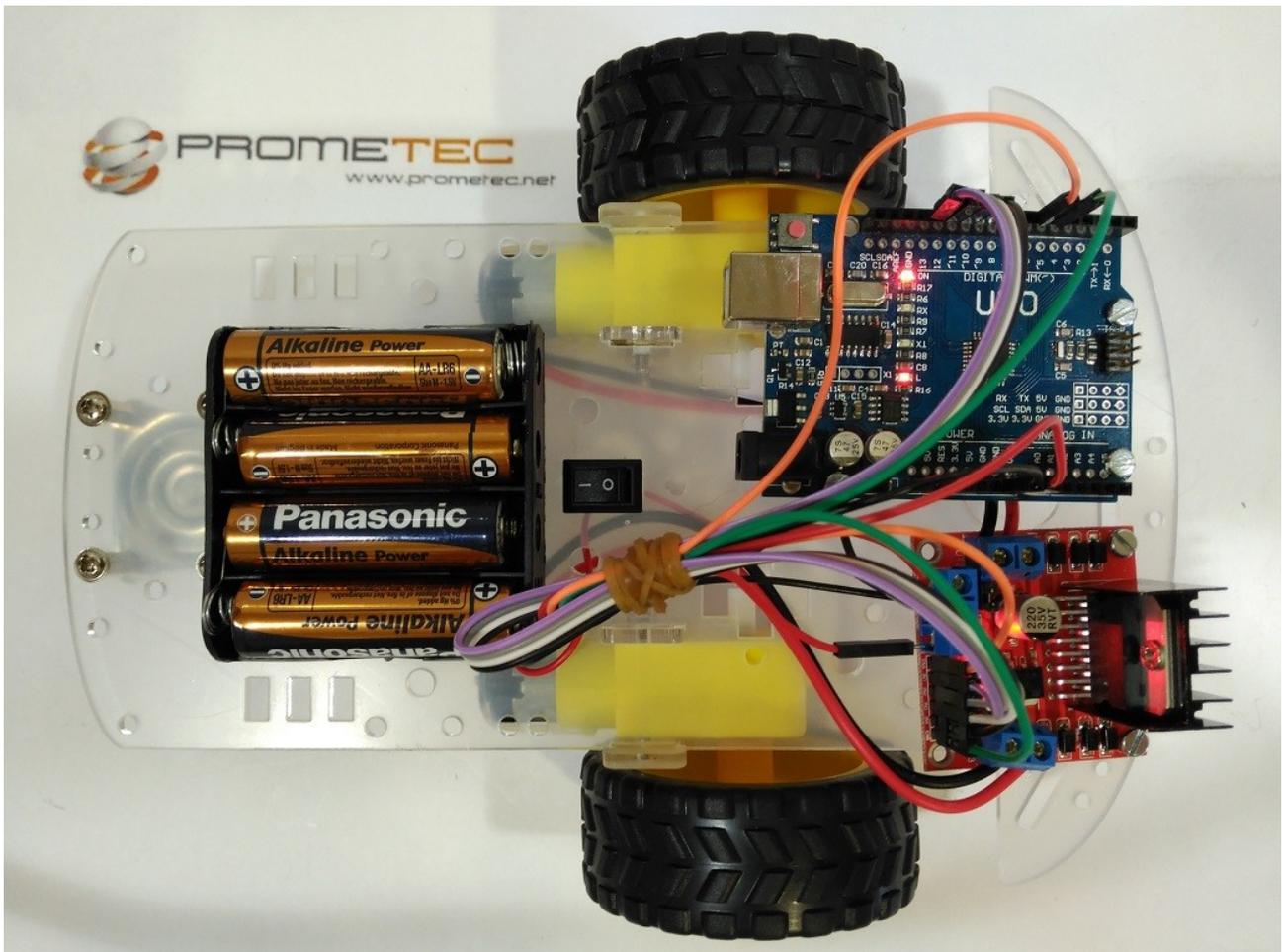
1  #include <Servo.h>
2
3  Servo myservo; // crea el objeto servo
4
5  int pos = 0;   // posicion del servo
6
7  void setup() {
8      myservo.attach(9); // vincula el servo al pin digital 9
9  }
10
11 void loop() {
12     //varia la posicion de 0 a 180, con esperas de 15ms
13     for (pos = 0; pos <= 180; pos += 1)
14     {
15         myservo.write(pos);
16         delay(15);
17     }
18
19     //varia la posicion de 0 a 180, con esperas de 15ms
20     for (pos = 180; pos >= 0; pos -= 1)
21     {
22         myservo.write(pos);
23         delay(15);
24     }
25 }

```

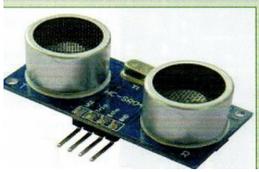
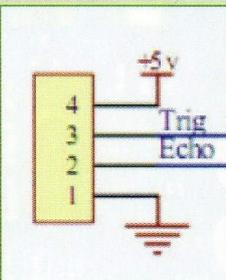
Una vez conectados todos los cables podemos colocarle una goma o una brida para tenerlos unidos que esté un poquito más ordenado. Además podemos meterle ya las pilas, que os recomendamos que sean unas buenas pilas alcalinas.



Si pulsamos el interruptor deberían encenderse los LEDs de alimentación tanto del Arduino como del L298N.



Montaje del sensor de ultrasonidos y programación del vehículo

Medidor de distancias	Sensor de ultrasonidos (HC-SR04)			<p>Utiliza ondas de alta frecuencia y mide el tiempo que tarda la onda en regresar para calcular la distancia de un objeto. Consta de dos partes: una es el emisor(que emite la señal) y la otra es el receptor (que recibe la señal si ésta rebota sobre algún obstáculo cercano).</p>
<p>Este sensor dispone de 4 patillas de entrada: VCC(se conecta a 5V), GND(a GND), TRIG(emite la onda) y ECHO(la recibe); estas dos van a dos entradas digitales a nuestra elección (0,13).</p>				

Utilizamos el sensor **HC-SR04**, que envía a través de uno de los cilindros (Trigger) un sonido inaudible y espera que dicho sonido rebote sobre un objeto y sea captado por otro cilindro (Echo).

Conecta el sensor de ultrasonidos (HC-SR04)de la siguiente manera: VCC(se conecta a 5V), GND (a GND), TRIG al pin 2 y ECHO al pin 3.

Conecta la placa Arduino al ordenador y copia el código:

