

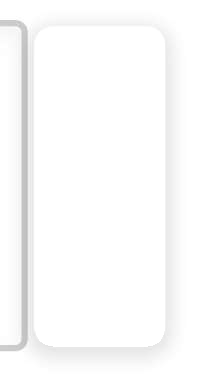
Acep ar

Rech zar

ARDUINO

**Motor DC con Encoder - Velocidad - Posición**

Inicio» Arduino » **Motor DC con Encoder - Velocidad - Posición**



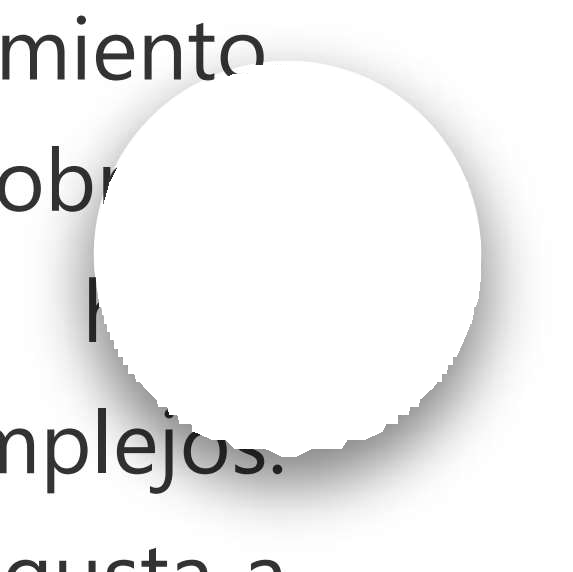
Buscar:

**f**

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.

**SERGIO ANDRÉS CASTAÑO GIRALDO**

Mi nombre es Sergio Andres Castaño Giralda, 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 crecí



*voy* a ir publicando material s asunto desde temas básicos temas un poco más com

Suscríbete al sitio web, dale me gusta a

la página en Facebook y únete al canal



te sitio web usa Cookies de terceros y propios, si continuas navegando las aceptas. - - L er Más

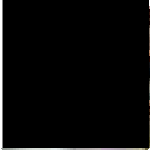
* 1. Gravity: 2x2A Motor Shield for Arduino Twin
  2. Metal DC Geared Motor w/Encoder - 6V 210RPM 1OKg.cm
  3. Kit de Iniciación de Arduino (Alta Calidad)

1. Como funciona un encoder de cuadratura de Efecto Hall
   1. Salidas Digitales en Arduino
   2. Medidor de Velocidad de un Motor DC con PIC
   3. Referencia Analógica AREF en Arduino
2. Medir la velocidad y posición motor DC
   1. Medición de posición
   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



ouTube-

**ista de Materiales**

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

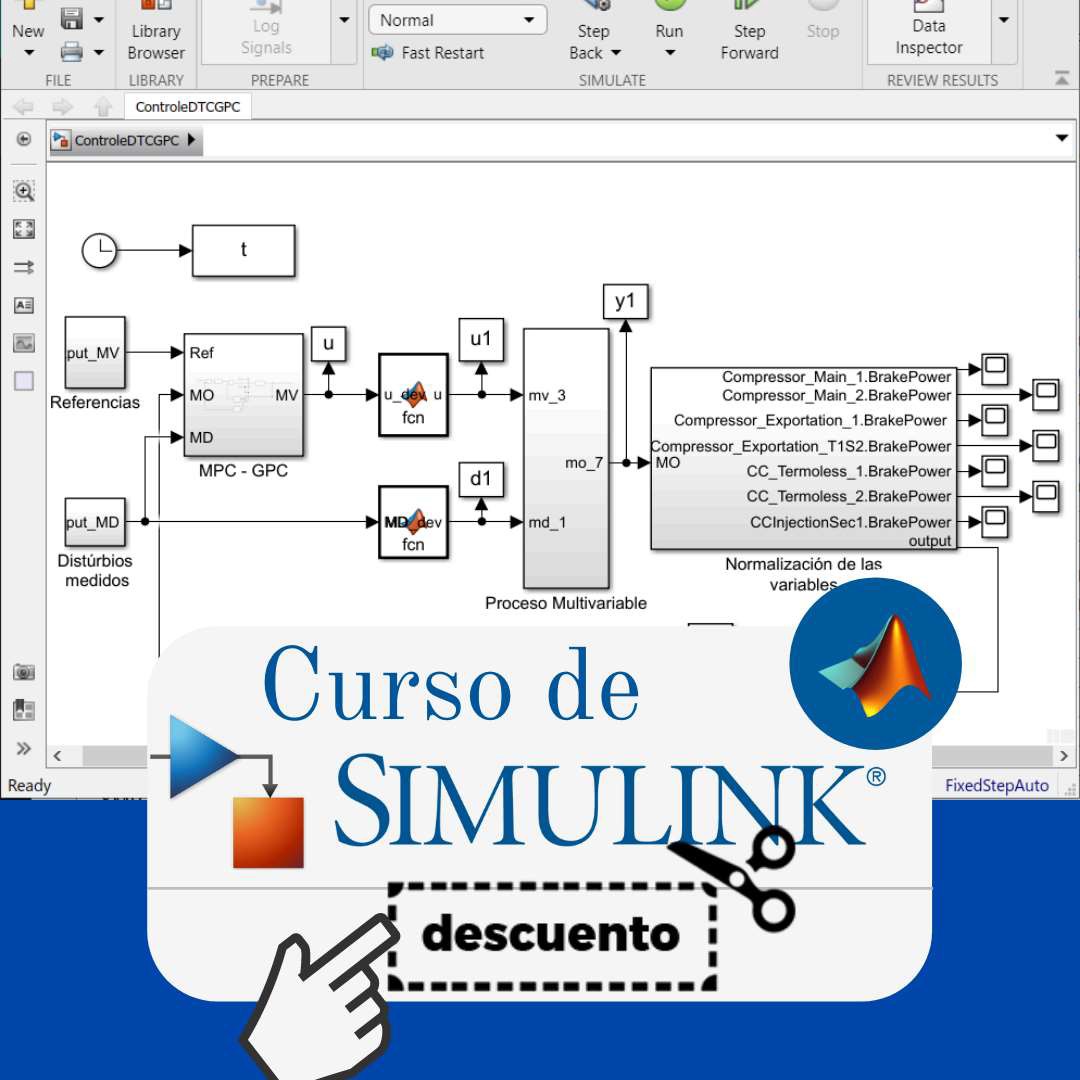
Dando Click en las siguientes imágenes, tendrás un **CUPÓN de DESCUENTO**

