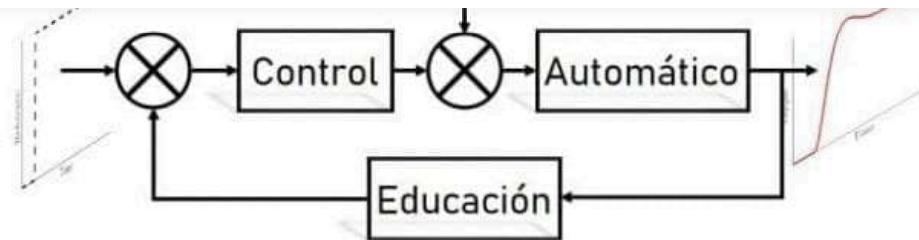


Este sitio web usa Cookies de terceros y propios, si continuas navegando las aceptas. [Aceptar](#) [Rechazar](#) [Leer Más](#)



[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.

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 temas un poco más complejos. Suscríbete al sitio web, dale me gusta a la página en Facebook y únete al canal

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

- 1.1. Gravity: 2x2A Motor Shield for Arduino Twin
- 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. Salidas Digitales en Arduino
 - 2.2. Medidor de Velocidad de un Motor DC con PIC
 - 2.3. Referencia Analógica AREF en 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.



Sergio A. Castaño Giraldo

[YouTube](#) 132 K



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.

Lista de Materiales

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

Aceptar

Rechazar

[Leer Más](#)

MOTOR DC CON ENCODER DE 5V O 12V

- Driver de Potencia Puente H (L298)
- Potenciómetro 5k (o cualquiera)
- 2 Pulsadores
- Cables
- Fuente de Alimentación de 5v o 12v dependiendo del motor empleado.



**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](#)



👉 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](#)

cuadratura de Efecto Hall

Motor DC con Encoder - Arduino [Velocidad y Posición]  DFRobot



SÍGUEME EN

CANAL EN TELEGRAM

Únete a @ConAutEdu en
Telegram



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.

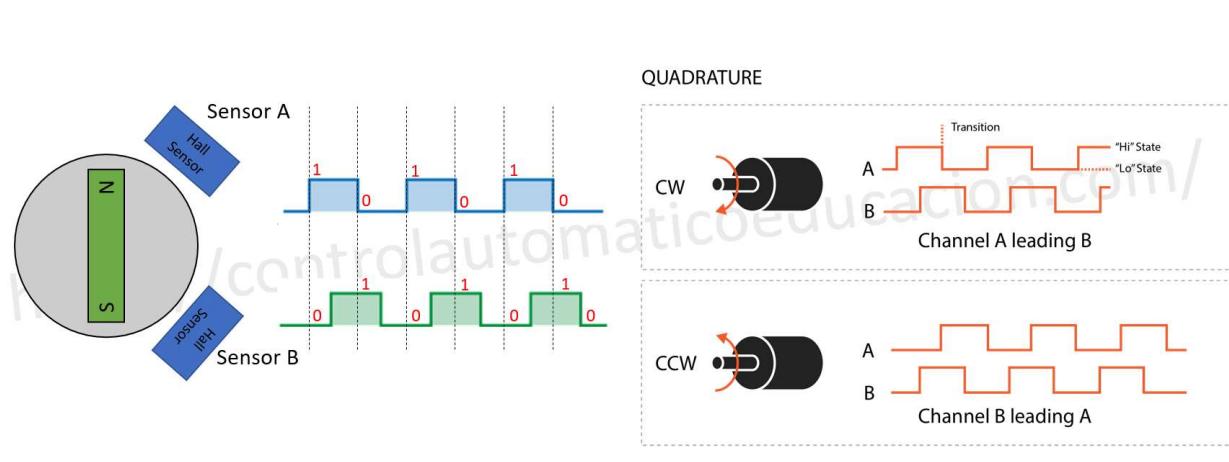
Generalmente contamos con 2 sensores que transforman el giro del motor (trasductor) a 2 señales cuadradas que presentan un desfase de 90°. Gracias a

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

Aceptar

Rechazar

[Leer Más](#)



Cuando el imán gira en el **sentido de las agujas del reloj**, la salida del sensor A se activará primero. Cuando se **gira en sentido antihorario**, por otro lado, la salida del sensor B se activará primero

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

los 100K suscriptores en el Canal!!

