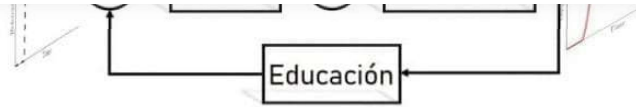


Este sitio web usa Cookies de terceros y propios, si continuas navegando las aceptas.

Aceptar

Rechazar

[Leer Más](#)



[ARDUINO](#)

PWM Arduino

[Inicio](#) » [Arduino](#) » **PWM Arduino**



Hola controleros y controleras en el día de hoy vamos a aprender a usar el **PWM ARDUINO**, o Modulación por Ancho de Pulso (*Pulse Width Modulation -PWM*), el cual es una de las señales de control más importantes y utilizadas con el Arduino.

Contenidos [[ocultar](#)]

1. PWM Arduino
2. Timer – PWM Arduino
 - 2.1. Motor Paso a Paso Arduino
 - 2.2. Display 7 Segmentos – Antirebote – PullUP con Arduino
 - 2.3. XBEE modo AT
3. Configurar el PWM en Arduino
 - 3.1. Sintaxis
 - 3.2. Parámetros
4. PWM Frecuencia (Frequency) Arduino
 - 4.1. ¿Cómo se cambia la frecuencia PWM?
5. PWM Led Arduino
6. Motor DC Arduino – PWM
 - 6.1. Bomba 12 V 130PSI
 - 6.2. Bomba cc 12 V 70PSI 3.5L/min
 - 6.3. Driver para Control de Motores BTS7960 43A de alta potencia
7. Códigos
 - 7.1. LED
 - 7.2. Motobomba



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

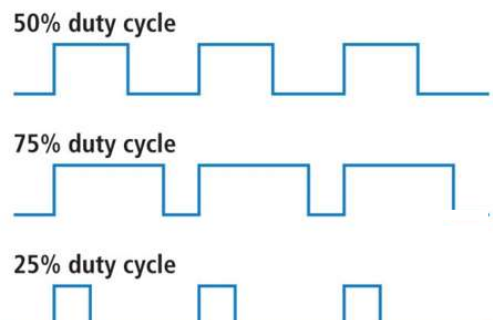
[Leer Más](#)

PWM Arduino

Salida PWM Arduino ❤️ [Cómo Cambiar Frecuencia PWM] # 018



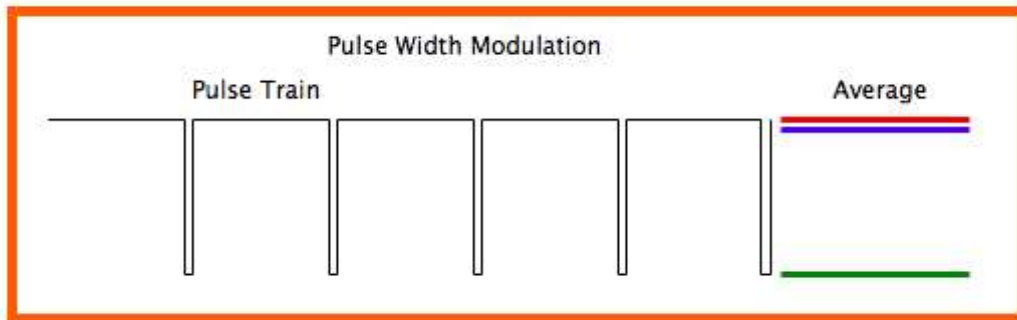
La señal de PWM con Arduino es generada en los pines especificados con el símbolo (~), en la mayoría de las placas Arduino disponibles en el mercado (aquellas que tienen como corazón el microcontrolador **ATmega168** o **ATmega328**), las Salidas PWM o **PWM Output Arduino** se encuentran en los pines **3, 5, 6, 9, 10 y 11**. En Arduino Mega, esto funciona del pin **2 al 13** y del **44 al 46**. Y en modelos más antiguos con un micro **ATmega8** solo tiene como PWM Output en Arduino los pines **9, 10 y 11**, A continuación, podemos ver una imagen con la ubicación de las Salidas PWM en un Arduino UNO.