=

que es **Exclusivo** del sitio web, dado

que si buscas los cursos en Udemy **N<:r-­**

vas a obtener ningún descuento.



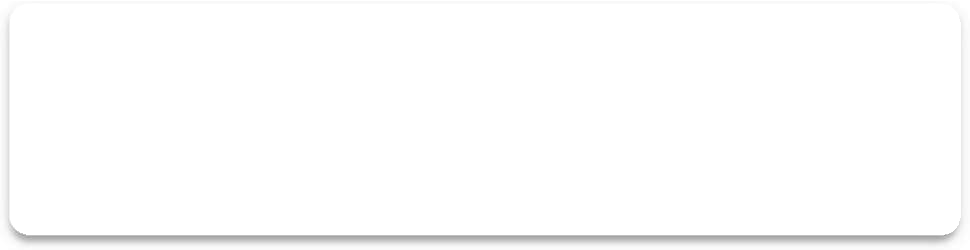
A eptar

R chazar

* Driver de Potencia Puente H (L298)
* Potenciometro Sk (o cualquiera)
* 2 Pulsadores
* Cables
* Fuente de Alimentción de Sv o 12v dependiendo del motor empleado.

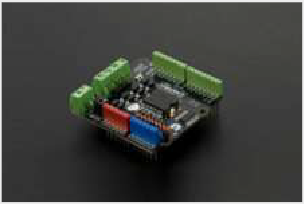
 

**e**



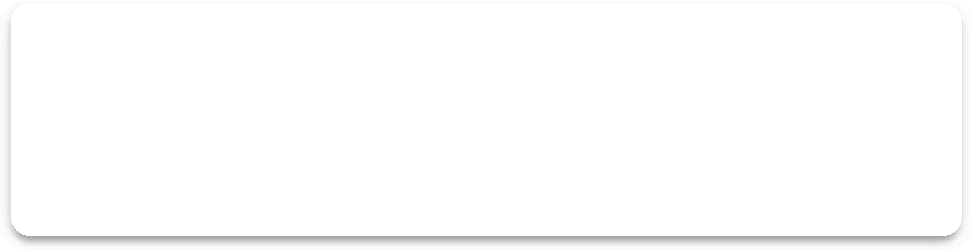
**vity: 2x2A Motor Shield for Arduino Twin**

**Comprar en DFRobot**



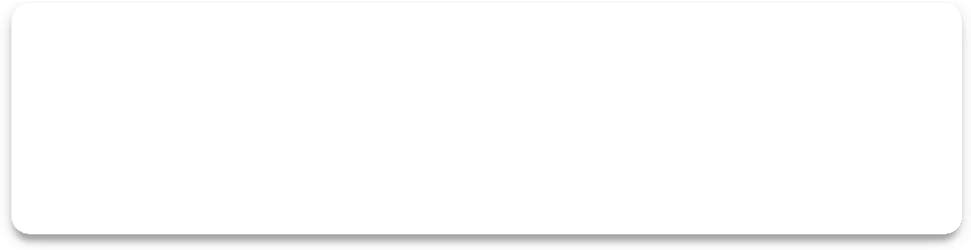
**Geared Motor w/Encoder - 6V 210RPM**

**10Kg.cm**



**iación de Arduino (Alta Calidad)**

 Accede al curso de



-

**FUNDAMENTOS EN INSTRUMENTACIÓN INDUSTRIAL=**

dando click en la imagen de aquí abajo





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

**cuadratura de Efecto Ha**

eer Más

#### SÍGUEME EN



Motor DC con Encoder - Arduino [Velocidad y Posición]

