

Estás aquí: [Inicio](#) » [Proyectos Arduino](#)

# Crea un robot Arduino con nuestro kit

39 comentarios / [Proyectos Arduino](#) / Por Isaac PE / 15 minutos de lectura

## Índice de Contenidos



Con material barato y fácil de manejar, vamos a montar una **base robótica controlada por Arduino** y que podremos programar para manejar a nuestro antojo los dos motores que guían el robot. Además, con el tiempo podrás añadirle nuevos gadgets a tu robot Arduino

para ir avanzando en la materia de la robótica o la mecatrónica.

 Spanish

La **mecatrónica** es un tipo de ingeniería que engloba otros tipos de ingenierías, como la mecánica, la electrónica, la de control y la informática. En definitiva, nos sirve para crear máquinas móviles automáticas o inteligentes. Al fin y al cabo es lo que hacemos cuando trabajamos con programación, electrónica y elementos mecánicos (actuadores, motores, engranajes, ruedas, ...).

En este artículo te explicaremos **paso a paso** como se monta, conecta y programa todo el conjunto para la base robótica. Así, si nunca has tenido contacto con robots, aprenderás su modo de funcionamiento de la forma más sencilla.

## Materiales necesarios

- Placa Arduino disponible en nuestra tienda.
- Kit robot de inicio disponible en nuestra tienda.
- Controlador de motores HG7881CP de LC Technology Inc. que viene incluido en el kit.  
([Descargar Datasheet](#))
- Cable unifilar para realizar las conexiones entre Arduino y el robot.
- Y como herramientas para el montaje he empleado un simple soldador de estaño (e hilo de estaño para soldar) y el destornillador que viene con el kit.

## Introducción al material del kit

En el **kit robot**, junto con el controlador de motores, componen el material base para la construcción. En el kit vienen diversos componentes que te voy a presentar antes de nada para que los tengas presentes.





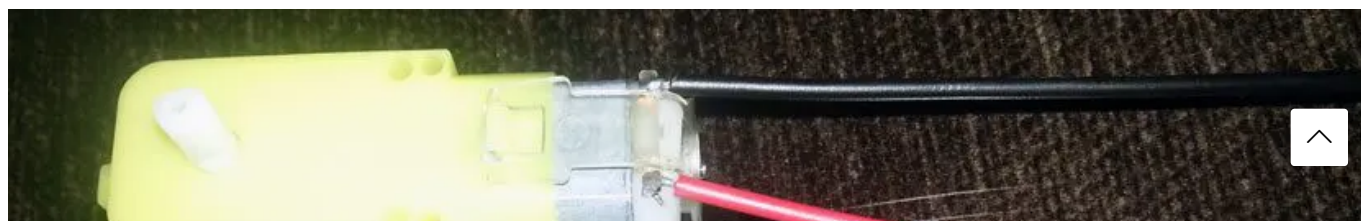
1. Una de las cosas que vienen es un práctico **destornillador** que te servirá para el montaje y desmontaje de toda la tornillería integrada o para trabajos posteriores. El destornillador es de doble punta, para poder contar con un cabezal plano o de estrella en cualquier momento.
2. Verás también cuatro hilos o **cables unifilares**, dos rojos y dos negros. Estos los emplearemos para soldarlos a los conjuntos de motores de CC que también vienen.
3. La **plataforma**, de metacrilato transparente, viene recubierta de un protector de papel marrón que puedes quitar en cualquier momento. Ésta nos servirá de soporte estructural para adherir allí todos los demás componentes, es decir, es el chasis de nuestro robot. Ya viene perforada para un fácil montaje, con agujeros de más para otros posibles montajes y extensiones futuras.
4. Los dos **conjuntos de motor**, como los he llamado, son en realidad dos pequeños motores de CC que transmiten su potencia a unos engranajes de plástico internos (dentro de la carcasa amarilla) para transformar el movimiento giratorio del motor en un movimiento transversal del eje blanco que vemos sobresalir por ambos lados. Además, los engranajes internos actúan como reductores de velocidad para evitar transmitir a las ruedas demasiadas RPMs. Los motores pueden funcionar con un rango amplio de voltajes, pero siempre de corriente continua.
5. Las **dos ruedas** serán las encargadas de mover la base gracias a la potencia transmitida por los motores. Su montaje es muy sencillo, ya que encajan perfectamente en los ejes de los motores.
6. Otro de los componentes que integran el kit es el **rack para las 4 pilas tipo AA** que alimentarán a los motores. El rack coloca las baterías en serie, lo que permite sumar el voltaje de cada una de las pilas para conseguir mayor potencia en los motores. ^