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

[Leer Más](#)

Por ejemplo, el PWM de un PIN en Arduino es simplemente una señal binaria (0v o 5v) que podremos configurar para que trabaje un determinado tiempo en **Encendido o 5v** y el resto de tiempo en **Apagado o 0v**, y repita este procedimiento infinitamente, pudiendo variar este ancho de pulso en cualquier instante, tal y como se puede observar en la siguiente figura, que representa el **PWM en un PIN de Arduino**.



15. PWM

Esto es interesante, porque el voltaje medio que se le aplica a la carga es proporcional al tiempo que la señal se mantuvo en 5v y con esto podemos lograr varias aplicaciones como controlar la luminosidad de un LED (**PWM Led Arduino**), la velocidad de un motor (**PWM Motor Arduino**), regular resistencias calefactoras a través de relés de estado sólido, [controlar un servomotor con Arduino](#), etc.

Recordemos que una resolución de 8bits en una salida PWM significa que tenemos 256 niveles. Es decir, representamos el Duty cycle mediante un número de 0 a 255.

Timer – PWM Arduino

Como hablamos en la entrada de Timer Arduino, la señal y frecuencia del PWM del Arduino están gobernados por los diferentes temporizadores o TIMERS que posee la placa de desarrollo.

Cada uno de los Timers del Arduino está en la capacidad de controlar 2 o 3 pwm asignando un registro de comparación para cada señal, de tal forma que cuando el temporizador alcance el valor del registro de comparación simplemente cambia el



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

[Leer Más](#)

BOARD	PWM PINS	PWM FREQUENCY
Uno, Nano, Mini	3, 5, 6, 9, 10, 11	490 Hz (pins 5 and 6: 980 Hz)
Mega	2 – 13, 44 – 46	490 Hz (pins 4 and 13: 980 Hz)
Leonardo, Micro, Yún	3, 5, 6, 9, 10, 11, 13	490 Hz (pins 3 and 11: 980 Hz)
Uno WiFi Rev.2	3, 5, 6, 9, 10	976 Hz
MKR boards *	0 – 8, 10, A3 (18), A4 (19)	732 Hz
MKR1000 WiFi *	0 – 8, 10, 11, A3 (18), A4 (19)	732 Hz
Zero *	3 – 13, A0 (14), A1 (15)	732 Hz
Due **	2-13	1000 Hz
101	3, 5, 6, 9	pins 3 and 9: 490 Hz, pins 5 and 6: 980 Hz

El PWM en los pines 5 y 6 tendrán ciclos de trabajo más altos de lo esperado. Esto se debe a las interacciones con las funciones `millis()` y `delay()`, que comparten el mismo TIMER interno utilizado para generar esas salidas PWM. Esto se notará principalmente en configuraciones de ciclo de trabajo bajo (por ejemplo, 0-10) y puede dar como resultado que un valor de 0 no apague completamente la salida en los pines 5 y 6.



Motor Paso a Paso Arduino



Display 7 Segmentos – Antirebote – PullUP con Arduino



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

[Leer Más](#)



XBEE modo AT

Configurar el PWM en Arduino

Para la configuración del PWM Output Arduino simplemente utilizamos la siguiente instrucción:

Sintaxis

`analogWrite (pin, valor)`

Parámetros

pin: el pin PWM Arduino a ser escrito. Tipos de datos permitidos: int.

valor: el ciclo de trabajo: entre 0 (siempre apagado) y 255 (siempre encendido). Tipos de datos permitidos: int.

No products found.

PWM Frecuencia (Frequency) Arduino

La señal PWM está determinada por la velocidad del TIMER. Suponiendo que está utilizando un Arduino UNO, el reloj de este contador es igual al reloj del sistema dividido por un valor de preescaler. El preescaler es un valor de 3 bits almacenado en los tres bits menos significativos del registro del TIMER: CS02, CS01 y CS00.