DFRobot

►

**CANAL EN TELEGRAM**

Únete a @ConAutEdu en

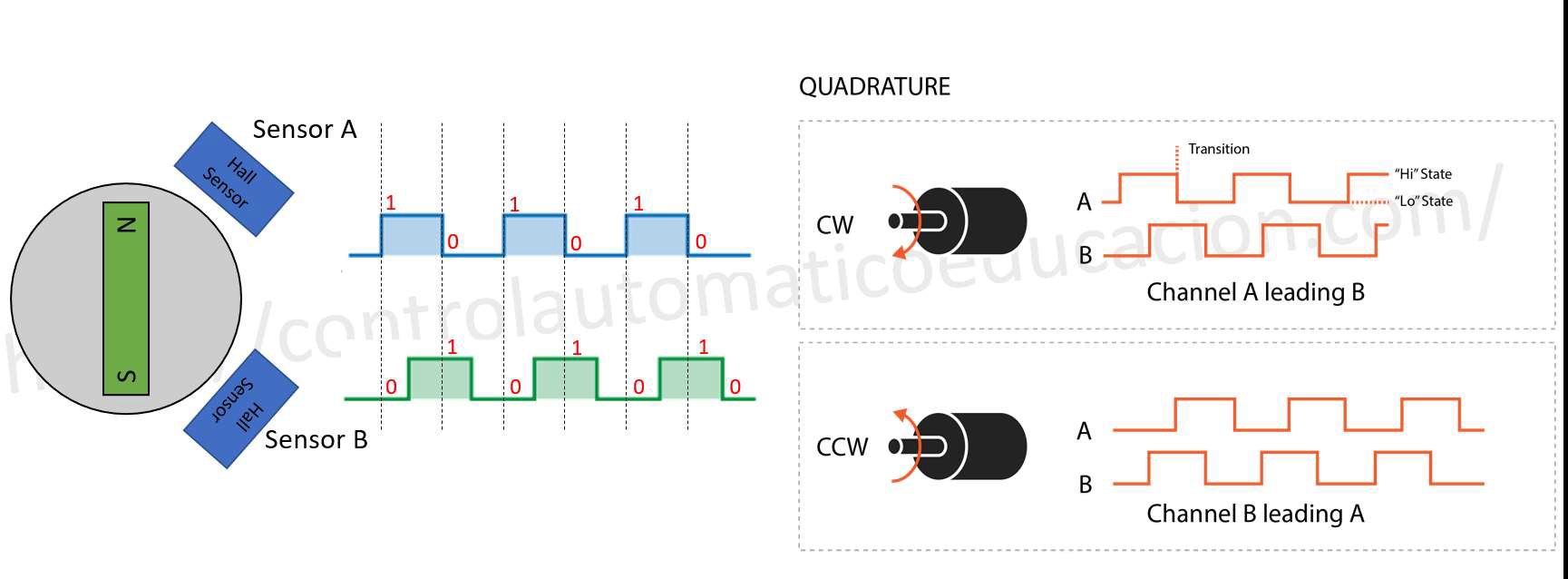
Telegram

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



(t asd cto ) a 2 señales c ad adas q e p esentan n desfase de 90º G acias a





Somos 100.000 ...

►

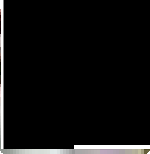
ando el imán gira en el **sentido de las agujas del reloj,** la salida del sensor A se activará primero. Cuando se **gira en sentido antihorario,** por otro lado, la salida del sensor B se activará primero



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

**CANALES DE VOUTUBE**

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

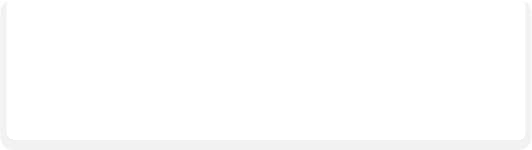
Sergio A. Castaño Giraldo



ouTube-

oce pode-se inscrever no Canal de YouTube em Portugues:

Sergio A. Castaño Giraldo - Brasil



uTube

**GUE LA PÁGINA DE FACEBOOK**

**-**

**CURSOS**



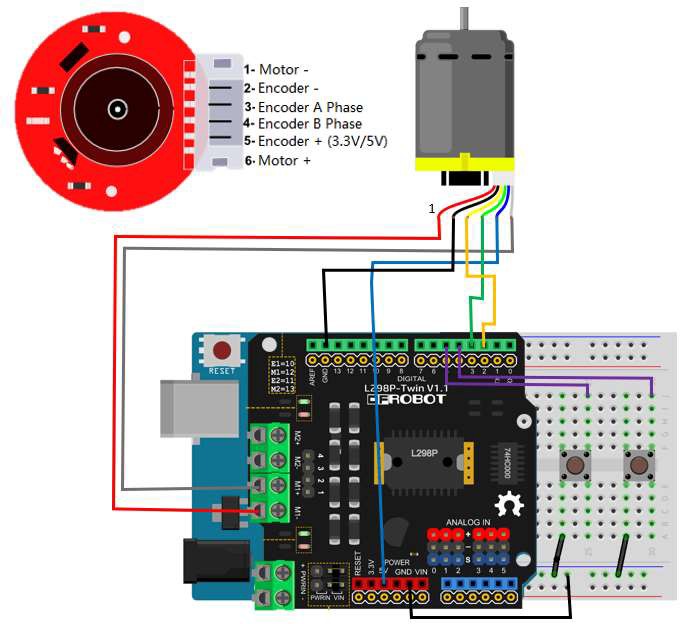
Ac ptar

Re hazar

los 100K suscriptores en el Canal!!



ste sitio web usa Cookies de terceros y propios, si continuas navegando las aceptas. -  **eer Más**



Control de Procesos Control Predictivo Control Realimentado

Herramientas y Tecnologías de Desarrollo

Instrumentación Labview MATLAB

Microcontroladores (PIC) MicroPython

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

Python desde Cero

Sistemas Dinámicos Lineales

**ACCESO**

  Invítame a un Café  Acerca de mi



sobretemperatura

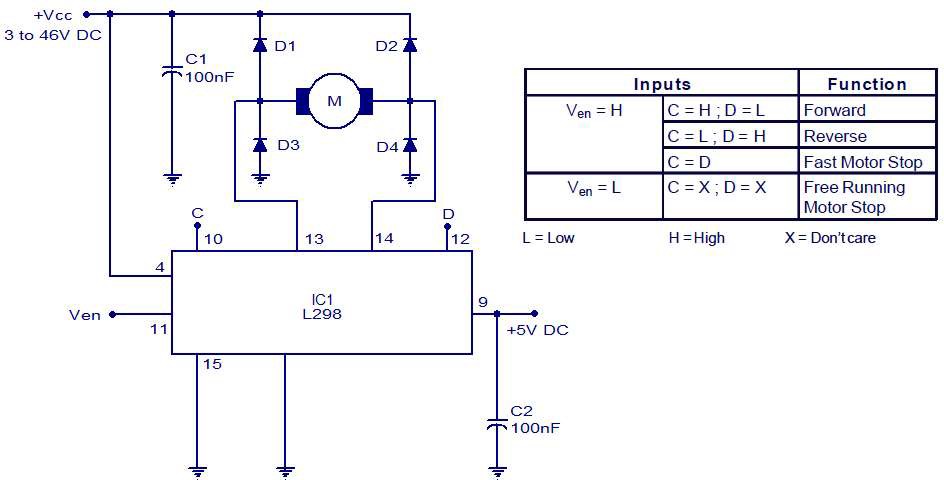
ste sitio web usa Cookies de terceros y propios, si continuas navegando las aceptas. -



### te s ó de a e tac ó . estado de oto depe de á de e óg co de los pines 1O, 11, 12 y se describe en la tabla que se muestra debajo del esquema del circuito.

