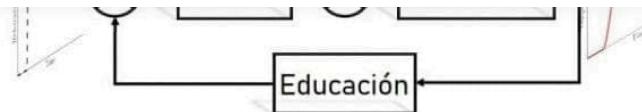


Este sitio web usa Cookies de terceros y propios, si continuas navegando las aceptas.

Aceptar

Rechazar

[Leer Más](#)



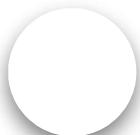
[ARDUINO](#)

Motor DC con Encoder – Velocidad – Posición

[Inicio](#) » [Arduino](#) » **Motor DC con Encoder – Velocidad – Posición**



En esta entrada vamos a aprender como funciona el Motor DC con Encoder el cual nos va a permitir saber el sentido de giro del motor, conocer la posición en la que se desplaza el motor y finalmente la velocidad con la que gira el motor. Esto lo podemos acoplar a cualquier sistema embebido, como por ejemplo nuestra placa de desarrollo del Arduino.



Este sitio web usa Cookies de terceros y propios, si continuas navegando las aceptas. [Aceptar](#) [Rechazar](#)

[Leer Más](#)

- 1.2. Metal DC Geared Motor w/Encoder – 6V 210RPM 10Kg.cm
- 1.3. Kit de Iniciación de Arduino (Alta Calidad)
- 2. Como funciona un encoder de cuadratura de Efecto Hall
 - 2.1. Bluetooth HC-05 / HC06
 - 2.2. Motor Paso a Paso Arduino
 - 2.3. PWM Arduino
- 3. Medir la velocidad y posición motor DC
 - 3.1. Medición de posición
 - 3.2. Medición de velocidad
 - 3.3. Código de Arduino

Antes de comenzar, te invito para que le des un vistazo a nuestro [CURSO GRATUITO DE ARDUINO DESDE CERO](#).

Y también para que te suscríbas al canal, si te gusta la programación de dispositivos embebidos, la programación y la teoría del control.



Listado de Materiales

- Placa de Desarrollo de Arduino
- Motor DC con Encoder de 5v o 12v
- Driver de Potencia Puente H (L298)
- Potenciómetro 5k (o cualquiera)
- 2 Pulsadores



Este sitio web usa Cookies de terceros y propios, si continuas navegando las aceptas.

Aceptar

Rechazar

[Leer Más](#)



Gravity: 2x2A Motor Shield for Arduino Twin

[Comprar en DFRobot](#)



Metal DC Geared Motor w/Encoder – 6V 210RPM 10Kg.cm

[Comprar DFRobot](#)



Kit de Iniciación de Arduino (Alta Calidad)

[Comprar en DFRobot](#)

Como funciona un encoder de cuadratura



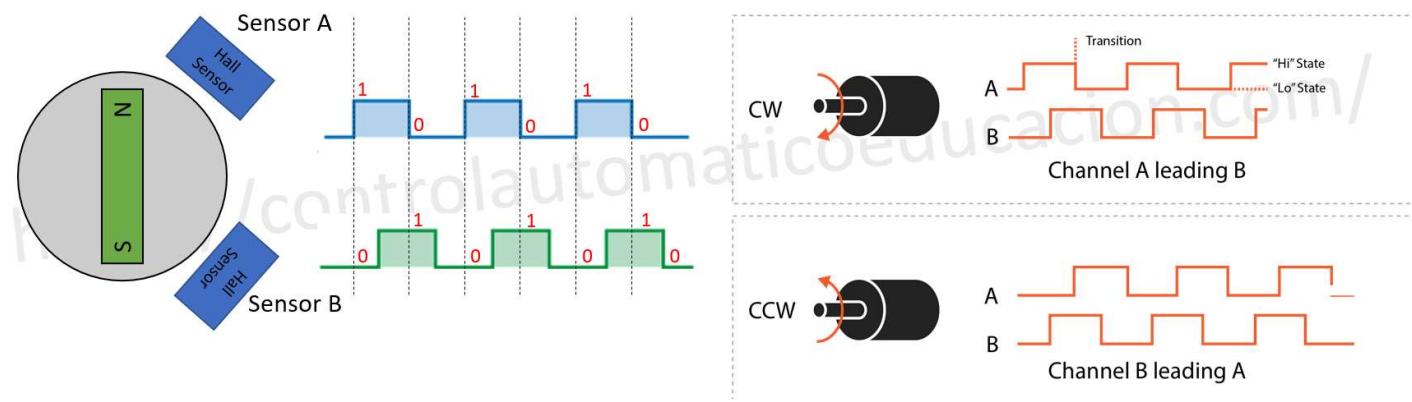
Este sitio web usa Cookies de terceros y propios, si continuas navegando las aceptas. [Aceptar](#) [Rechazar](#)

[Leer Más](#)



Un Encoder (codificador) funciona a través de la detección de los cambios en el campo magnético creado por un imán conectado al eje del motor. A medida que el motor gira, las salidas del Encoder se dispararán periódicamente.

