



S/. 0,00

CATEGORÍAS

[All Blog News](#)[Tutorial Módulo Controlador de servos PCA9685 con Arduino](#)

ARTÍCULOS POPULARES

[Tutorial sensor de flujo de agua](#) abr 13, 2016[Configuración del módulo bluetooth HC-05 usando comandos AT](#) ago 08, 2015[Tutorial transmisor de celda de carga HX711, Balanza Digital](#) ago 23, 2015

ARTÍCULOS RECIENTES

[Arduino Day 2020 por Naylamp Mechatronics!](#) mar 06, 2020[Usando ESP8266 con el IDE de Arduino](#) oct 21, 2016[Tutorial Sensor de Distancia SHARP](#) sep 28, 2016

PROMOCIONES ESPECIALES



Case oficial para Raspberry Pi 3B
Case oficial para el Raspberry Pi 3B,...

S/. 25,00 ~~S/. 30,00~~

Todas las promociones especiales >

ETIQUETAS



sensor

Fuente

V-Slot

hexagonal

allen

screw

regulador

LED

LCD

robot

NOVEDADES



Impresora 3d Creality Ender-5 Plus

Impresora 3D de gran volumen de la marca Creality. Una evolución con...

S/. 2.550,00



Impresora 3d Creality Ender-5

Impresora 3D Open Source de la marca Creality. Reúne las ventajas de la...

S/. 1.750,00



Fuente de alimentación para Raspberry Pi 4 USB-C 5.1V/3A Oficial

Fuente conmutada de 5V y 2500mA, ideal para alimentar tu Raspberry Pi.

S/. 50,00



Raspberry Pi 4 B 2GB

Versión más moderna y potente de la computadora del tamaño de una...

S/. 250,00



Raspberry Pi 4 B 4GB

Versión más moderna y potente de la computadora del tamaño de una...

S/. 290,00



Teensy 4.0

La tarjeta de desarrollo más avanzada de la familia Teensy, hasta 22x...

S/. 150,00



All metal hotend - Micro Swiss clone

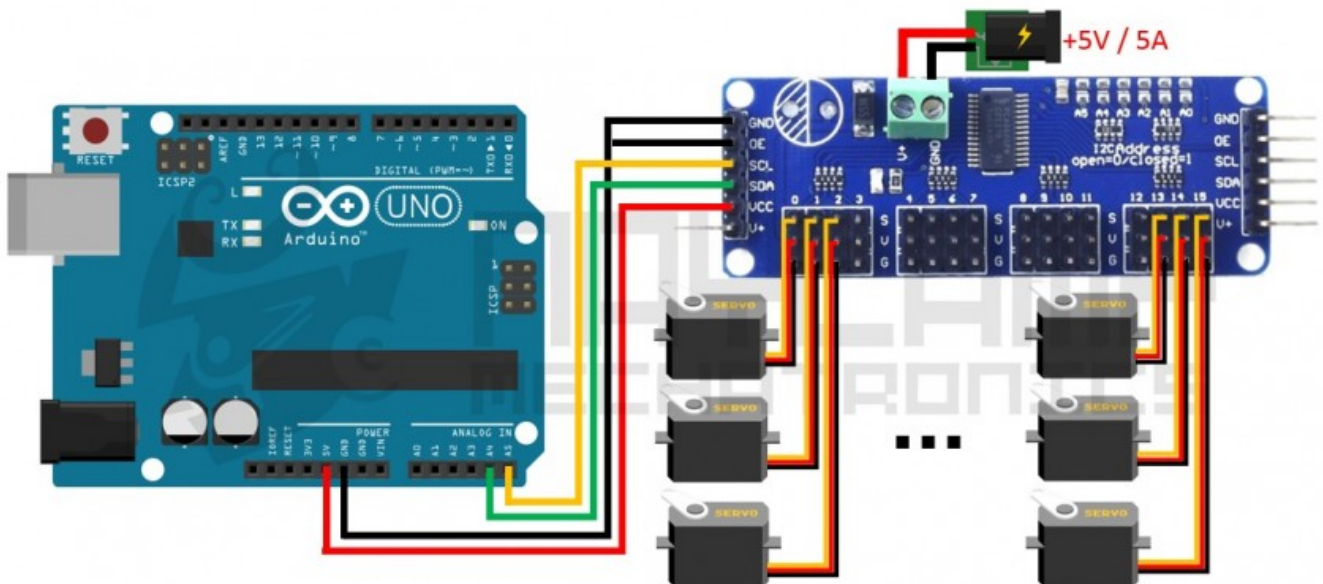
Hotend para impresión 3D de tipo Bowden.