7. Por último encontrarás una pequeña **bolsa de plástico** con tornillos y tuercas para el montaje, así como dos ruedecitas para colocar en los ejes internos de los motores y que te servirán para poner un cuentavueeltas y controlar mejor al robot, y dos soportes metálicos que sujetarán los motores a la plataforma. La rueda loca también servirá como rueda directriz y como punto de apoyo para la zona delantera de la plataforma.
8. El **controlador de motores** doble nos permitirá controlar los motores en conjunto con Arduino. Este controlador de motores tiene dos circuitos, uno para cada motor (o una sola fase con 4 líneas para motores de 2 tipos de etapas). Además cuenta con dos chips controladores HG7881. La pequeña plaquita soporta intensidades de 800 mA y puede funcionar con voltajes de entre 2.5 y 12v (siempre de continua).

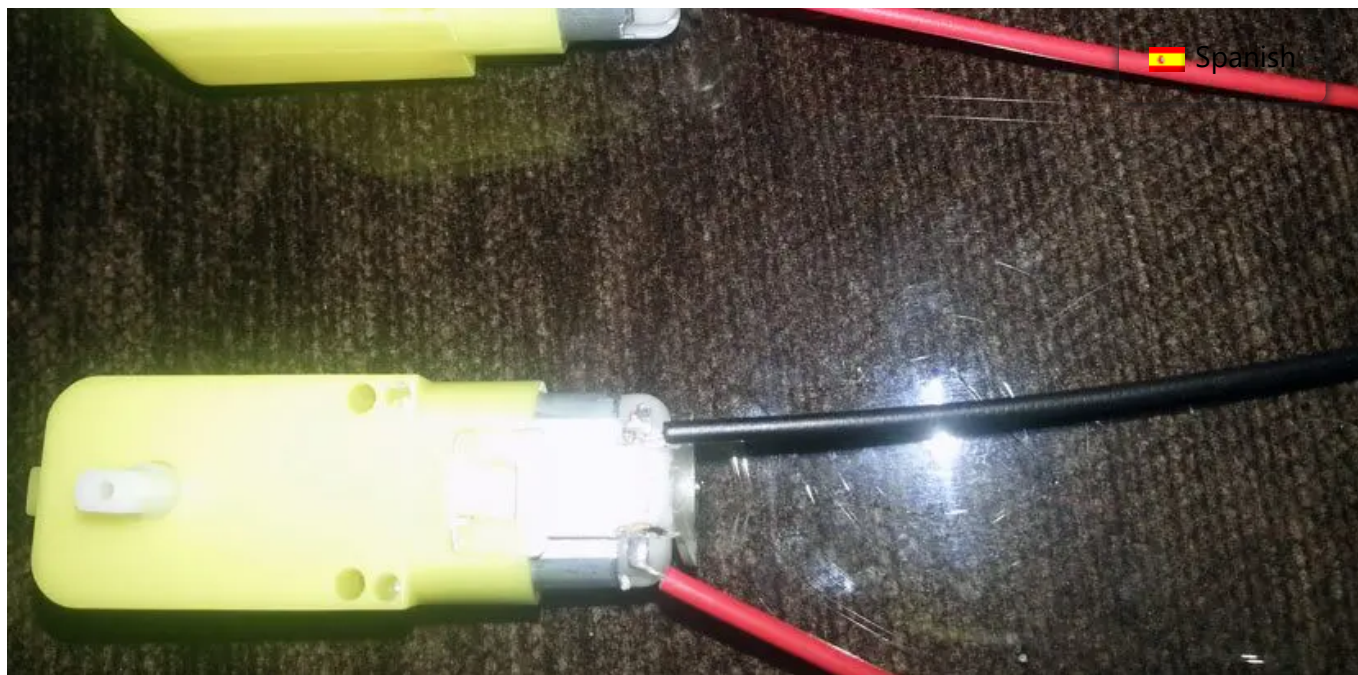
## Montaje del kit robot Arduino

Una vez sepas de qué dispones, es el momento de **empezar a montarlo**:

- Yo he querido empezar por lo más "complicado" por decirlo de alguna manera, puesto que no es nada difícil. Simplemente que lleva algo de más tiempo. Me refiero a **soldar los cables** unifilares a los bornes de cada motor. Uno rojo y uno negro a cada motor. Para eso, he calentado el soldador de estaño y con un poco de estaño he realizado las soldaduras. Lo he querido hacer antes de nada, porque así con los motores desmontados se accede mejor a ellos que no luego una vez estén montados.
- Lo siguiente que he hecho es **acoplar las ruedecitas** que vienen en la misma bolsa que las tuercas, tornillos y rueda loca. Simplemente con un poco de presión se encaja el eje de plástico blanco en la ranura central del cuentavueeltas.
- Luego con los tornillos más largos que vienen en el kit sujetamos los dos **soportes metálicos** que se incluyen con el conjunto motor. Como verás, el envoltorio amarillo del motor tiene unos agujeros pasantes realizados en la misma cara donde hemos colocado las ruedas. Pasa los tornillos por ellos y con una rosca apriétalo.





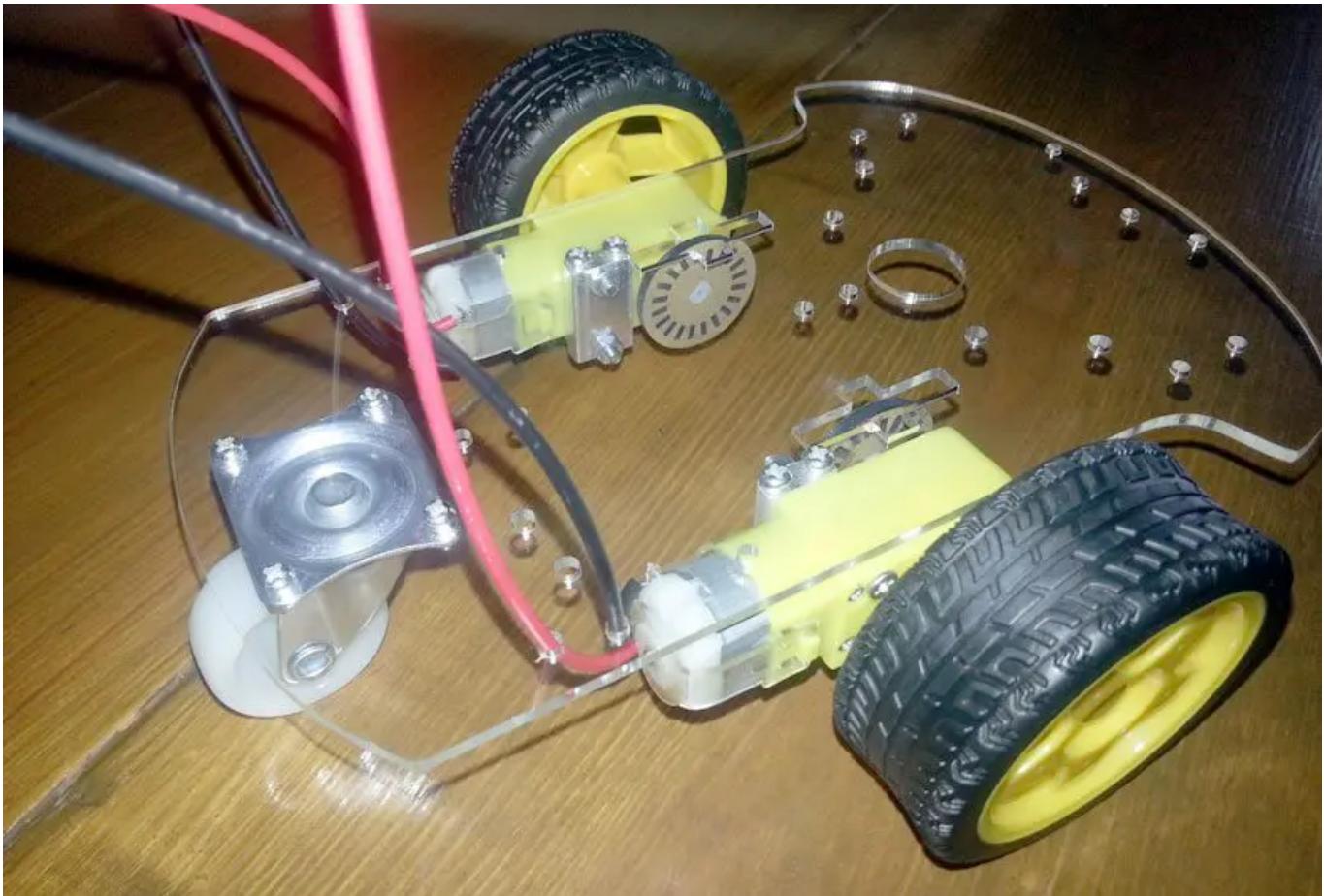


***Nota:*** Un pequeño consejo para la soldadura, verás que vienen con un plástico transparente que los sujeta a la parte amarilla, pues retíralo un poco hacia el lado contrario a donde estés haciendo la soldadura para que no se funda con el calor. Y si es la primera vez que sueldas, verifica que la soldadura haya quedado bien (sin pegotes, roturas,...) y de un color brillante (si el metal presenta un aspecto mate significa que has realizado una soldadura en frío, a más baja temperatura de lo óptimo y es de peor calidad).

- Ahora en el otro extremo del eje blanco de plástico que sobresale del conjunto motor, **colocamos las ruedas** que se incluyen en el kit. Encajan igual que las pequeñas, así que es sencillo. Una vez acopladas, vamos a proceder a sujetarlas al soporte de metacrilato, al igual que la rueda loca. El proceso es sencillo con cuatro tornillos cortos (dos para cada soporte metálico) vamos a sujetar los motores a la plataforma haciéndolos pasar por los agujeros paralelos que se encuentran más cerca de las dos ranuras en forma de cruz que tiene la plataforma. La rueda loca va atornillada con ^