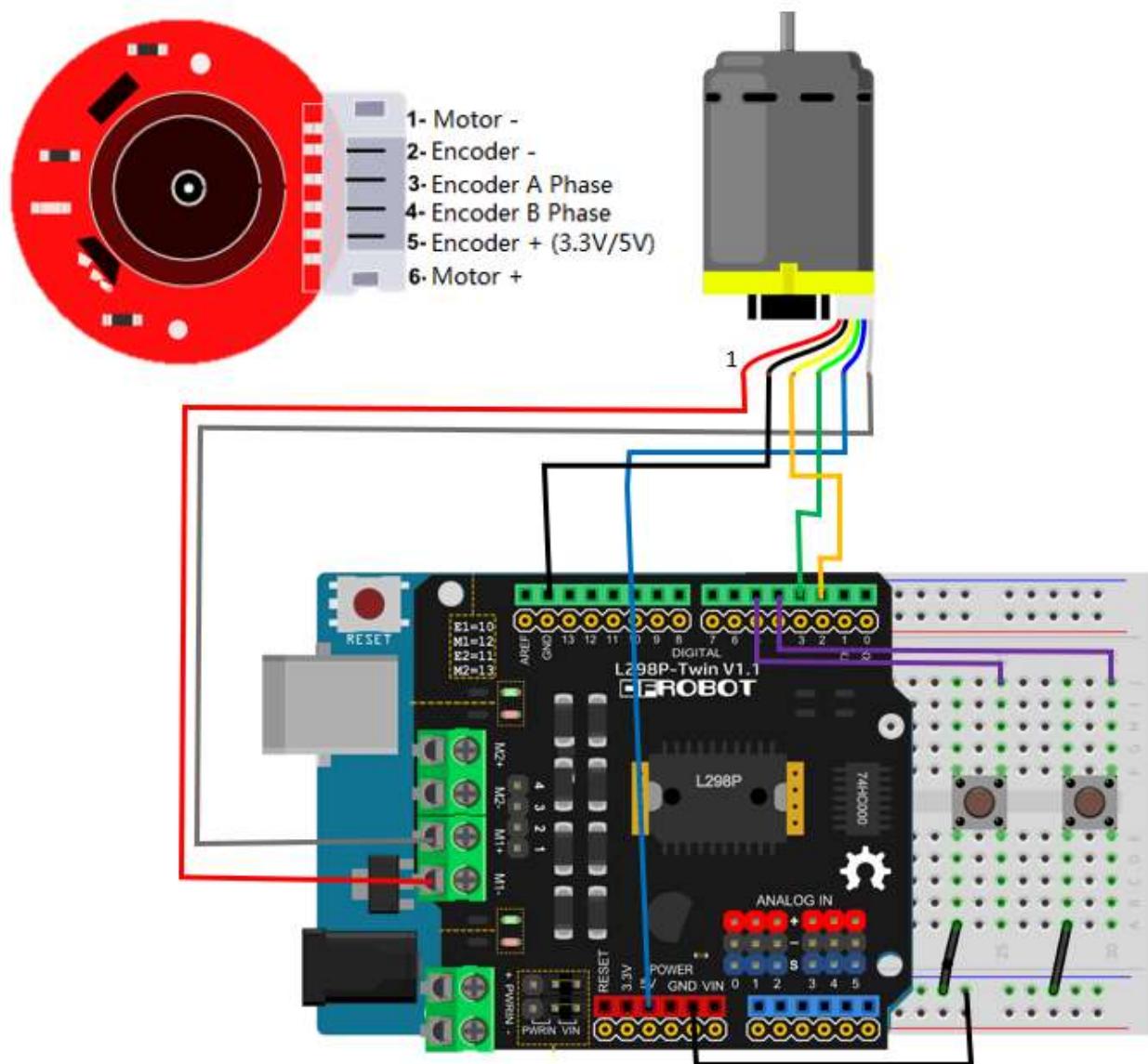
Generalmente contamos con 2 sensores que transforman el giro del motor (trasductor) a 2 señales cuadradas que presentan un desfase de 90°. Gracias a este desfase, a estos codificadores se les denomina como **encoders en cuadratura**, ocupando un cuadrante del círculo de 360°



Este sitio web usa Cookies de terceros y propios, si continuas navegando las aceptas. [Aceptar](#) [Rechazar](#)

[Leer Más](#)

Este efecto lo podemos ver fácilmente en nuestro Arduino, para eso implementamos el siguiente circuito.



En el circuito anterior se esta empleando un módulo Shield L298, se puede usar cualquier otro módulo o montar directamente el circuito.