S/. 190,00

Todos los nuevos productos >

Tutorial Módulo Controlador de servos PCA9685 con Arduino

Posted by Tutoriales 36 Comments

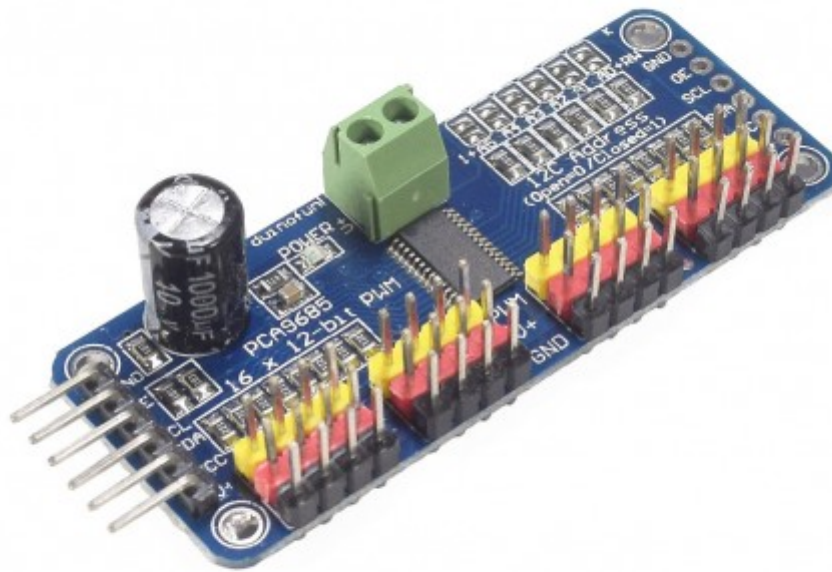


EL controlador PCA9685 fue diseñado para controlar leds por PWM, pero también nos permite controlar servos, ya que estos también se controlan por PWM, aplicación que actualmente es muy usada.

El Módulo Controlador de servos PCA9685 tiene la placa diseñada para el control de

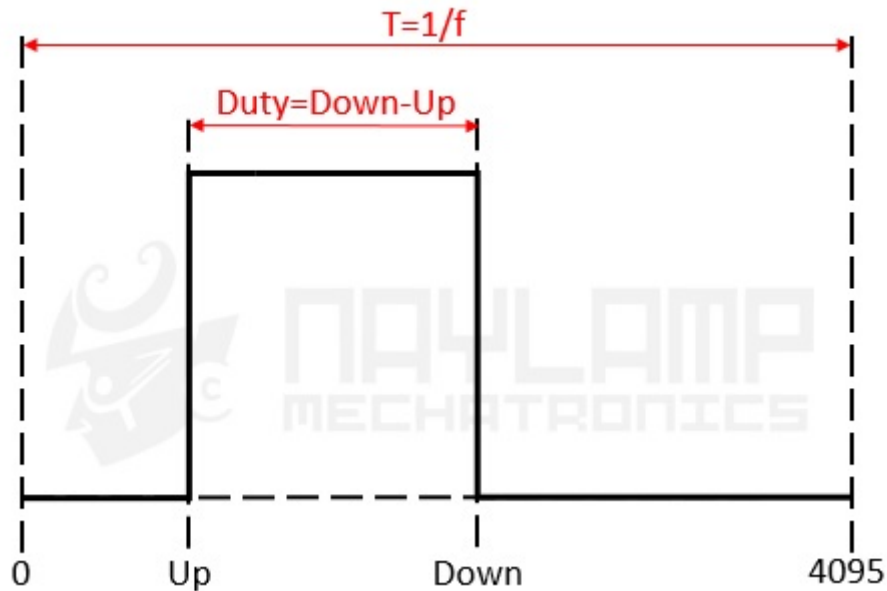
servos, tiene los pines en el orden correcto para simplemente conectar los servomotores, además una bornera para la alimentación de los servos y conectores para la alimentación de la parte lógica junto con los pines I2C para comunicarse con arduino.

Se puede establecer la dirección I2C soldando los puentes A0-A5 con esto podemos usar el mismo bus I2C para controlar más módulos PCA9685 u otros dispositivos I2C



EL PCA9685 nos permite controlar individualmente 16 salidas PWM con 12 bits de resolución y con frecuencia máxima de 1600Hz.