Somos 100.000 ...



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:



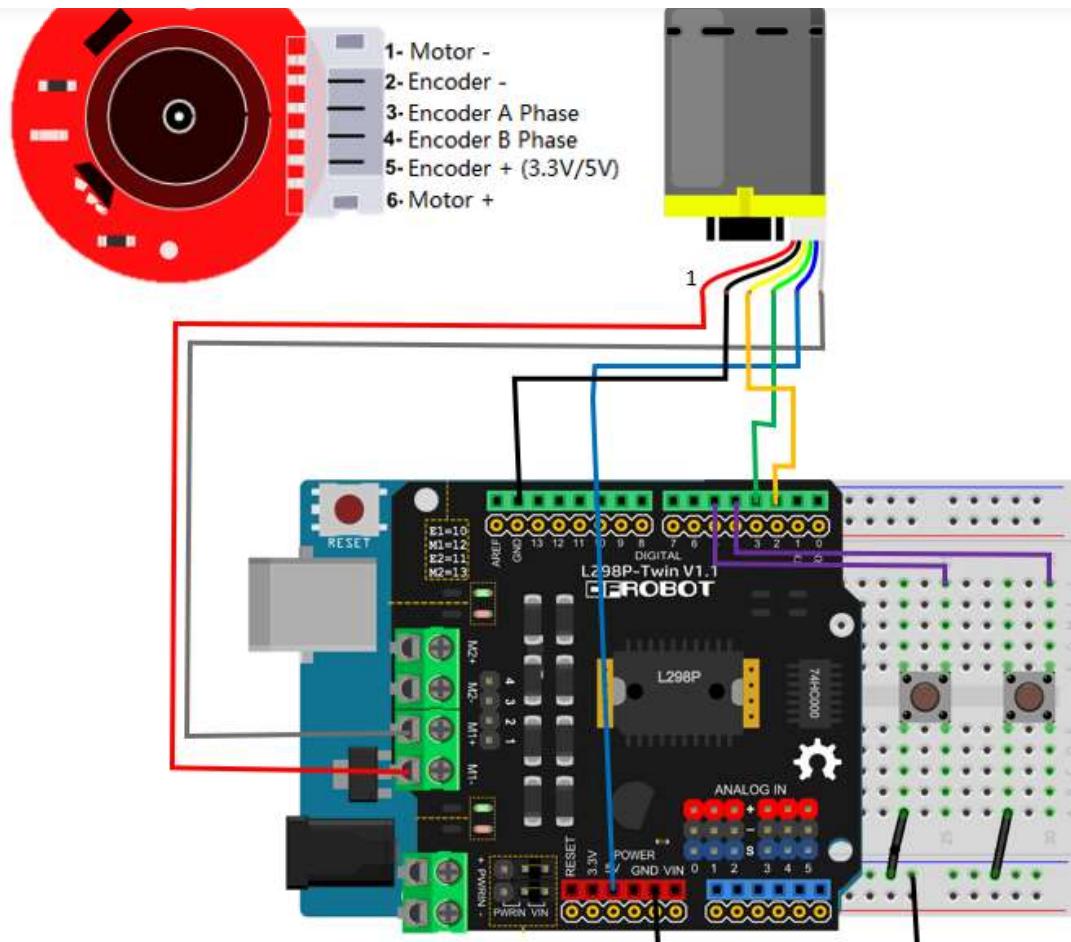
Sergio A. Castaño Giraldo - Brasil

YouTube 3 K

SIGUE LA PÁGINA DE FACEBOOK

CURSOS

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

[Aceptar](#)[Rechazar](#)[Leer Más](#)

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 voltaje de saturación bajo y protección contra sobrtemperatura.

[Control de Procesos](#)[Control Predictivo](#)[Control Realimentado](#)[Herramientas y Tecnologías de Desarrollo](#)[Instrumentación](#)[Labview](#)[MATLAB](#)[Microcontroladores \(PIC\)](#)[MicroPython](#)[Python desde Cero](#)[Sistemas Dinámicos Lineales](#)

ACCESO

[Invítame a un Café](#)[Acerca de mi](#)