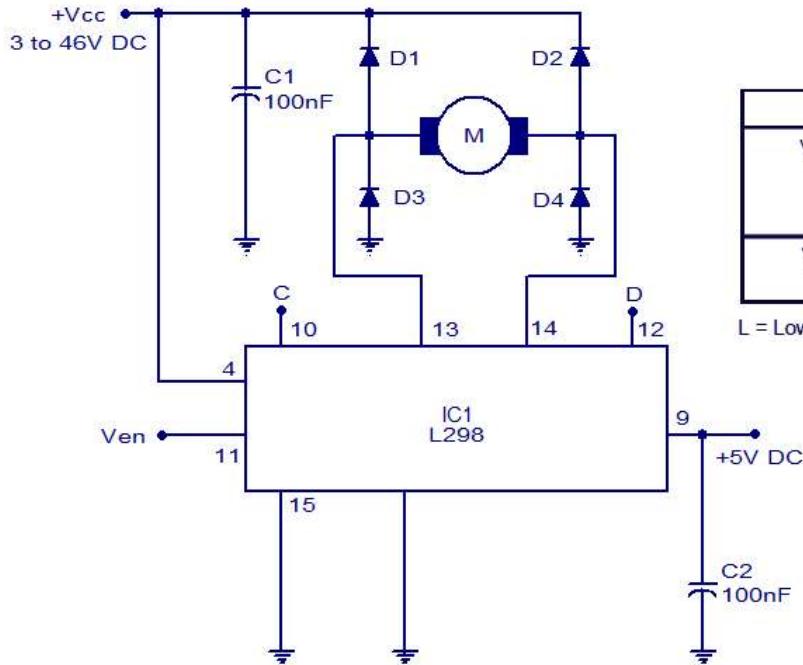
El L298 es un controlador de puente completo dual que tiene un amplio rango de voltaje operativo y puede manejar corrientes de carga de hasta 3A. El IC también cuenta con



Este sitio web usa Cookies de terceros y propios, si continuas navegando las aceptas. [Aceptar](#) [Rechazar](#)

[Leer Más](#)

describe en la tabla que se muestra debajo del esquema del circuito.



Inputs		Function
V _{en} = H	C = H ; D = L	Forward
	C = L ; D = H	Reverse
	C = D	Fast Motor Stop
V _{en} = L	C = X ; D = X	Free Running Motor Stop

L = Low

H = High

X = Don't care

Este sitio web usa Cookies de terceros y propios, si continuas navegando las aceptas. [Aceptar](#) [Rechazar](#)

[Leer Más](#)



Encoder Metal Gearmotor 12V DC High Speed 130RPM...

✓prime



Bemonoc Encoder Metal Gearmotor 12V DC High Speed...

BESTSELLER NO. 3



Azssmuk Quadrature Encoder Metal Gearmotor Geared...

El código en Arduino es:



Este sitio web usa Cookies de terceros y propios, si continuas navegando las aceptas. [Aceptar](#) [Rechazar](#)

[Leer Más](#)

```
5. //***** by: Sergio Andres Castaño Giraldo      *****/
6. //***** https://controlautomaticoeducacion.com/ *****/
7. //*****
8. //***** ****
9. //***** ****
10.
11. #define ENCODER_A      2 // Amarillo
12. #define ENCODER_B      3 // Verde
13. #define BUTTON_FORWARD  4
14. #define BUTTON_BACKWARD 5
15.
16. // Pines de Control Shield
17. const int E1Pin = 10;
18. const int M1Pin = 12;
19. const int E2Pin = 11;
20. const int M2Pin = 13;
21.
22. typedef struct{
23.     byte enPin;
24.     byte directionPin;
25. }Motor;
26.
27. //Creo el motor
28. const Motor motor = {E1Pin, M1Pin};
29.
30. const int Forward = LOW;
31. const int Backward = HIGH;
32.
33. void setup(){
34.     Serial.begin(9600);
35.     //Encoders como entradas
36.     pinMode(ENCODER_A, INPUT);
37.     pinMode(ENCODER_B, INPUT);
38.     //Pulsadores
39.     pinMode(BUTTON_FORWARD, INPUT_PULLUP);
40.     pinMode(BUTTON_BACKWARD, INPUT_PULLUP);
41.     //Configura Motor
42.     pinMode(motor.enPin, OUTPUT);
```



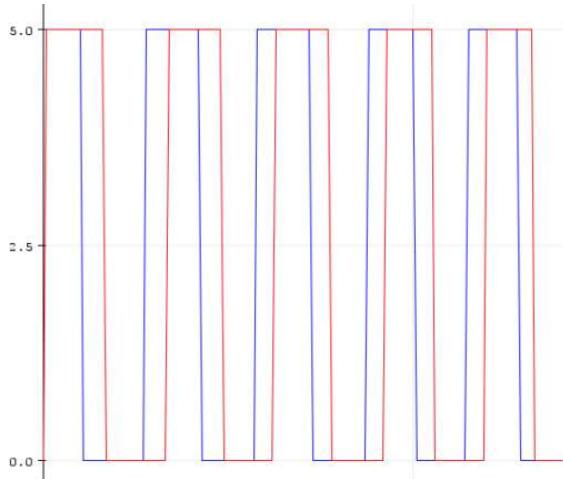
Este sitio web usa Cookies de terceros y propios, si continuas navegando las aceptas. [Aceptar](#) [Rechazar](#)

[Leer Más](#)