Existen tres registros de temporizador / contador: **TCCR0B**, **TCCR1B** y **TCCR2B**



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

[Leer Más](#)

emitir una señal PWM a una determinada frecuencia. Los pines Arduino **9 y 10** están controlados por **TCCR1B**, por lo que pueden configurarse a una frecuencia diferente de los pines 6 y 5. Los pines Arduino **11 y 3** están controlados por **TCCR2B**, por lo que pueden configurarse a una tercera frecuencia. Pero no puede establecer diferentes frecuencias para los pines controlados por el mismo preescalador (por ejemplo, los pines 6 y 5 deben estar en la misma frecuencia).

¿Cómo se cambia la frecuencia PWM?

En void setup () establezca o borre los bits **CS02, CS01 y CS00** en el registro TCCRnB correspondiente.

Para Arduino Uno, Nano, YourDuino RoboRED, Mini Driver, Lilly Pad y cualquier otra placa que use ATmega 8, 168 o 328

```

1. //Set PWM frequency for D5 & D6 -----
2.
3. //TCCR0B = TCCR0B & B11111000 | B00000001; // set timer 0 divisor
4. //TCCR0B = TCCR0B & B11111000 | B00000010; // set timer 0 divisor
5. TCCR0B = TCCR0B & B11111000 | B00000011; // set timer 0 divisor
6. //TCCR0B = TCCR0B & B11111000 | B00000100; // set timer 0 divisor
7. //TCCR0B = TCCR0B & B11111000 | B00000101; // set timer 0 divisor
8.
9.
10. // Set PWM frequency for D9 & D10 -----
11.
12. //TCCR1B = TCCR1B & B11111000 | B00000001; // set timer 1 divisor
13. //TCCR1B = TCCR1B & B11111000 | B00000010; // set timer 1 divisor
14. TCCR1B = TCCR1B & B11111000 | B00000011; // set timer 1 divisor
15. //TCCR1B = TCCR1B & B11111000 | B00000100; // set timer 1 divisor
16. //TCCR1B = TCCR1B & B11111000 | B00000101; // set timer 1 divisor
17.
18. // Set PWM frequency for D3 & D11 -----
19.
20. //TCCR2B = TCCR2B & B11111000 | B00000001; // set timer 2 divisor
21. //TCCR2B = TCCR2B & B11111000 | B00000010; // set timer 2 divisor
22. //TCCR2B = TCCR2B & B11111000 | B00000011; // set timer 2 divisor
23. TCCR2B = TCCR2B & B11111000 | B00000100; // set timer 2 divisor

```



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

[Leer Más](#)