**eer Más**

**Politica de Cookies**

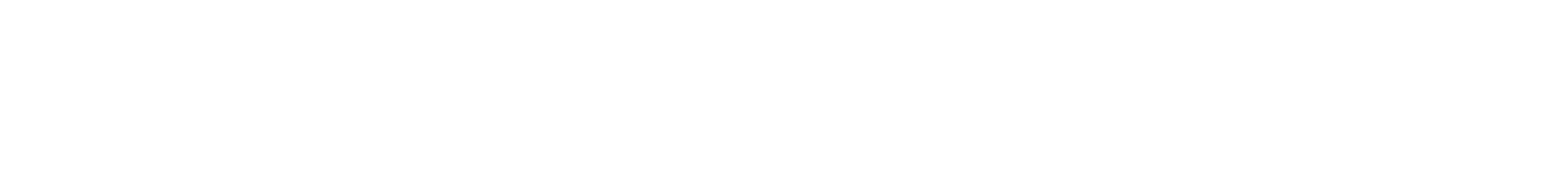




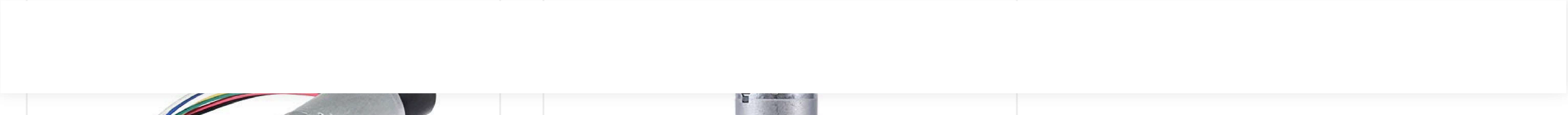
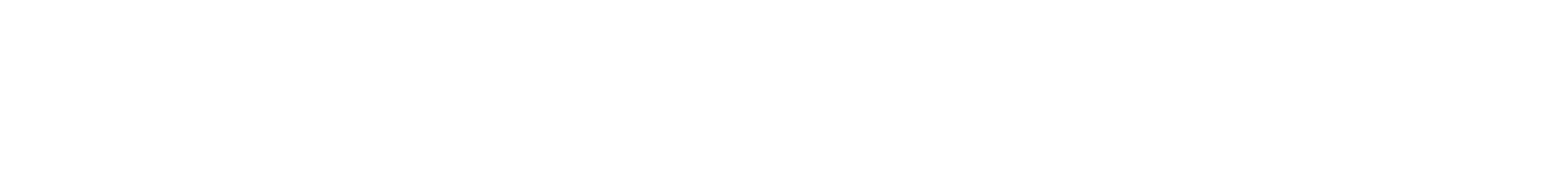
BESTSELLER NO. 1

BESTSELLER NO. 2

REBAJAS



Ver en Amazon



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

**eer Más**

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

(104)

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

(11)

tt&.39

Ver en Amazon



BESTSELLER NO. 3



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

**eer Más**

### El código en Arduino es:



**Ver en Amazon**

l. //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//



1. //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//
2. //\*\*\*\*\*
3. //\*\*\*\*\*

Posición y Velocidad de Motor DC \*\*\*\*\*//

\*\*\*\*\*//

23

//\*\*\*\*\* by: Sergio Andres Castaño Giraldo \*\*\*\*\*//

//\*\*\*\*\* https://controlautomaticoeducacion.com/ \*\*\*\*\*//

//\*\*\*\*\* \*\*\*\*\*//

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

**#define ENCODER A 2** // **Amarillo #define ENCODER B 3** // **Verde #define BUTTON FORWARD 4**

**#define BUTTON BACKWARD 5**

// Pines de Control Shield const **int** ElPin 10; const **int** MlPin 12; const **int** E2Pin 11; const **int** M2Pin 13;

**typedef struct{**

byte enPin;

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

**eer Más**

|  |  |  |
| --- | --- | --- |
| 27. | //Creo | el motor |
| 28. | const | Motor motor {ElPin, MlPin}; |
| 29. |  |  |
| 30. | const | **int** Forward = LOW; |
| 31. | const | **int** Backward = HIGH; |
| 32. |  |  |

33.



34.

35.

36.

37.

38.

39.

40.

41.

42.

43.

44.

45.

46.

47.

48.

49.

50.

51.

52.

53.

54.

55.

56

**void** setup(){ Serial.begin(9600);

//Encoders como entradas pinMode(ENCODER\_A, INPUT); pinMode(ENCODER\_B, INPUT);

//Pulsadores

pinMode(BUTTON\_FORWARD, INPUT\_PULLUP); pinMode(BUTTON\_BACKWARD, INPUT PULLUP);

//Configura Motor pinMode(motor.enPin, OUTPUT); pinMode(motor.directionPin, OUTPUT);

**void** loop(){

// Pulsador hacia adelante

**if(!** digitalRead(BUTTON FORWARD)){ digitalWrite(motor.directionPin, Forward); digitalWrite(motor.enPin, HIGH);

imprimir cuadratura();

**else if(!** digitalRead(BUTTON BACKWARD)){ digitalWrite(motor.directionPin, Backward); digitalWrite(motor.enPin, HIGH);

imprimir cuadratura();



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

**eer Más**

60.