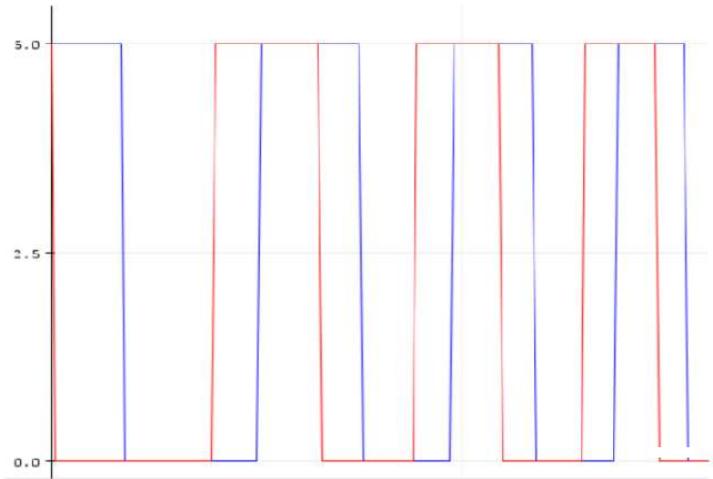
```

48.     if(! digitalRead(BUTTON_FORWARD)) {
49.         digitalWrite(motor.directionPin, Forward);
50.         digitalWrite(motor.enPin, HIGH);
51.         imprimir_cuadratura();
52.     }
53.     else if(! digitalRead(BUTTON_BACKWARD)) {
54.         digitalWrite(motor.directionPin, Backward);
55.         digitalWrite(motor.enPin, HIGH);
56.         imprimir_cuadratura();
57.     }
58.     else{
59.         digitalWrite(motor.enPin, LOW);
60.     }
61. }
62.
63. void imprimir_cuadratura(){
64.     int a = digitalRead(ENCODER_A);
65.     int b = digitalRead(ENCODER_B);
66.     Serial.print(a*5);
67.     Serial.print(" ");
68.     Serial.println(b*5);
69. }
```

La señal de cuadratura observada por el serial plotter:



BACKWARD



FORWARD



Este sitio web usa Cookies de terceros y propios, si continuas navegando las aceptas. [Aceptar](#) [Rechazar](#)

[Leer Más](#)

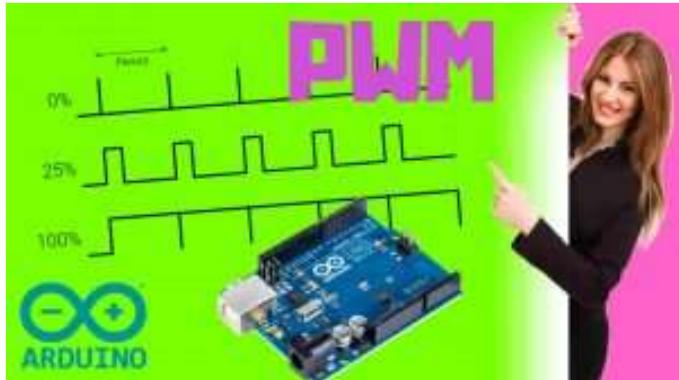


Bluetooth HC-05 / HC06

Aprende como usar el módulo HC05/HC06 para realizar una comunicación Bluetooth con Arduino de forma inalámbrica. Códigos y videos tutoriales.

Motor Paso a Paso Arduino

Aprende a usar un MOTOR PASO A PASO ARDUINO bien sea UNIPOLAR o BIPOLAR de una forma sencilla de entender. Observa el ejemplo y el video explicativo.



PWM Arduino

Aprende a usar el PWM con Arduino y usa esta señal de control en tus proyectos. Entiende como regular procesos y modificar la frecuencia del PWM.

Medir la velocidad y posición motor DC

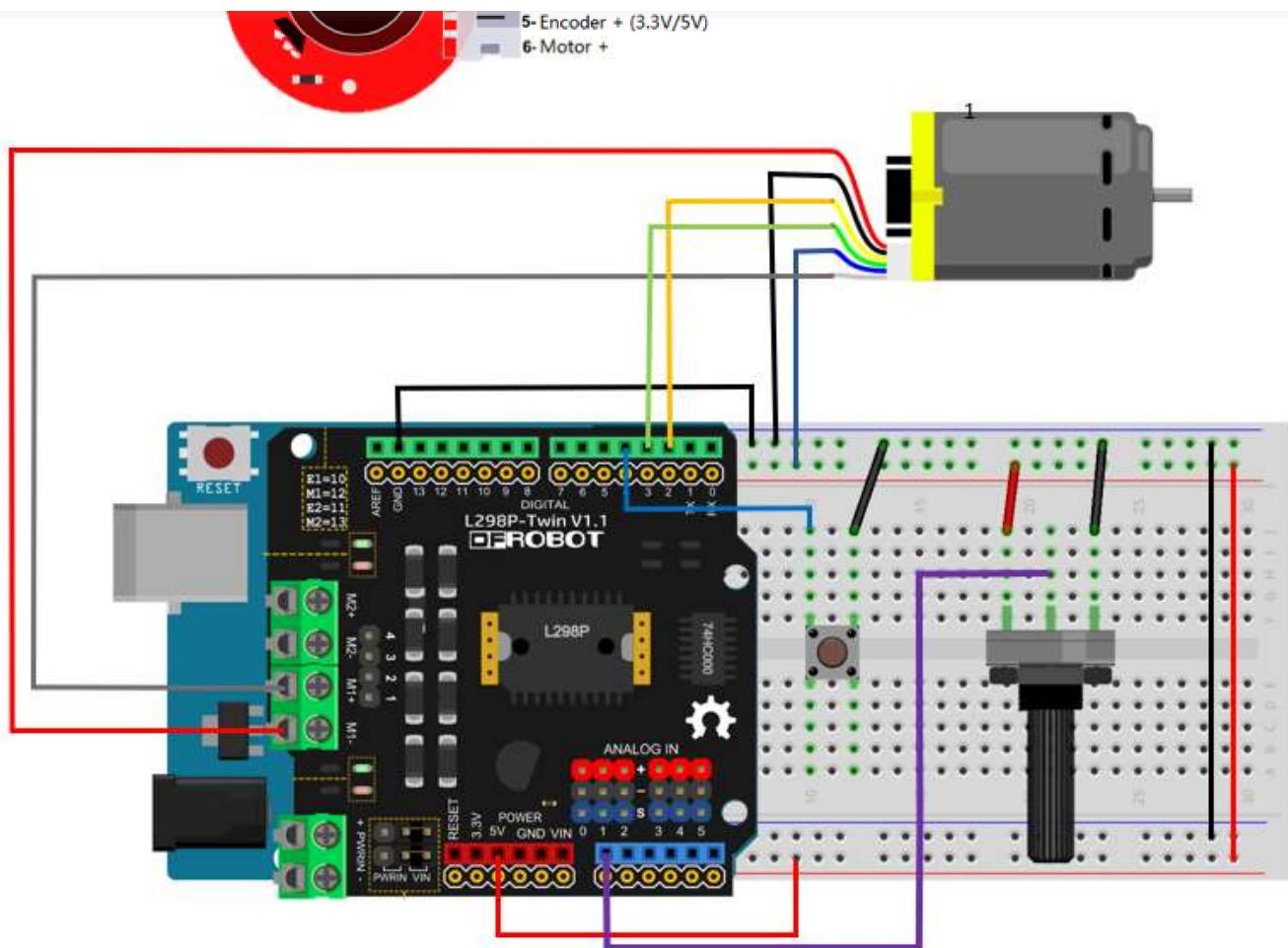
El hecho de disponer el encoder acoplado al rotor del motor de corriente continua, nos brinda la posibilidad de poder medir su posición y velocidad para ser empleado en muchas aplicaciones de robótica.