ATmega1280 o ATmega2560

```

1. // Set PWM frequency for D4 & D13 -----
2. //TCCR0B = TCCR0B & B11111000 | B00000001; // set timer 0 divisor
3. //TCCR0B = TCCR0B & B11111000 | B00000010; // set timer 0 divisor
4. TCCR0B = TCCR0B & B11111000 | B00000011; <>// set timer 0 divisor
5. //TCCR0B = TCCR0B & B11111000 | B00000100; // set timer 0 divisor
6. //TCCR0B = TCCR0B & B11111000 | B00000101; // set timer 0 divisor
7. // Set PWM frequency for D11 & D12 -----
8. //TCCR1B = TCCR1B & B11111000 | B00000001; // set timer 1 divisor
9. //TCCR1B = TCCR1B & B11111000 | B00000010; // set timer 1 divisor
10. TCCR1B = TCCR1B & B11111000 | B00000011; // set timer 1 divisor
11. //TCCR1B = TCCR1B & B11111000 | B00000100; // set timer 1 divisor
12. //TCCR1B = TCCR1B & B11111000 | B00000101; // set timer 1 divisor
13. //Set PWM frequency for D9 D10 -----
14. //TCCR2B = TCCR2B & B11111000 | B00000001; // set timer 2 divisor
15. //TCCR2B = TCCR2B & B11111000 | B00000010; // set timer 2 divisor
16. //TCCR2B = TCCR2B & B11111000 | B00000011; // set timer 2 divisor
17. TCCR2B = TCCR2B & B11111000 | B00000100; // set timer 2 divisor
18. //TCCR2B = TCCR2B & B11111000 | B00000101; // set timer 2 divisor
19. //TCCR2B = TCCR2B & B11111000 | B00000110; // set timer 2 divisor
20. //TCCR2B = TCCR2B & B11111000 | B00000111; // set timer 2 divisor
21. // Set PWM frequency for D2, D3 & D5 -----
22. //TCCR3B = TCCR3B & B11111000 | B00000001; // set timer 3 divisor
23. //TCCR3B = TCCR3B & B11111000 | B00000010; // set timer 3 divisor
24. TCCR3B = TCCR3B & B11111000 | B00000011; // set timer 3 divisor
25. //TCCR3B = TCCR3B & B11111000 | B00000100; // set timer 3 divisor
26. //TCCR3B = TCCR3B & B11111000 | B00000101; // set timer 3 divisor
27. // Set PWM frequency for D6, D7 & D8 -----
28. //TCCR4B = TCCR4B & B11111000 | B00000001; // set timer 4 divisor
29. //TCCR4B = TCCR4B & B11111000 | B00000010; // set timer 4 divisor
30. TCCR4B = TCCR4B & B11111000 | B00000011; // set timer 4 divisor
31. //TCCR4B = TCCR4B & B11111000 | B00000100; // set timer 4 divisor
32. //TCCR4B = TCCR4B & B11111000 | B00000101; // set timer 4 divisor
33. //Set PWM frequency for D44, D45 & D46 -----
34. //TCCR5B = TCCR5B & B11111000 | B00000001; // set timer 5 divisor
35. //TCCR5B = TCCR5B & B11111000 | B00000010; // set timer 5 divisor
36. TCCR5B = TCCR5B & B11111000 | B00000011; // set timer 5 divisor
37. //TCCR5B = TCCR5B & B11111000 | B00000100; // set timer 5 divisor

```



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

[Leer Más](#)