61.

62.

63.

64.

65.

66.

67.

68.

69.

**void** imprimir cuadratura() {

**int** a

**int** b

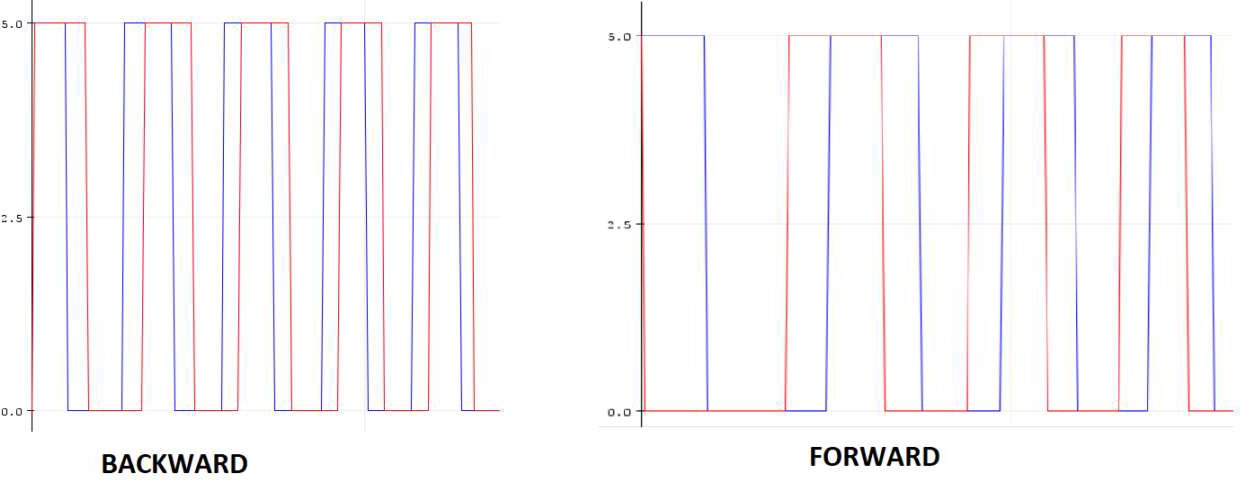
digitalRead(ENCODER\_A);

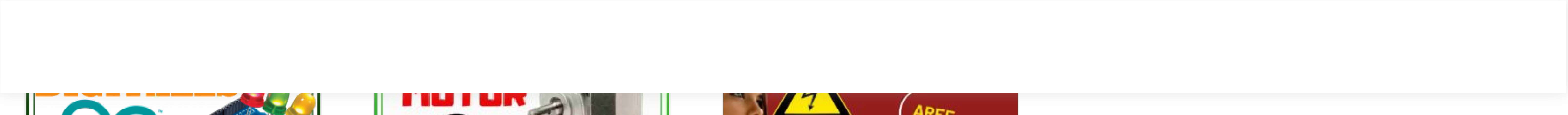
digitalRead(ENCODER\_B);

Serial.print(a\*S); Serial.print(" ");

Serial.println(b\*S);

**La señal de cuadratura observada por el serial plotter:**





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

eer 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.

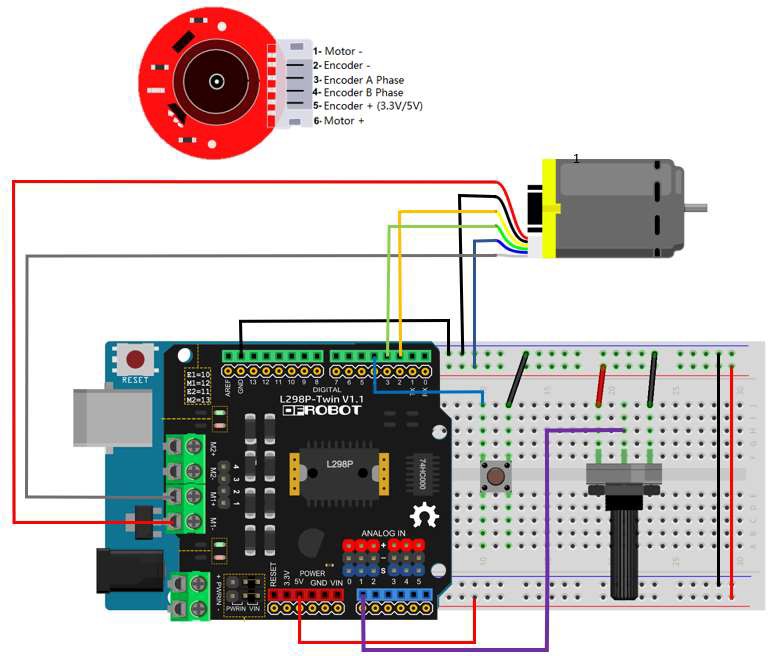
**Medir la velocidad y posición motor**

# 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

ste sitio web usa Cookies de terceros y propios, si continuas navegando las aceptas. -  **eer 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 Potenciometro.**

## Medición de posición



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

eer Más

te upe ó co e a co de sub da.

a te upe ó se e e e ta u a

variable que llamamos **theta** (si la salida del encoder B es alta) o se resta uno (si la salida del encoder B es baja).