```

long distancia;
long tiempo;

void setup(){
  Serial.begin(9600);
  pinMode(2, OUTPUT);
  pinMode(3, INPUT);
  pinMode(10, OUTPUT);
  pinMode(11, OUTPUT);
}

void loop(){
  digitalWrite(10,HIGH);
  digitalWrite(11,LOW);
  digitalWrite(2,LOW);
  delayMicroseconds(5);
  digitalWrite(2, HIGH);
  delayMicroseconds(10);
  tiempo=pulseIn(3, HIGH);
  distancia= int(0.017*tiempo);

  if (distancia <=10){
    digitalWrite(10,LOW);
    digitalWrite(11,LOW);
  }
  delay(1000);
}

```

1. En primer lugar definimos la variable distancia y tiempo, que usamos para hacer los cálculos.
2. En el **setup** configuramos el pin 2 (Trig) como salida para emitir la señal, el pin 3 (Echo) como entrada para recibirla y los pines 10 y 11 como salida para controlar el motor.
3. Dentro del **loop**, las dos primeras instrucciones son para hacer que el motor empiece a girar, transmitiendo el movimiento al eje de las ruedas y haciendo que avance.
4. Las dos instrucciones siguientes son para estabilizar el sensor. Ponemos LOW a la salida del Trig.
5. En la siguiente emitimos la señal durante 10 microsegundos.
6. Mediante la función **pulseIn(3, HIGH)** medimos la longitud del pulso entrante. Con pulseIn medimos el tiempo que transcurre entre el envío del pulso ultrasónico y el momento en el que el sensor recibe el rebote, es decir: desde que el pin 3 empieza a recibir el rebote, HIGH, hasta que deja de hacerlo, LOW, medimos el tiempo entre pulsos
7. Calculamos la distancia: velocidad de la luz 340m/s (0,034cm/ μ s) como es ida y vuelta (0,017cm/ μ s) . $E = v \cdot t$. **$E = 0,017 \cdot t$**