LED PWM Arduino 🚩 [Variar INTENSIDAD] #019

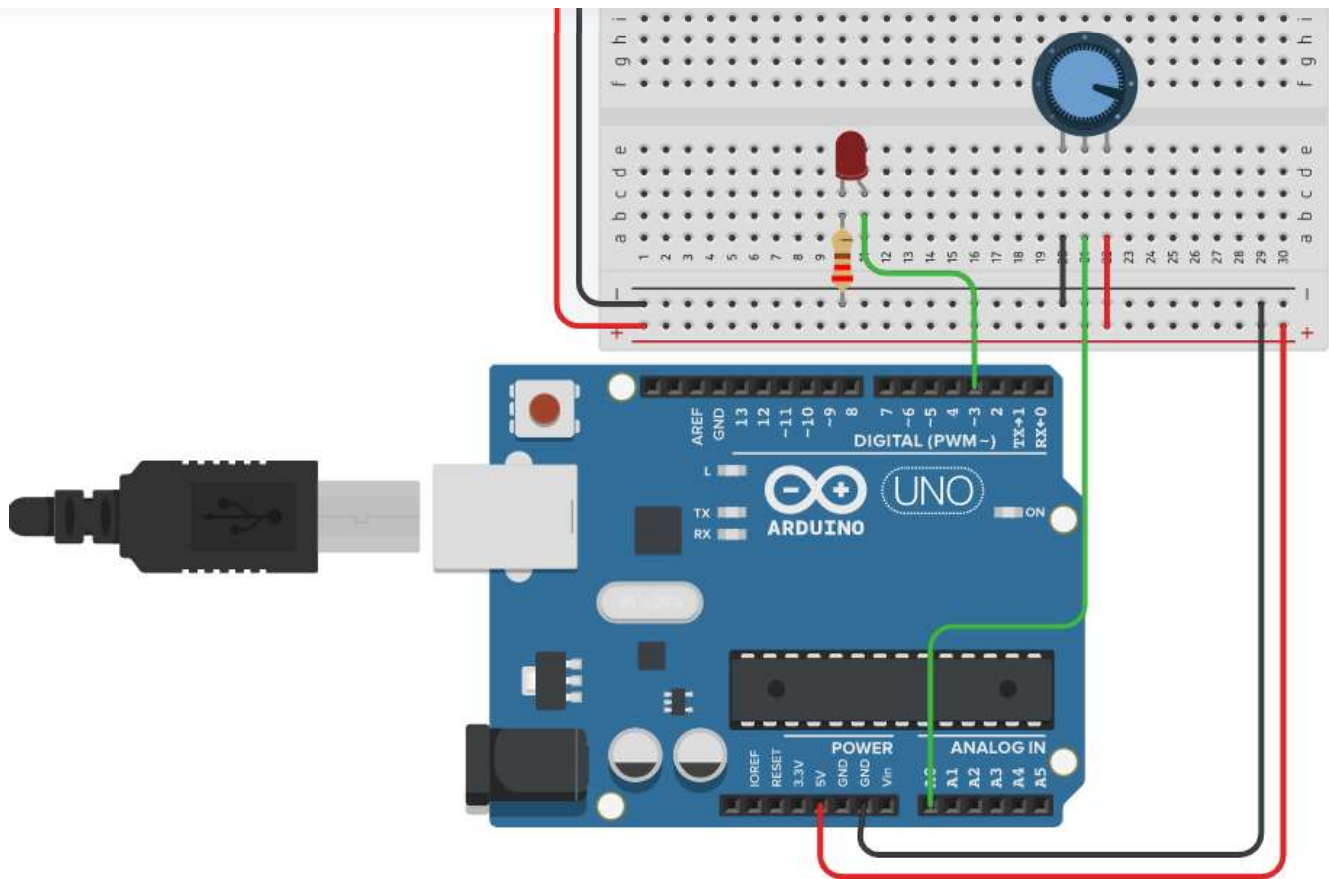


En este primer ejemplo veremos cómo variar la intensidad de un LED y el PWM es perfecto para controlar dicha intensidad del LED con Arduino. Para el control PWM del LED con Arduino vamos a utilizar un potenciómetro conectado a un Pin Análogo.



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

[Leer Más](#)