Para esta práctica vamos a modificar ligeramente el circuito presentado anteriormente



Este sitio web usa Cookies de terceros y propios, si continuas navegando las aceptas. [Aceptar](#) [Rechazar](#)

[Leer Más](#)



De esta manera con el pulsador podemos seleccionar el modo de operación: Modo de regulación de velocidad, o modo de posición. Ambos controlados por la [lectura ADC con Arduino](#) del Potenciómetro.

Medición de posición

Para la lectura de la posición, vamos a tener que recurrir a las [interrupciones con Arduino](#), donde escogeremos el Sensor 1 (A) para que active la interrupción con el flanco de subida. En la interrupción se incrementa una variable que llamamos **theta** (si la salida del encoder B es alta) o se resta uno (si la salida del encoder B es baja).

Para garantizar que la variable *theta* se almacene de modo que pueda ser leída con precisión por las funciones de loop y de interrupción, debe utilizar el calificador [*volátil*](#).



Este sitio web usa Cookies de terceros y propios, si continuas navegando las aceptas.

Aceptar

Rechazar

[Leer Más](#)

Finalmente se establece una velocidad fija en el motor DC, con un PWM de 200, y se pregunta por la lectura ADC del potenciómetro para poder ejercer un control aproximado de la posición del motor. Este control se logra con dos condicionales donde mantenemos una tolerancia de más o menos 2 grados, con el fin que el rotor pueda detenerse en ese pequeño umbral o threshold.

Medición de velocidad

La medición de velocidad es muy similar a la de posición, solo que en este caso, se debe configurar el arduino para que quede capturando los datos de cualquiera de los dos sensores del encoder (A o B) por un periodo de tiempo que sea conocido, para realizar posteriormente el cálculo de la velocidad.

Como ya se está leyendo la interrupción con el sensor 1 (A) lo que hacemos es incrementar una variable entera volatil llamada **pulsos**.

Dentro del void loop, haremos uso de la función millis() del Arduino, para contabilizar de una forma precisa 1 segundo.

Una vez ha transcurrido 1 segundo, entramos en el condicional y dentro de la macro ATOMIC calculamos las RPM del motor con base a 1 segundo, empleando la resolución del encoder (Número de pulsos por giro) en este caso es de 374.22. Actualizamos la variable **timeold** para que millis() vuelva y contabilice nuevamente 1 segundo y ceramos el contador **pulsos** para realizar una nueva medición.

Código de Arduino

El siguiente será el código que vamos a implementar para esta práctica y el cual nos será muy útil para otros proyectos queharemos posteriormente en el sitio web, por lo tanto es importante que lo entiendan a la perfección.