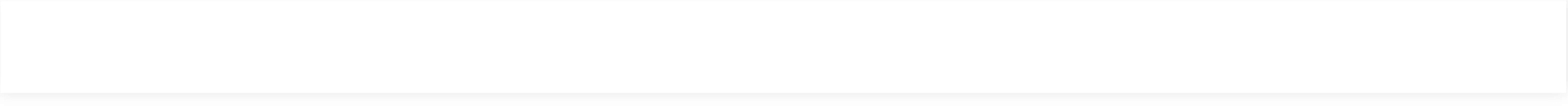
Para garantizar que la variable *theta* se almacene de modo que pueda ser leída con precisión por las funciones de loop y de interrupción, debe utilizar el calificador *volátil* .

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 potenciometro 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.



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

**eer 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 (Numero 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.



l.

2.

3.

4.

5.

6.

7.

8.

9.

10.

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

//\*\*\*\*\*

//\*\*\*\*\*

//\*\*\*\*\*

//\*\*\*\*\*

//\*\*\*\*\*

Posición

y Velocidad de Motor DC

by: Sergio

Andres Castaño Giralda

https://controlautomaticoeducacion.com/

\*\*\*\*\*//

\*\*\*\*\*//

\*\*\*\*\*//

\*\*\*\*\*//

\*\*\*\*\*//

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

ste sitio web usa Cookies de terceros y propios, si continuas navegando las aceptas. -  **eer Más**

14.

15.

16.

17.

18.

19.

20.

21.

**#define ENCODER A #define ENCODER B #define BUTTON MOD**

// Pin del Potenciometro const **int** pot = AO;

1. **// Amarillo**
2. // **Verde**

**4**

22.



43

// Pines de Control Shield const **int** ElPin 10; const **int** MlPin 12; const **int** E2Pin 11; const **int** M2Pin 13;

//Variable global de posición compartida con la interrupción volatile **int** theta = O;

//Variable global de pulsos compartida con la interrupción volatile **int** pulsos= O;

unsigned **long** timeold;

**float** resolution = 374.22;

//Variable Global Velocidad

**int** vel = O;

//Variable Global Posicion

**int** ang = O;

//Variable Global MODO

**bool** modo= **false;**

ste sitio web usa Cookies de terceros y propios, si continuas navegando las aceptas. -  **eer Más**

47.



48.

49.

50.

51.

52.

53.

54.

55.

56.

57.

58.

59.

60.

61.

62.

63.

64.

65.

66.

67.

68.

69.

70.

71.

72.

73.

74.

75.

76

byte enPin;

byte directionPin;

}Motor;

//Creo el motor

const Motor motor {ElPin, MlPin};

//Constantes de dirección del Motor const **int** Forward = LOW;

const **int** Backward = HIGH;

**void** setup(){

// set timer 1 divisor to 1024 for PWM frequency of 30.64 TCCRlB = TCCRlB & B11111000 I B00000101;

Serial.begin(9600);

//Encoders como entradas pinMode(ENCODER\_A, INPUT); pinMode(ENCODER\_B, INPUT);

//Pulsadores

pinMode(BUTTON MOD, INPUT PULLUP);

//Configura Motor pinMode(motor.enPin, OUTPUT); pinMode(motor.directionPin, OUTPUT);

//Configurar Interrupción timeold = O;

attachinterrupt(digitalPinTointerrupt(ENCODER\_A),leerEncodE

**void** loop(){

**float** posicion;

ste sitio web usa Cookies de terceros y propios, si continuas navegando las aceptas. - **eer Más**

80.



81.

82.

83.

84.

85.

86.

87.

88.

89.

90.

91.

92.

93.

94.

95.

96.

97.

98.

99.

100.

101.

102.

103.

104.

105.

106.

107.

108.

109

//Lee el Valore del Potenciometro value = analogRead(pot);

//Cambia de Modo Velociadad o Posición if(debounce(BUTTON\_MOD)){

modo= !modo;

theta = O;

**if** (modo){

//Transforma el valor del Pota velocidad vel = map(value,0,1023,0,255);

//Activa el motor dirección Forward con la velocidad setMotor(motor, vel, **false);**

//Espera un segundo para el calculo de las RPM

**if** (millis() - timeold >= 1000)

//Modifica las variables de la interrupción forma atómj ATOMIC\_BLOCK(ATOMIC\_RESTORESTATE){

//rpm = float(pulsos \* 60.0 / 374.22); //RPM

rpm = **float((60.0** \* 1000.0 / resolution) / (millis() timeold = millis();

pulsos= O;

Serial.print("RPM: "); Serial.println(rpm); Serial.print("PWM: "); Serial println(vel);

ste sitio web usa Cookies de terceros y propios, si continuas navegando las aceptas. - **eer Más**

|  |  |  |
| --- | --- | --- |
| 113. | //Transforma el valor del Pota ángulo |  |
| 114. | ang = map(value,0,1023,0,360); |
| 115. |  |
| 116. | //Modifica las variables de la interrupción forma | atómicc |
| 117. | ATOMIC BLOCK(ATOMIC RESTORESTATE){ |  |
| 118. | posicion = **(float(theta** \* 360.0 /resolution)); |  |

119.

120.

1. //Posiciona el ángulo con tolerancia+- 2
2. if(ang > posicion+2){

vel dir

200;

**true;**

1. **else** if(ang < posicion-2){

|  |  |  |
| --- | --- | --- |
| 127. | vel | 200; |
| 128. | dir | **false;** |
| 129. |  |  |
| 130. | **else{** |  |
| 131. | vel | O; |
| 132. |  |  |

133.



134.

135.

136.

137.

138.

139.

140.

141.

142

setMotor(motor, vel, dir);

//Función para dirección y velocidad del Motor

**void** setMotor(const Motor motor, **int** vel, **bool** dir){ analogWrite(motor.enPin, vel);

**if**(dir)

digitalWrite(motor.directionPin, Forward);

**else**

ste sitio web usa Cookies de terceros y propios, si continuas navegando las aceptas. -  **eer Más**

146.



147.

148.

149.

150.