Motor DC Arduino – PWM

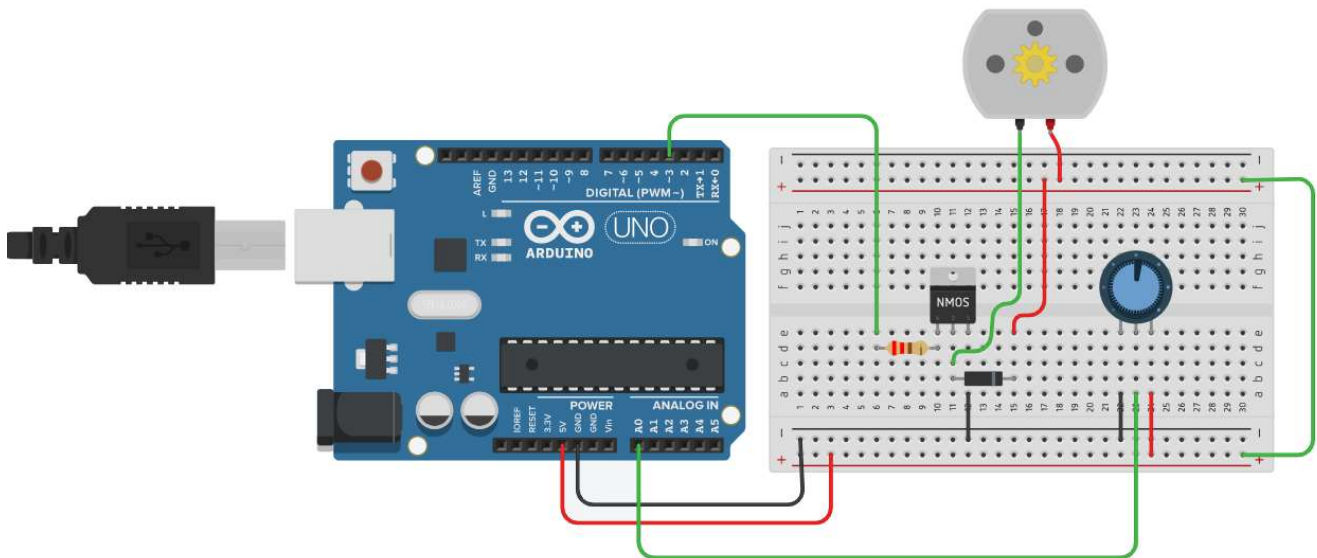


Este sitio web usa Cookies de terceros y propios, si continúas navegando las aceptas. [Aceptar](#) [Rechazar](#)

[Leer Más](#)



Con esta práctica vamos a aprender como variar la velocidad de un motor DC con Arduino. Para eso vamos a utilizar un potenciómetro, un mosfet IRFZ44N, un diodo 1N4001, una resistencia de 220ohms.

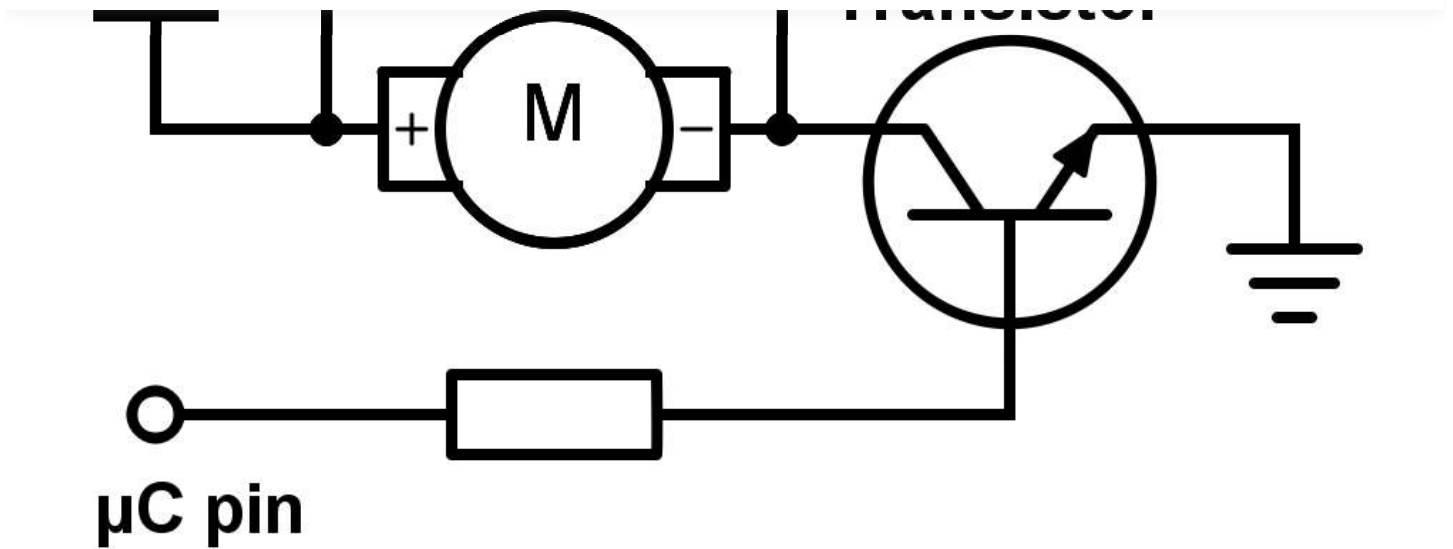


Es muy importante colocar el **diodo en paralelo** al motor, para proteger el mosfet y nuestro arduino de las corrientes inversas generados por la bobina del motor.