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

Aceptar

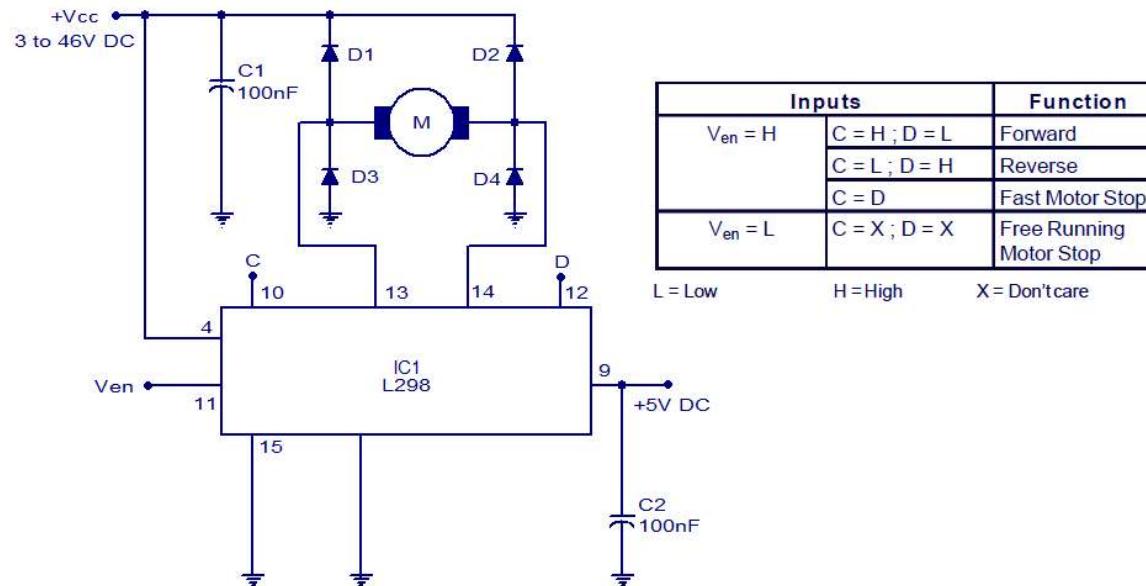
Rechazar

[Leer Más](#)

tensión de alimentación. El estado del motor dependerá del nivel lógico de los pines 10, 11, 12 y se describe en la tabla que se muestra debajo del esquema del circuito.



[Política de Cookies](#)



BESTSELLER NO. 1

BESTSELLER NO. 2

REBAJAS

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...

(104)

prime



DC 12V High Torque Worm Geared Motor Gear...

(11)

prime

\$18.39



BESTSELLER NO. 3



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



El código en Arduino es:

```
1. //*****  
2. //*****  
3. //***** Posición y Velocidad de Motor DC *****//  
4. //*****  
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;
```

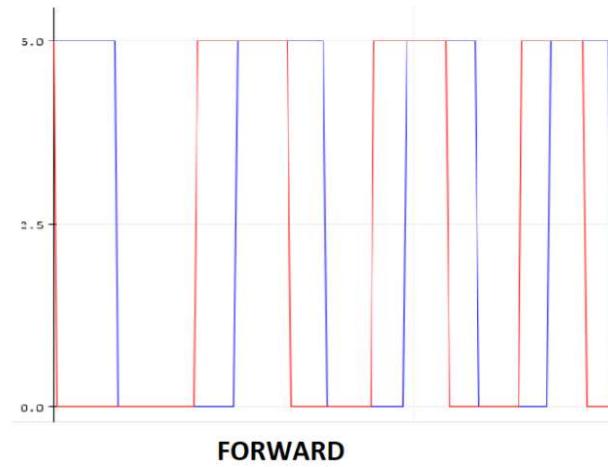
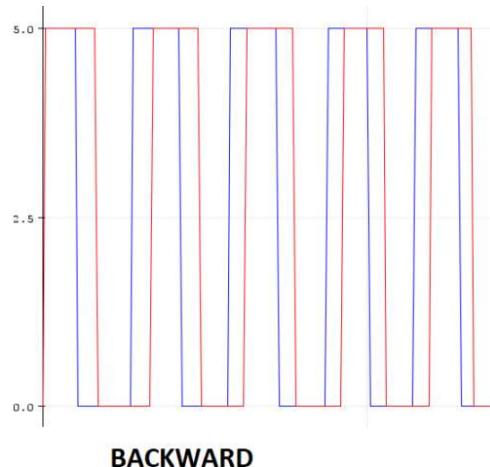
Este sitio web usa Cookies de terceros y propios, si continuas navegando las aceptas. [Aceptar](#) [Rechazar](#) [Leer Más](#)

```
27. //Cree 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);
43.     pinMode(motor.directionPin, OUTPUT);
44. }
45.
46. void loop() {
47.     // Pulsador hacia adelante
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();
```