Este sitio web usa Cookies de terceros y propios, si continuas navegando las aceptas. [Aceptar](#) [Rechazar](#)

[Leer Más](#)

```
7. //*****
8. //*****
9. //*****
10.
11. // this library includes the ATOMIC_BLOCK macro.
12. #include <util/atomic.h>
13.
14. #define ENCODER_A      2 // Amarillo
15. #define ENCODER_B      3 // Verde
16. #define BUTTON_MOD     4
17.
18.
19. // Pin del Potenciómetro
20. const int pot = A0;
21.
22. // Pines de Control Shield
23. const int E1Pin = 10;
24. const int M1Pin = 12;
25. const int E2Pin = 11;
26. const int M2Pin = 13;
27.
28. //Variable global de posición compartida con la interrupción
29. volatile int theta = 0;
30.
31. //Variable global de pulsos compartida con la interrupción
32. volatile int pulsos = 0;
33. unsigned long timeold;
34. float resolution = 374.22;
35.
36. //Variable Global Velocidad
37. int vel = 0;
38.
39. //Variable Global Posicion
40. int ang = 0;
41.
42. //Variable Global MODO
43. bool modo = false;
44.
```



Este sitio web usa Cookies de terceros y propios, si continuas navegando las aceptas. [Aceptar](#) [Rechazar](#)

[Leer Más](#)

```
50.  
51. //Creo el motor  
52. const Motor motor = {E1Pin, M1Pin};  
53.  
54. //Constantes de dirección del Motor  
55. const int Forward = LOW;  
56. const int Backward = HIGH;  
57.  
58. void setup() {  
59.     // set timer 1 divisor to 1024 for PWM frequency of 30.64 Hz  
60.     TCCR1B = TCCR1B & B11111000 | B00000101;  
61.     Serial.begin(9600);  
62.     //Encoders como entradas  
63.     pinMode(ENCODER_A, INPUT);  
64.     pinMode(ENCODER_B, INPUT);  
65.     //Pulsadores  
66.     pinMode(BUTTON_MOD, INPUT_PULLUP);  
67.     //Configura Motor  
68.     pinMode(motor.enPin, OUTPUT);  
69.     pinMode(motor.directionPin, OUTPUT);  
70.     //Configurar Interrupción  
71.     timeold = 0;  
72.     attachInterrupt(digitalPinToInterrupt(ENCODER_A), leerEncoder, RISING);  
73. }  
74.  
75. void loop() {  
76.     float posicion;  
77.     float rpm;  
78.     int value, dir=true;  
79.  
80.     //Lee el Valore del Potenciómetro  
81.     value = analogRead(pot);  
82.  
83.     //Cambia de Modo Velociadad o Posición  
84.     if(debounce(BUTTON_MOD)) {  
85.         modo = !modo;  
86.         theta = 0;  
87.     }
```



Este sitio web usa Cookies de terceros y propios, si continuas navegando las aceptas. [Aceptar](#) [Rechazar](#)

[Leer Más](#)

```
93.     //Activa el motor dirección Forward con la velocidad
94.     setMotor(motor, vel, false);
95.
96.     //Espera un segundo para el calculo de las RPM
97.     if (millis() - timeold >= 1000)
98.     {
99.         //Modifica las variables de la interrupción forma atómica
100.        ATOMIC_BLOCK(ATOMIC_RESTORESTATE) {
101.            //rpm = float(pulsos * 60.0 / 374.22); //RPM
102.            rpm = float((60.0 * 1000.0 / resolution) / (millis() - timeold));
103.            timeold = millis();
104.            pulsos = 0;
105.        }
106.        Serial.print("RPM: ");
107.        Serial.println(rpm);
108.        Serial.print("PWM: ");
109.        Serial.println(vel);
110.    }
111. }
112. else{
113.     //Transforma el valor del Pot a ángulo
114.     ang = map(value,0,1023,0,360);
115.
116.     //Modifica las variables de la interrupción forma atómica
117.     ATOMIC_BLOCK(ATOMIC_RESTORESTATE) {
118.         posicion = (float(theta * 360.0 /resolution));
119.     }
120.
121.     //Posiciona el ángulo con tolerancia +- 2
122.     if(ang > posicion+2){
123.         vel = 200;
124.         dir = true;
125.     }
126.     else if(ang < posicion-2){
127.         vel = 200;
128.         dir = false;
129.     }
130. }
```



Este sitio web usa Cookies de terceros y propios, si continuas navegando las aceptas. [Aceptar](#) [Rechazar](#)

[Leer Más](#)

```
136.  
137. //Función para dirección y velocidad del Motor  
138. void setMotor(const Motor motor, int vel, bool dir){  
139.     analogWrite(motor.enPin, vel);  
140.     if(dir)  
141.         digitalWrite(motor.directionPin, Forward);  
142.     else  
143.         digitalWrite(motor.directionPin, Backward);  
144. }  
145.  
146. //Función anti-rebote  
147. bool debounce(byte input) {  
148.     bool state = false;  
149.     if(! digitalRead(input)){  
150.         delay(200);  
151.         while(! digitalRead(input));  
152.         delay(200);  
153.         state = true;  
154.     }  
155.     return state;  
156. }  
157.  
158. //Función para la lectura del encoder  
159. void leerEncoder(){  
160.     //Lectura de Velocidad  
161.     if(modo)  
162.         pulsos++; //Incrementa una revolución  
163.  
164.     //Lectura de Posición  
165.     else{  
166.         int b = digitalRead(ENCODER_B);  
167.         if(b > 0){  
168.             //Incremento variable global  
169.             theta++;  
170.         }  
171.         else{  
172.             //Decremento variable global  
173.             theta--;
```



Este sitio web usa Cookies de terceros y propios, si continuas navegando las aceptas.