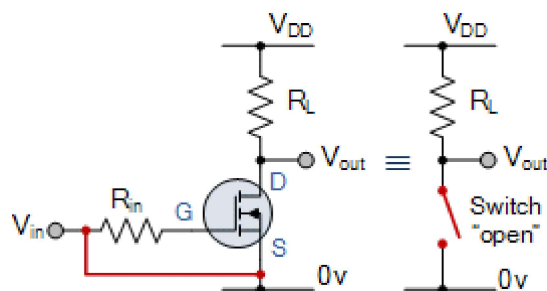


Este sitio web usa Cookies de terceros y propios, si continúas navegando las aceptas. [Aceptar](#) [Rechazar](#)

[Leer Más](#)



en este ejemplo estamos usando el MOSFET IRFZ44N como un Switch, dado que es un dispositivo capaz de mover grandes potencias. La idea básica es que la señal PWM Swichee a alta velocidad el mosfet y de esa forma podremos regular la velocidad del motor DC con Arduino.



Mosfet como Switch

En la implementación práctica, usaremos una motobomba de 12v, por lo tanto debemos usar dos fuentes de alimentación. Una para el arduino y otra para la motobomba. Recordar conectar ambas tierras de las fuentes, para crear una única tierra común.

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

[Leer Más](#)



Bomba 12 V 130PSI

[Comprar en Aliexpress](#)



Bomba cc 12 V 70PSI 3.5L/min

[Comprar en Aliexpress](#)



Driver para Control de Motores BTS7960 43A de alta potencia

[Comprar en Aliexpress](#)

No products found.

Códigos



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

[Leer Más](#)

[sociallocker id="948"]

LED

```
1. #define LED 3 // PIN 3 as LED
2. const int POT = A0; //Potenciometro
3. void setup() {
4.     // put your setup code here, to run once:
5.     pinMode(LED,OUTPUT);
6. }
7. void loop() {
8.     // put your main code here, to run repeatedly:
9.     analogWrite(LED,analogRead(POT)/4);
10. }
```

Motobomba

```
1. #define BOMBA 3 // PIN 3 as PUMP
2. const int POT = A0; //Potenciometro
3.
4. void setup() {
5.     // put your setup code here, to run once:
6.     pinMode(BOMBA,OUTPUT);
7. }
8.
9. void loop() {
10.    // put your main code here, to run repeatedly:
11.    analogWrite(BOMBA,analogRead(POT)/4);
12. }
```

[/sociallocker]

Eso es todo por la entrada del día 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 ☕](#)



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

[Leer Más](#)

Entradas relacionadas



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



Bluetooth HC-05 / HC06



Balanza Electronica con HX711 y Arduino



Acelerómetro y Giroscópio



Comunicación Serial Arduino Simulink/Matlab



Motor DC con Encoder – Velocidad – Posición



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

[Leer Más](#)

NOMBRE *

CORREO ELECTRÓNICO *

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 (7)



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

[Leer Más](#)

es posible usar una sola señal de PWM para el mismo circuito?

JOSE PIÑA ,

[Responder](#)



Sí, es posible utilizar una sola señal de PWM en un circuito para controlar varios dispositivos. Sin embargo, el detalle clave aquí es que todos los dispositivos compartirán la misma señal PWM y, por lo tanto, estarán sujetos a la misma modulación de ancho de pulso. Esto significa que no podrás controlar los dispositivos de manera independiente.

Un ejemplo común de esto podría ser el uso de una sola señal PWM para controlar la luminosidad de múltiples LEDs. Si todos los LEDs están conectados a la misma señal PWM, todos tendrán la misma luminosidad. No podrás hacer que un LED sea más brillante que otro, ya que todos están siguiendo la misma señal PWM.