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

```
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:



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

Aceptar

Rechazar

[Leer Más](#)



Salidas Digitales en Arduino

Salidas Digitales en Arduino Uno, Mega, Leonardo o Cualquier Arduino. Aprende a programarlas y configurarlas de diferentes formas. Mira el video y el código

Medidor de Velocidad de un Motor DC con PIC

Aprende como hacer un Medidor de Velocidad con PIC (Tacometro digital) empleando el PIC C Compiler (CCS C) + Display LCD.

Referencia Analógica AREF en Arduino

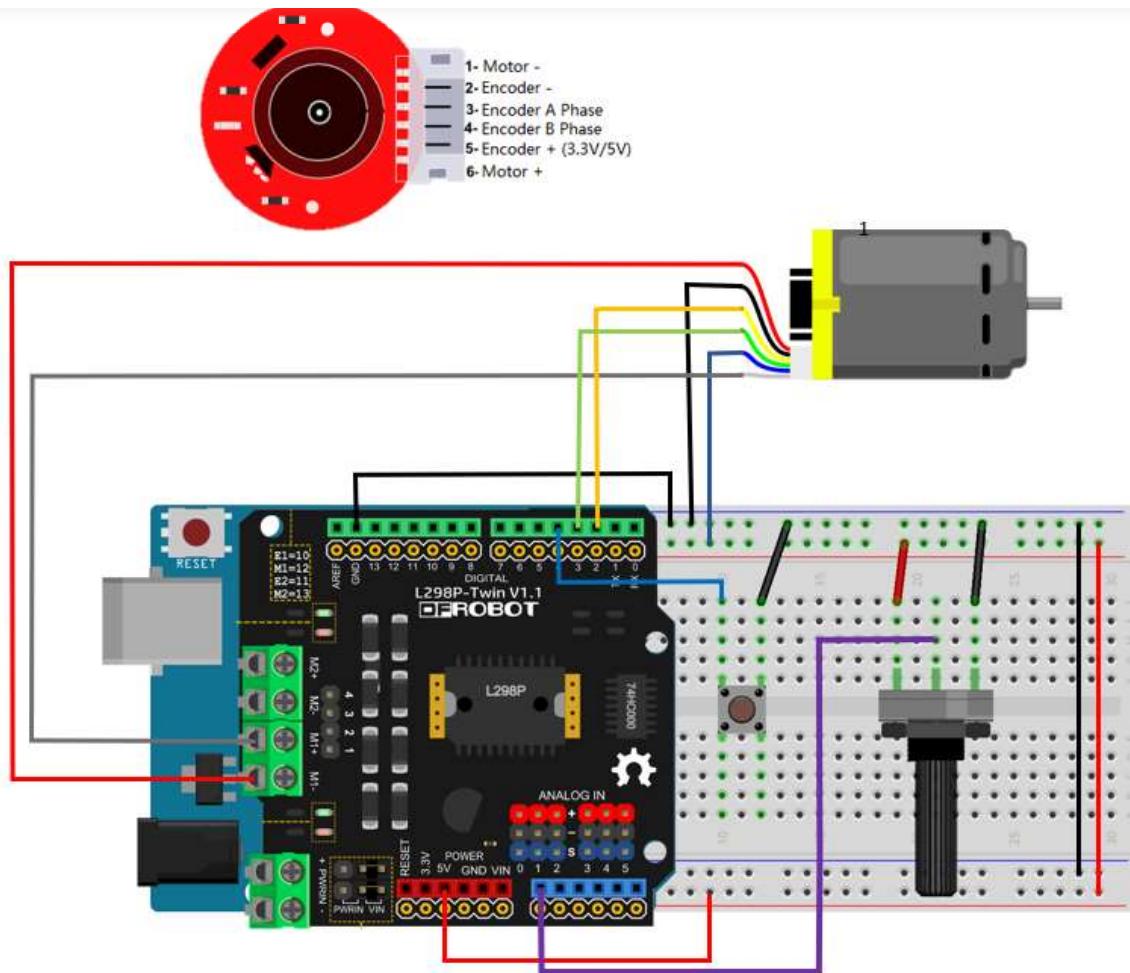
Aprende cual es la función del PIN AREF ARDUINO y como nos puede ayudar a obtener el máximo potencial de la conversión ADC de la PLACA. Ejemplos y Códigos.

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 levemente el circuito presentado anteriormente, donde solo dejaremos un solo pulsador y agregaremos un

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

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

Aceptar

Rechazar

[Leer Más](#)

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](#).

Además, se necesita una macro **ATOMIC_BLOCK** para acceder a la variable de posición. La macro ATOMIC_BLOCK evita que la interrupción cambie parte de la variable *theta* mientras se está leyendo.

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.

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

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 que haremos posteriormente en el sitio web, por lo tanto es importante que lo entiendan a la perfección.