Aceptar

Rechazar

[Leer Más](#)

Eso es todo por la entrada del dia de hoy, espero les haya gustado y hayan aprendido algo nuevo. Si te ha servido el contenido de esta entrada, de los videos y los códigos de implementación y deseas apoyar mi trabajo invitandome a un café super barato, puedes hacerlo en el siguiente link:

 [Invitar a Sergio a un Café ☕](#)

Que esten muy bien, nos vemos en la siguiente entrada.

[Volver al Curso de Arduino](#)

Entradas relacionadas



Control PID en Tanque Esférico con Arduino



Sensor de Flujo YF-S201: Medición de Caudal con Arduino



Este sitio web usa Cookies de terceros y propios, si continuas navegando las aceptas. [Aceptar](#) [Rechazar](#)

[Leer Más](#)



Bluetooth HC-05 / HC06



Balanza Electronica con HX711 y Arduino



Acelerómetro y Giroscópio



Comunicación Serial Arduino
Simulink/Matlab

Deja una respuesta

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

NOMBRE *



Este sitio web usa Cookies de terceros y propios, si continuas navegando las aceptas. [Aceptar](#) [Rechazar](#)

[Leer Más](#)

WEB

RECIBIR UN CORREO ELECTRÓNICO CON LOS SIGUIENTES COMENTARIOS A ESTA ENTRADA.

RECIBIR UN CORREO ELECTRÓNICO CON CADA NUEVA ENTRADA.

[Publicar el comentario](#)

Este sitio usa Akismet para reducir el spam. [Aprende cómo se procesan los datos de tus comentarios.](#)

Comentarios (18)



Como hago la conexion con el driver l298n?

Daniel ,

[Responder](#)



Este sitio web usa Cookies de terceros y propios, si continuas navegando las aceptas. [Aceptar](#) [Rechazar](#)

[Leer Más](#)



Si se puede.

Sergio C,

[Responder](#)



que programa utilizaste para simular los componentes?

juan ,

[Responder](#)



Uso el Proteus principalmente

Sergio C,

[Responder](#)



Este sitio web usa Cookies de terceros y propios, si continuas navegando las aceptas.

Aceptar

Rechazar

[Leer Más](#)

Estupendo tutorial. Muy claro e instructivo.

Gabilon ,

He visto que en tu web hay varios tutoriales PID, ¿has publicado este mismo código de para arduino pero incorporando el algoritmo PID para actuar sobre el setMotor de forma que se mantenga la velocidad constante?

Muchas gracias

[Responder](#)



Buenas noches señor Sergio Andres Castaño Giraldo, tengo una pregunta . Ronaldo Sanchez ,
En el código de hallar velocidad del motor DC con encoder, hay una variable llamada resolution (resolución) ¿Qué es resolution? ¿Por qué la variable resolution=374.22? ¿De dónde se obtendrá el valor de la variable resolution? ¿Es una característica del motor que se encuentra en el datasheet? ¿En qué parte del datasheet se encontraría?

```
//rpm = float(pulsos * 60.0 / 374.22); //RPM  
rpm = float((60.0 * 1000.0 / resolution) / (millis() - timeold) * pulsos);
```

[Responder](#)



Este sitio web usa Cookies de terceros y propios, si continuas navegando las aceptas. [Aceptar](#) [Rechazar](#)

[Leer Más](#)

[Responder](#)



Buenos días señor Sergio Andres Castaño Giraldo

Ronaldo Sanchez ,

Tengo una inquietud

¿Donde se puede conseguir la librería ?

En el Arduino no viene incluida la librería necesaria para el control de posición y velocidad del motor DC

¿Usted me podría responder esta pregunta? por favor señor Sergio ,he buscado en muchos lugares y no encuentro la libreria

[Responder](#)



Realmente desconozco si existe una librería. De igual forma, en esta entrada Sergio C, mostramos como manipular el motor bien sea por velocidad o por posición, con esto puedes implementar un controlador PID para regular alguna de estas dos variables. Acá en el sitio web tenemos varias entradas sobre el controlador PID. Saludos.

[Responder](#)



Este sitio web usa Cookies de terceros y propios, si continuas navegando las aceptas. [Aceptar](#) [Rechazar](#)

[Leer Más](#)

La libreria que uso no es ATOMIC BLOCK sino la encuentro Ronaldo Sanchez, en el Arduino una libreria parecida que se puede descargar desde el Arduino llamada SIMPLY ATOMIC

[Responder](#)



Cómo se sustituye la shield por el módulo?

[Pedro Reyes Mateo](#),

[Responder](#)



Venden módulo que no necesariamente son Shields. O también puedes montar tu propio circuito, en el post coloque un circuito mostrando como realizarlo.

Sergio C,

[Responder](#)



Este sitio web usa Cookies de terceros y propios, si continuas navegando las aceptas. [Aceptar](#) [Rechazar](#)

[Leer Más](#)

Cómo se sustituye la shield por el módulo?

Jorge Vera ,

[Responder](#)



Muchas gracias ingeniero Sergio

luis fernando muñoz ,

[Responder](#)



De nada Luis, que bueno que te ha gustado. Gracias por comentar. Saludos! Sergio C ,

[Responder](#)



Este sitio web usa Cookies de terceros y propios, si continuas navegando las aceptas. [Aceptar](#) [Rechazar](#)

[Leer Más](#)



De nada Adeluna100, Saludos!!