[Responder](#)



saludos los videos estan muy bien comprensibles, tengo una consulta si yo quisiera modificar el pwm a la vez la frecuencia con potenciómetro. un VR. para pwm, la otra VR. para la frecuencia. tener una salida variable de acuerdo a mi necesidad. jhonny ,



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

[Leer Más](#)



Mis felicitaciones, Usted es un gran maestro, explica muy bien. me interesa mucho este curso, Deseo que cada vez que haga un nuevo capitulo por favor me los informen.
Gracias.

JOEL NOREÑA SERNA ,

[Responder](#)



Gracias Joel, que bueno que te gusta el curso. Recuerda que puedes ver la Sergio C , página principal del curso de Arduino donde puedes ver todas las entradas junto con todos los videos que hemos publicado hasta la fecha.

[Click aqui para ver todo el curso.](#)

[Responder](#)



saludos
muy agradecido por hacer la electrtonica universal abierta y al alcance de todos

alejandro torrecillas ,



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

[Leer Más](#)

De nada Alejandro, que bueno que te há gustado. 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 web usa Cookies de terceros y propios, si continúas navegando las aceptas. [Aceptar](#) [Rechazar](#)

[Leer Más](#)

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.

The image shows a Simulink model titled 'ControleDTCGPC'. The model includes a clock block 't', reference inputs 'put_MV' and 'put_MD', and disturbance inputs 'Distúrbios medidos'. The core control is an 'MPC - GPC' block with 'Ref', 'MO', and 'MV' inputs. The output 'u' goes to a 'u_dev fcn' block, which outputs 'u1'. The disturbance path goes through 'MD dev fcn' to 'd1', which then goes to 'md_1'. Both 'u1' and 'd1' are inputs to a 'Proceso Multivariable' block (labeled 'mo_7'). The process outputs 'y1'. The outputs are then processed by a 'Normalización de las variables' block, which outputs several 'BrakePower' signals for different components: 'Compressor_Main_1.BrakePower', 'Compressor_Main_2.BrakePower', 'Compressor_Exportation_1.BrakePower', 'Compressor_Exportation_T1S2.BrakePower', 'CC_Termoless_1.BrakePower', 'CC_Termoless_2.BrakePower', and 'CCInjectionSec1.BrakePower'. A 'Data Inspector' block is also present.

Overlaid on the bottom half of the screenshot is a promotional banner for a 'Curso de SIMULINK®' with a 'descuento' (discount) tag. The banner features the Simulink logo, a hand pointing to the discount tag, and a pair of scissors.

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

[Leer Más](#)

CUPON VALIDO

SISTEMAS

CONTROL

Udemy

Temperature Control Lab
<https://apmonitor.com/heat.htm>

MICROCHIP

The image is a promotional banner for a Udemy course. At the top, it says 'CUPON VALIDO' (Valid Coupon). The main title 'SISTEMAS CONTROL' is written in large, bold letters, with 'SISTEMAS' in red and 'CONTROL' in blue. To the right of the title is a hand cursor icon pointing towards the text. Below the title, there are two images: on the left, a blue PCB labeled 'Temperature Control Lab' with a URL 'https://apmonitor.com/heat.htm'; on the right, a black Microchip PIC microcontroller chip. The Udemy logo is in the top right corner.

👉 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 continúas navegando las aceptas. [Aceptar](#) [Rechazar](#)

[Leer Más](#)



SÍGUEME EN

CANAL EN TELEGRAM

Únete a @ConAutEdu en Telegram



VIDEO DE LA SEMANA



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

[Leer Más](#)



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 999+

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



Sergio A. Castaño Giraldo - Brasil

YouTube 999+

SIGUE LA PÁGINA DE FACEBOOK

[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](#)

[Microcontroladores \(PIC\)](#)



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

[Leer Más](#)

ACCESO



[Invítame a un Café](#)



[Acerca de mi](#)



[Contacto](#)



[Politica de Privacidad](#)



[Politica de Cookies](#)

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