151.

152.

153.

154.

155.

156.

157.

158.

159.

160.

161.

162.

163.

164.

165.

166.

167.

168.

169.

170.

171.

172.

173.

174.

175

//Función anti-rebote

**bool** debounce(byte input){

**bool** state = **false;**

**if(!** digitalRead(input)){ delay(200);

**while(!** digitalRead(input));

delay(200); state = **true;**

**return** state;

//Función para la lectura del encoder

**void** leerEncoder() {

//Lectura de Velocidad if(modo)

pulsos++; //Incrementa una revolución

//Lectura de Posición

**else{**

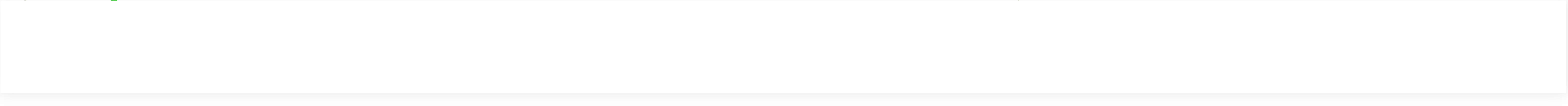
**int** b = digitalRead(ENCODER\_B);

**if** (b > O) {

//Incremento variable global theta++;

**else{**

//Decremento variable global theta--;



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

**eer 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**

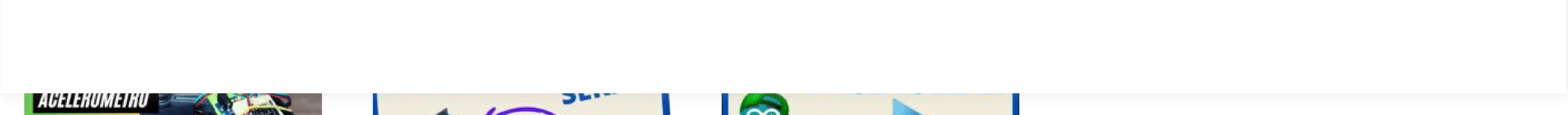


Sensor de Flujo YF­ S201: Medición de Caudal con Arduino

Bluetooth HC-05 / HC06

Balanza Electronica con HX711 y Arduino





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

**eer Más**

**Acelerómetro y**

**Giroscópio**

**Comunicación Serial Arduino Simulink/Matlab**

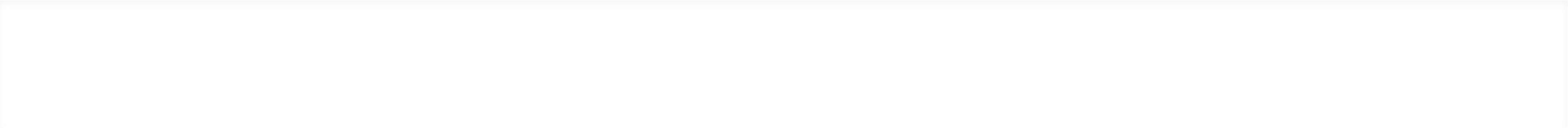
**Arduino con Simulink**

**Deja una respuesta**

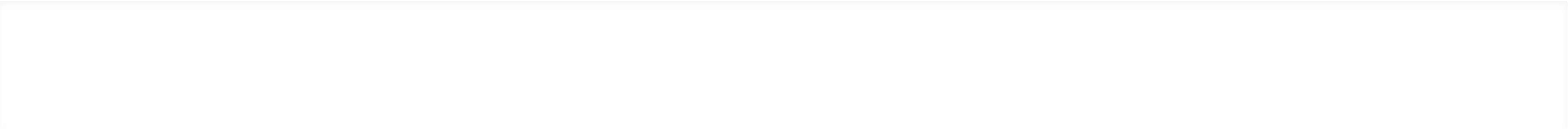
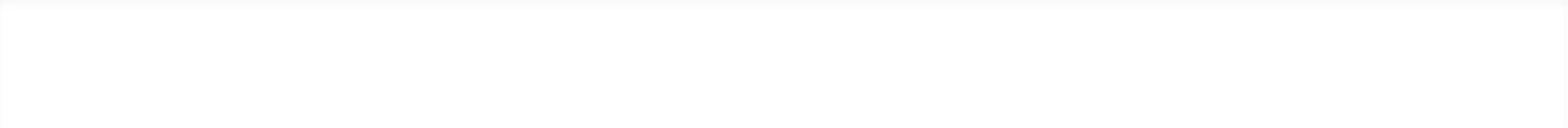
Tu dirección de correo electrónico no será publicada. Los campos obligatorios están marcados con \*