Sergio C,

[Responder](#)

Buscar:

SERGIO ANDRÉS CASTAÑO GIRALDO

Mi nombre es Sergio Andres Castaño Giraldo, y en este sitio web voy a compartir una de las cosas que mas me gusta en la vida y es sobre la Ingeniería de Control y Automatización. El sitio web estará en constante crecimiento, voy a ir publicando material sobre el asunto desde temas básicos hasta temas un poco más complejos. Suscríbete al sitio web, dale me gusta a la página en Facebook y únete al canal de youtube. Espero de corazón que la información que comparto en este sitio, te pueda ser de utilidad. Y nuevamente te doy las gracias y la bienvenida a control automático educación.

Cursos con DESCUENTO

Dando Click en las siguientes imágenes, tendrás un **CUPÓN** de **DESCUENTO** que es **Exclusivo** del sitio web, dado que si buscas los cursos en Udemy **NO** vas a obtener ningún descuento.



Este sitio web usa Cookies de terceros y propios, si continuas navegando las aceptas. [Aceptar](#) [Rechazar](#)

[Leer Más](#)



Este sitio web usa Cookies de terceros y propios, si continuas navegando las aceptas.

Aceptar

Rechazar

[Leer Más](#)



Este sitio web usa Cookies de terceros y propios, si continuas navegando las aceptas.

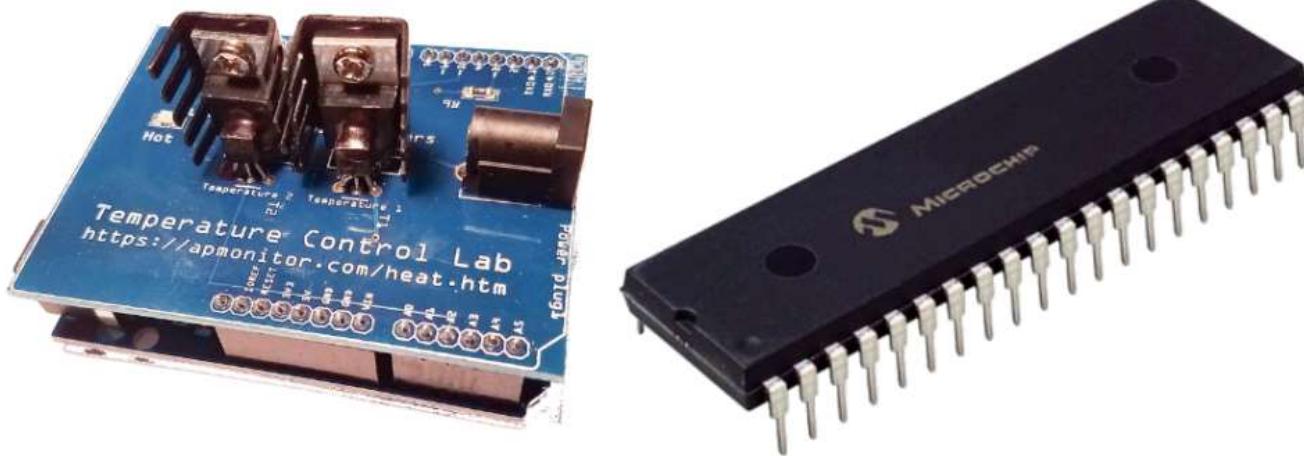
Aceptar

Rechazar

[Leer Más](#)

CUPON VALIDO

SISTEMAS CONTROL



👉 Accede al curso de **FUNDAMENTOS EN INSTRUMENTACIÓN INDUSTRIAL** dando click en la imagen de aquí abajo 👈



Este sitio web usa Cookies de terceros y propios, si continuas navegando las aceptas.

Aceptar

Rechazar

[Leer Más](#)



SÍGUEME EN

CANAL EN TELEGRAM

Únete a @ConAutEdu en Telegram



Este sitio web usa Cookies de terceros y propios, si continuas navegando las aceptas.

Aceptar

Rechazar

[Leer Más](#)



Somos 100.000 Controle...



CANALES DE YOUTUBE

No te pierdas mis videos cuando los subo a la red. Únete al canal de Youtube en Español.



Sergio A. Castaño Giraldo

YouTube 132 K

Você pode-se inscrever no Canal de YouTube em Português:

Sergio A. Castaño Giraldo - Brasil



YouTube 3 K

SIGUE LA PÁGINA DE FACEBOOK

CURSOS

[Análisis de Sistemas](#)

[Arduino](#)

[Control de Procesos](#)

[Control Predictivo](#)

[Control Realimentado](#)

[Herramientas y Tecnologías de Desarrollo](#)

[Instrumentación](#)

[Labview](#)

[MATLAB](#)



Este sitio web usa Cookies de terceros y propios, si continuas navegando las aceptas. [Aceptar](#) [Rechazar](#)

[Leer Más](#)

[Sistemas Dinámicos Lineales](#)

ACCESO

  [Invítame a un Café](#)

 [Acerca de mi](#)

 [Contacto](#)

 [Política de Privacidad](#)

 [Política de Cookies](#)

En este sitio web vas a encontrar cursos sobre microcontroladores, teoría de control, automatización y ejemplos de implementación de código