otros cuatro tornillos y cuatro roscas a los cuatro agujeros delanteros (solo encaja de una forma, ya que no es un cuadrado, sino un rectángulo).

 Spanish



***Nota:*** Asegúrate que las ruedas interiores no rocen en las ranuras en forma de cruz que tiene la plataforma, de lo contrario los motores perderán efectividad por el rozamiento. Como ves en la fotografía, yo he empleado dos de los agujeros que hay justo delante del motor para pasar los cables hacia arriba y facilitar luego las conexiones con Arduino y el controlador de motores.

- Finalmente **he sujetado el rack** de las pilas a la plataforma. Pon el rack en un extremo de la plataforma, yo he empleado los dos agujeros más próximos a la rueda loca. El motivo es dejar más espacio en el centro para colocar la placa de Arduino y el

controlador de motores. Solo es un consejo, tú puedes montarlo a tu manera. Por cierto, el montaje es sencillo, dos tornillos y dos tuercas.

 Spanish

**Nota:** He puesto la esquina de los cables hacia la zona interior. Parece una chorrada, pero así luego no tendremos que pasar los cables de un lado a otro para hacer las conexiones. También verás que el rack trae de fábrica el cable negro sacado por la ranura superior y el rojo sale por

*la zona de abajo. Yo he pasado el cable rojo con cuidado de no romperlo hacia la ranura por la que sale el cable negro. ¿Por qué? Sencillo, al apretar el rack con los tornillos y tuercas se ejerce presión y puede terminar cortando el cable rojo. Para evitar esos problemas es mejor pasarlo por arriba.*

 Spanish

Para terminar esta parte, lo que ha hecho es poner cuatro pilas AA (las normales de toda la vida: LR06 de 1,5v) en el rack y con los cables que salen del rack he tocado los dos cables que antes soldamos al motor para **comprobar que funciona** y no roza nada.