**NOMBRE\***



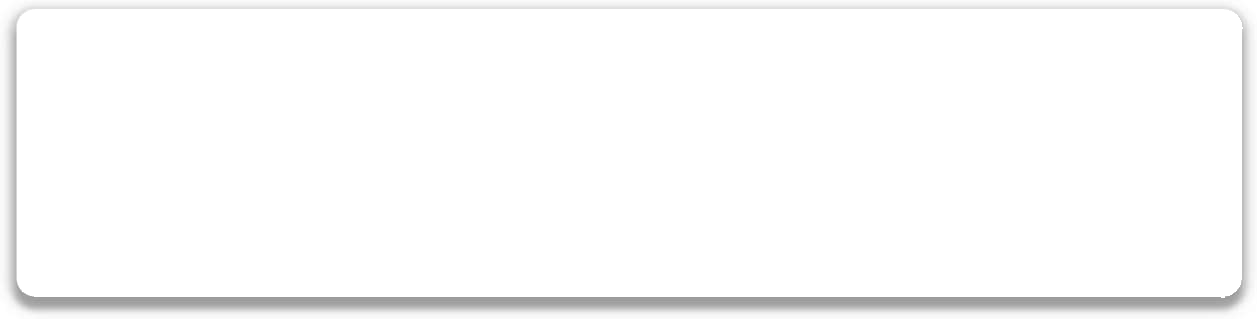
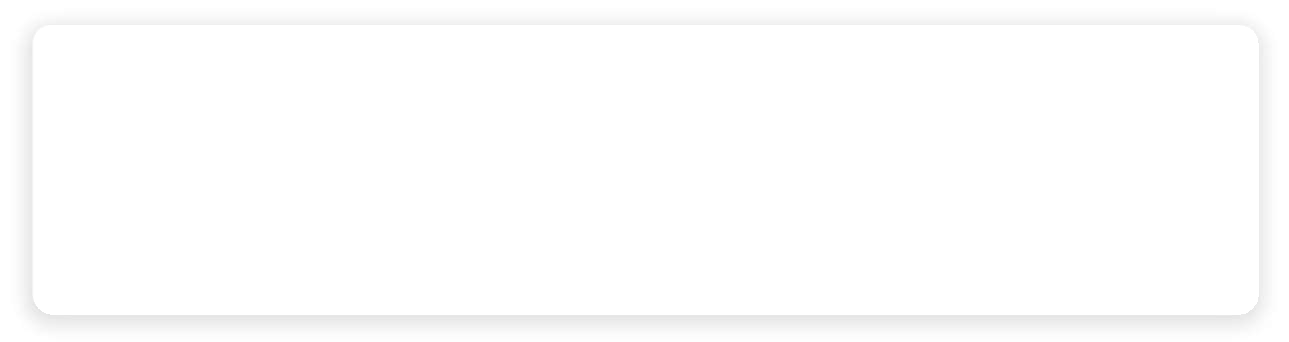
**CORREO ELECTRÓNICO** \*



sitio web usa Cookies de terceros y propios, si continuas navegando las aceptas. - - Lee Más

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

* **RECIBIR UN CORREO ELECTRÓNICO CON CADA NUEVA ENTRADA.**



licar 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,**

ResP-onder

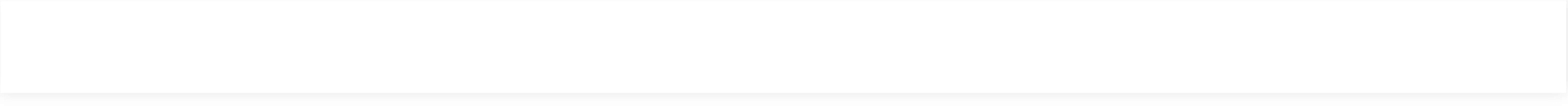


**Si se puede. Sergio C,**

ResP-onder

**rm**

**que programa utilizaste para simular los componentes? Juan,**



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

**eer Más**



Uso el Proteus principalmente Sergio C,

Responder



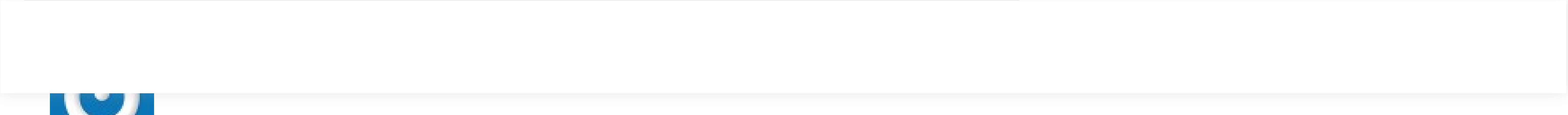
Estupendo tutorial. Muy claro e instructivo.

He visto que en tu web hay varios tutoriales PID, ¿has publicado este mismo codigo de para arduino pero incorporando el algoritmo PID para

Gabilon,

actuar sobre el setMotor de forma que se mantenga la velocidad constante? Muchas gracias

Responder



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

**eer Más**

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

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 ahi arriba antes del código, esa Sergio C, variable es la resolución del encoder (Numero de pulsos por

giro) en este caso es de 374.22. Ese valor te lo dá el datasheet del motor. Saludos.



Buenos días señor Sergio Andres Castaño Giralda Tengo una inquietud

¿Donde se puede conseguir la librería ?

Ronaldo Sanchez,

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 Sergio C, esta entrada mostramos como manipular el motor bien sea por

velocidad o por posición, con esto puedes implementar un

Responder



La librería que uso no es ATOMIC BLOCK sino Ronaldo Sanchez, la encuentro en el Arduino una librería

parecida que se puede descargar desde el Arduino llamada SIMPLY ATOMIC

Responder



Cómo se sustituye la shield por el módulo? Pedro Reyes Mateo,



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

**mostando como realizarlo.**

**Responder**



**Cómo se sustituye la shield por el módulo? Jorge Vera,**

**Responder**



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

**eer Más**

**Muchas gracias ingeniero Sergio luis femando muñoz,**

**Responder**



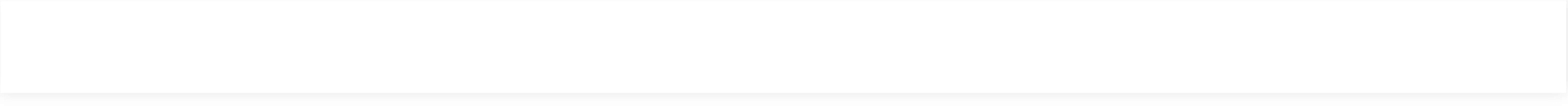
**De nada Luis, que bueno que te ha gustado. Gracias por comentar. Saludos!**

**Sergio C,**

**Responder**



**Muchas gracias, esta buenísimo todo el proceso.**



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

**eer Más**



**De nada Adeluna100, Saludos!! Sergio C,**

Responder

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