```
1. //*****  
2. //*****  
3. //*****      Posición y Velocidad de Motor DC      *****//  
4. //*****  
5. //***** by: Sergio Andres Castaño Giraldo      *****//  
6. //***** https://controlautomaticoeducacion.com/ *****//  
7. //*****  
8. //*****  
9. //*****  
10. //*****
```

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

```
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;
```

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

```
47.     byte enPin;
48.     byte directionPin;
49. }Motor;
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
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;
```

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

```
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. }
88.
89. if(modo) {
90.     //Transforma el valor del Pot a velocidad
91.     vel = map(value,0,1023,0,255);
92.
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()
103.            timeold = millis();
104.            pulsos = 0;
105.        }
106.        Serial.print("RPM: ");
107.        Serial.println(rpm);
108.        Serial.print("PWM: ");
109.        Serial.println(vel);
```

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

```
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.     else{
131.         vel = 0;
132.     }
133.     setMotor(motor, vel, dir);
134. }
135. }
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
```

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

```
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--;
174.         }
175.     }
```

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



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



Bluetooth HC-05 / HC06



Balanza Electronica con HX711 y Arduino

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

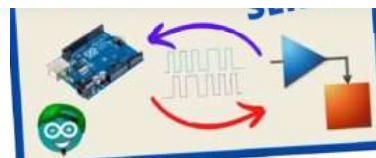
Aceptar

Rechazar

[Leer Más](#)



Acelerómetro y
Giroscópio



Comunicación Serial
Arduino
Simulink/Matlab



Arduino con
Simulink

Deja una respuesta

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

NOMBRE *

CORREO ELECTRÓNICO *

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

Aceptar

Rechazar

[Leer Más](#)

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](#)



Se Puede sustituir el modulo por un L293d

Carlos Oswaldo ,

[Responder](#)



Si se puede.

Sergio C ,

[Responder](#)



que programa utilizaste para simular los componentes?

juan ,

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



Uso el Proteus principalmente

Sergio C ,

[Responder](#)



Estupendo tutorial. Muy claro e instructivo.

Gabilon ,

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

[Responder](#)

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



Buenas noches señor Sergio Andres Castaño Giraldo,tengo una pregunta . Ronaldo Sanchez,

En el codigo de hallar velocidad del motor DC con encoder,hay una variable llamada resolution (resolucion) ¿que es resolution?¿porque la variable resolution=374.22? ¿De donde se obtendira el valor de la variable resolution?¿Es una caracteristica del motor que se encuentra en el datashhet?¿En que parte del datasheet se encontraria?

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

[Responder](#)



Hola Ronaldo, eso esta explicado ahí arriba antes del código, esa Sergio C, variable es la resolución del encoder (Número de pulsos por giro) en este caso es de 374.22. Ese valor te lo dà el datasheet del motor. Saludos.

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



Buenos días señor Sergio Andres Castaño Giraldo

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

Ronaldo Sanchez ,

[Responder](#)



Realmente desconozco si existe una librería. De igual forma, en esta entrada mostramos como manipular el motor bien sea por velocidad o por posición, con esto puedes implementar un

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

[Responder](#)



La libreria que uso no es ATOMIC BLOCK sino Ronaldo Sanchez,
la encuentro 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 ,](#)

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



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

[Responder](#)



Cómo se sustituye la shield por el módulo?

Jorge Vera ,

[Responder](#)

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



Muchas gracias ingeniero Sergio

Luis fernando muñoz ,

[Responder](#)



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

Sergio C ,

[Responder](#)



Muchas gracias, esta buenísimo todo el proceso.

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](#)

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