**Nota:** Negro con negro y rojo con rojo, en sistemas de CC tienes que tener precaución con los polos porque puedes estropear algunos dispositivos si no respetas los signos, en este caso es un motor y si lo haces al revés no pasa nada, solamente inviertes el sentido del giro del eje al invertir el voltaje, ya aprenderás más sobre esto cuando tratemos el controlador de motores.

Por cierto, guarda los tornillos y tuercas sobrantes, te pueden hacer falta si alguno se pierde en un futuro o si quieres seguir ampliando la plataforma con nuevos sensores.






Ya hemos explicado como se monta el kit, ahora nos falta hablar sobre el controlador de motores, conexionado y programar Arduino para que funcione.

Decir también, que las **posibilidades del kit robot Arduino** son infinitas, tantas como tú imagines. Añadiéndole sensores de proximidad puedes crear un robot autónomo que circule por cualquier lugar sin chocar con nada, capaz de cambiar de dirección solo. Puedes crear sobre la base un robot aspirador con algunos motores y accesorios que puedes construir tú mismo, puedes hacer un robot teledirigido mediante shield bluetooth o RF de Arduino y un largo etc.

Pero antes, vamos a dejar la base lista para que funcione...

## Conexión con el controlador de motores CC

La placa **Arduino** no puede gestionar o controlar directamente motores de corriente continua. Aunque hiciésemos un sketch para darle voltaje al motor o quitárselo gracias a las salidas de Arduino, en reacción a ciertos parámetros, los 20mA que proporciona no son  suficientes para estos motores. Pero eso no es lo que queremos para nuestro robot y por eso

necesitamos el controlador de motores.

 Spanish

Los **motores de cc** que vienen en el kit pueden funcionar perfectamente con 6v y unos 300mA, como hay dos motores, la intensidad demandada será de el doble, 600mA. Nuestro controlador HG7881 de LC Technology puede funcionar entre 2.5 y 12v, con intensidades de 800mA. Así que es perfecto para lo que queremos, además es bueno que soporte hasta 12v, ya que recuerda que el rack posiciona las pilas en serie y suministra una tensión de 6v total (1.5v x 4 pilas) cuando las pilas están al inicio de su vida (carga máxima). Es muy importante este dato, ya que con voltajes inferiores a 6v, el motor no puede funcionar a plena carga.

Con el **driver HG7881** nos basta para nuestra plataforma, además es compacto para ahorrar peso y espacio en nuestra plataforma robotizada. Además es recomendable que lo dejes en un espacio abierto y que no esté al lado de otros elementos que produzcan calor, algunos elementos de los drivers de motores suelen calentarse por la potencia que manejan.

El conexionado de los **cables de los motores** al driver o controlador de motor es sencillo. Si te fijas en el board del controlador aparecen dos módulos verdes de plástico con dos tornillos cada uno. Son fichas de empalme para hacer las conexiones de los cables de los motores a ellos. La conexión es sencilla, con el fantástico destornillador del kit sueltas un poco los tornillos y metes los cables de los motores tal cual se muestra en la siguiente imagen. Luego aprietas los tornillos y listo.



**Nota:** Yo he aprovechado uno de los agujeros de la plataforma y los tornillos y tuercas que sobraron para sujetar el driver a la plataforma, ya que este también cuenta con perforaciones. He empleado solo un tornillo/tuerca y uno de los agujeros centrales que hay entre los orificios en forma de cruz.

En la imagen anterior también puedes ver un esquema de los pines de **conexión del driver**. Como ves, los dos centrales son los de alimentación, Vcc para la tensión y GND para tierra. Aquí deben ir conectados los cables del rack de pilas. Para no dejar el driver inutilizado por si lo queremos emplear para otros experimentos, podemos utilizar jumpers (opción por la que yo he optado) para atrapar el hilo en el pin o hacer una conexión simplemente enrollando el cable. Si quieres, puedes soldarlo como hicimos con los cables del motor, pero para sustituirlo o utilizarlo en otra aplicación te será más difícil...

Con una **crimpadora** (en cristiano, pela cables), si dispones de ella o si no con unas simples tijeras, puedes quitar el envoltorio aislante de los cables del rack de pilas para dejar un poco más de hilo conductor desnudo y poder enrollarlo en los pines centrales. Además, es me<sup>^</sup>...

que queden así para poderlos quitar en cualquier momento e interrumpir el suministro, a no ser que dispongas de un interruptor para desconectar el suministro, de lo contrario no podrás pararlo a no ser que quites las pilas.

***Nota:*** Si quieres soldarlos porque veas que es mejor y no vas a emplear el dirver para otros fines, puedes hacerlo y si no dispones de interruptor, una cosa que se me ocurre fácil y rápida para desconectar el aparato es ^



*mediante una lámina de plástico fina, colocándola entre el electrodo negativo de la pila y el contacto metálico del rack para interrumpir el flujo de electricidad.*

 Spanish

## Conexionado con Arduino y programación

Ya solo nos falta escribir el **código fuente** de nuestro sketch y hacer las conexiones entre el controlador de motor y Arduino. Lo que haremos a continuación es conectar con más hilo unifilar los pines de Arduino y los pines de ambos lados del controlador que han quedado libres.

Yo he elegido esta configuración, si lo prefieres puedes crear tu otra, pero no olvides rectificar el código fuente para que coincidan con los **pines** que has empleado. De las salidas digitales de Arduino he escogido la 9 y 5 para A-IA y A-IB respectivamente, mientras que para las B-IA y B-IB he seleccionado las 10 y 6. Estos pines del controlador de motores son los que darán la velocidad y dirección al motor. El pin A-IA y A-IB controlarán el motor A (fíjate que junto a las fichas de empalme verdes donde has conectado los cables que vienen de los motores, en la placa hay una instrucción que pone Motor A), mientras que B-IA y B-IB controla el motor B. Conecta los cables de este modo y se puede decir que físicamente está terminado.

Si Arduino manda una señal baja al pin A-IA y una alta al A-IB, el motor gira en un sentido a como lo haría si A-IA está en nivel alto y A-IB en bajo. ¿Ves que sencillo es controlar la dirección de los motores? Esto es así si los conectamos a unos pines digitales normales, pero si los conectamos a pines digitales **PWM** de Arduino (recuerda que son aquellas que tienen el símbolo ~), además podremos controlar la velocidad de los motores. Por eso elegí los pines PWM en este ejemplo.

Sin el **controlador de motores** solo podríamos dar potencia a uno u otro motor o quitársela y si quisiésemos cambiar el sentido deberíamos invertir la colocación de los cables. Para modificar la velocidad habría que jugar con distintas pilas o voltajes... Como ves poco práctico y es por ello que el controlador de motores nos abre un mundo nuevo de

^

 Spanish

 Spanish

```
delay(1000);  
atras();  
delay(1000);  
izquierda();  
delay(1000);  
derecha();  
delay(1000);  
}
```

//Configuracion de cada dirección para que el controlador lo haga

```
void atras()  
{  
  analogWrite(A1A, 0);  
  analogWrite(A1B, velocidad);  
  analogWrite(B1A, 0);  
  analogWrite(B1B, velocidad);  
}
```

//Si te fijas, para cambiar la dirección de giro es justo lo contrario al anterior

```
void adelante()  
{  
  analogWrite(A1A, velocidad);  
  analogWrite(A1B, 0);  
  analogWrite(B1A, velocidad);  
  analogWrite(B1B, 0);  
}
```

```
void izquierda()  
{  
  analogWrite(A1A, velocidad);  
  analogWrite(A1B, 0);
```

^

```
analogWrite(BIA, 0);  
analogWrite(BIB, velocidad);  
}  
  
void derecha()  
{  
  analogWrite(AIA, 0);  
  analogWrite(AIB, velocidad);  
  analogWrite(BIA, velocidad);  
  analogWrite(BIB, 0);  
}  
  
[/cpp]
```


## ¿Qué velocidad máxima es capaz de alcanzar nuestro robot Arduino?

*Aportación al artículo de **Jose Antonio Navarro**.*

Nuestra plataforma está alimentada con cuatro pilas o baterías de 1,5 Voltios, eso nos da una tensión de 6 voltios para alimentar a nuestros motores. Bien, pues si alimentamos directamente de las pilas al motor, podemos calcular la velocidad máxima que podrá alcanzar nuestro robot. Veamos como:

1. Medimos las revoluciones por minuto que produce nuestro motor, para ello, una vez montada la plataforma, le pegamos un pequeño trozo de cinta en el borde de la rueda, elevamos la plataforma para que no se desplace por la mesa y conectamos el motor a las pilas (recordad que estén al máximo de carga para que los cálculos sean precisos). Usando un cronómetro cualquiera contamos las vueltas que da la rueda en un espacio de tiempo determinado, digamos 30 segundos o 1 minuto (evidentemente esto es posible porque tenemos un motor con una reductora grande y las revoluciones que



produce son relativamente pocas). En nuestro caso yo he acercado el dedo  Spanish ligeramente a la rueda, sin tocarla, y he ido contando los golpecitos que me daba la cinta en el dedo, mientras miraba el cronómetro. He repetido esta operación varias veces para asegurarme que no cometía errores y luego he realizado la media de las tres medidas. Eso me ha dado un valor de 115 revoluciones en 30 seg.

2. Medimos el diámetro de la rueda que es de unos 66 mm con lo que el perímetro o longitud de la circunferencia de la rueda es:

$$l = 2 * \pi * r$$

$$l = 2 * 3,14 * 33$$

$$l = 207,24 \text{ mm ó } 20,7 \text{ cm}$$

3. Si ya vimos que nuestra rueda gira a 115 revoluciones cada 30 segundos, significa que gira a:

$$115 / 30 = 3,83 \text{ rev/seg.}$$

4. Si cada revolución avanza 20,7 cm, cada segundo nuestro robot recorrerá:

$$20,7 * 3,83 = 79,28 \text{ cm/seg.}$$

5. O lo que es lo mismo:



$$79,28 * 60 / 100 = 47,57 \text{ m/min. (metros por minuto)}$$

 Spanish

6. O lo que es lo mismo:

$$47,57 * 60 / 1000 = \underline{\underline{2,85 \text{ km/h}}}$$

**Nota:** Estos cálculos son en vacío y como os comentaba antes, con la carga de las pilas al máximo. Cuando esté todo montado y apoyado en superficie la velocidad será algo menor ya que tendrá que mover todo ese peso.

Espero que les haya gustado y le encuentren muchas aplicaciones. Cualquier duda estamos por los comentarios.

39 comentarios en "Crea un robot Arduino con nuestro kit"

[Comentarios anteriores](#)

**LUIS FUENTES**

10 DIC, 2021 A LAS 14:30

Como encargo el kit??

[Respon](#) 

**MATIAS LOPEZ**

27 AGO, 2021 A LAS 00:15

 Spanish

buenisimo... tendras los archivos stl para imprimir las partes de plastico...

[Responder](#)

**ALFREDO**

7 ENE, 2020 A LAS 11:07

El controlador de motores no viene incluido en el kit o por lo menos a mi no me llegó.

[Responder](#)

**ALBERTO NAVARRO**

8 ENE, 2020 A LAS 12:07

¡Buenas Alfredo!

Hemos hecho varios cambios en el kit, puede que cuando tu lo compraste no lo mandáramos conjunto (con un precio menor).

Si te fijas ahora lo tenemos en un formato mucho más completo.

¡Saludos!

[Responder](#)



**YAMILETH HERNANDEZ**

25 AGO, 2017 A LAS 22:52

 Spanish

que tipo de arduino es

[Responder](#)

**ALBERTO NAVARRO**

26 AGO, 2017 A LAS 00:57

Arduino UNO.

[Responder](#)

**ALVARO**

14 JUL, 2017 A LAS 02:42

Disculpe su controlador de motores es igual, mejor o inferior al  
Controlador de motores HG7881CP

[Responder](#)

**ISAAC PE**

20 NOV, 2017 A LAS 18:39

Hola Álvaro,

EL controlador de motores incluido en el kit de robótica es similar al HG7881CP.

Un saludo y gracias por seguirnos.

[Respon](#)



[Comentarios anteriores](#)

 Spanish

## Deja un comentario

Tu dirección de correo electrónico no será publicada. Los campos obligatorios están marcados con \*

Escribe aquí...

Nombre\*

Correo electrónico\*

**Publicar comentario »**