La salida PWM que envía el PCA9685 es de la siguiente forma:



Básicamente lo que se tiene que establecer es la frecuencia de la señal PWM, frecuencia que será la misma para las 16 salidas PWM. Para establecer el ciclo de trabajo (Duty) tenemos que manipular el flanco de subida (Up) y flanco de bajada (Down), esto se configura individualmente para cada salida PWM, La resolución del PWM es de 12 bits (de 0 a 4095)

Conexión Entre Arduino y Modulo PCA9685

Módulo PCA9685	Arduino Uno, Nano, Mini.	Arduino Mega , DUE	Arduino Leonardo
GND	GND	GND	GND
OE	GND	GND	GND
SCL	A5	21	3
SDA	A4	20	2
VCC	5V	5V	5V
+V	No conectado	No conectado	No conectado

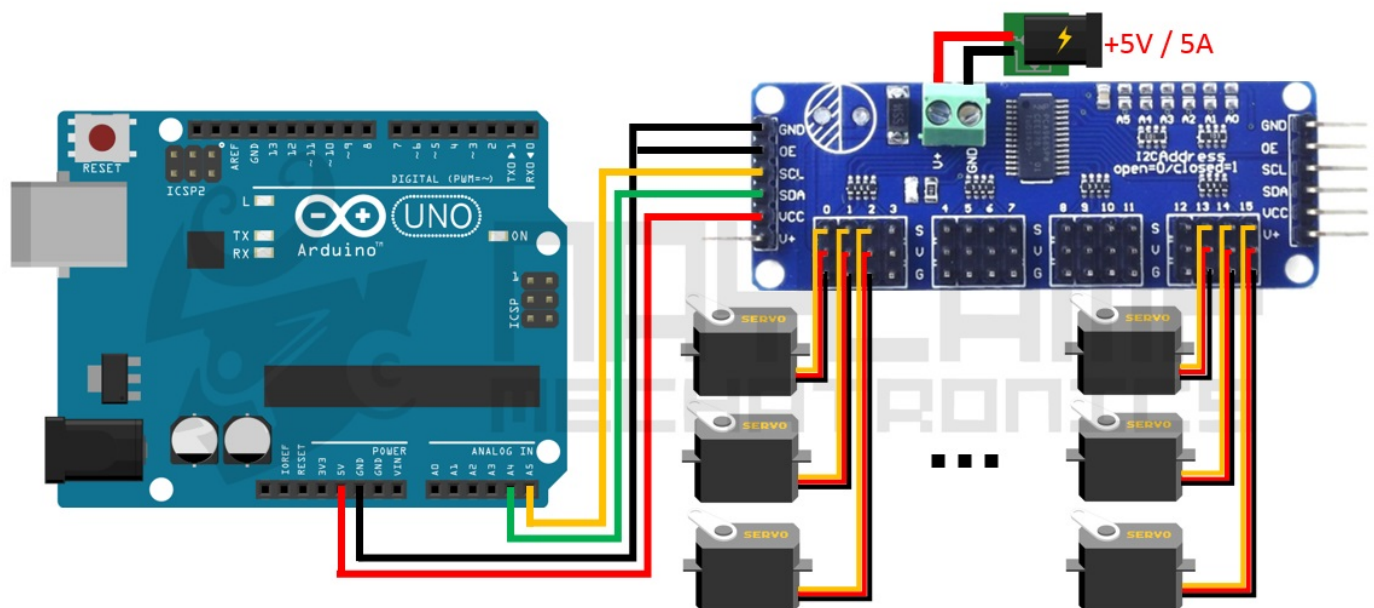
VCC es la alimentación para la parte lógica del módulo y V+ es la alimentación para los servomotores, entonces adicionalmente se tienen que conectar una fuente externa a V+ y

GND, fuente que debe ser del voltaje correspondiente a los servos. No usar los 5V que entrega Arduino para alimentar a los servos pues los 5V que entrega la placa Arduino es de poca corriente.

La mayoría de servos trabajan con voltajes de 4.5 y 6V nosotros usaremos una fuente de 5V / 5A.

La corriente mínima de la fuente externa depende del tipo de servomotores que se use y de la cantidad de servos que estarán conectados, Si bien la corriente no es un dato constante en el servomotor es mejor sobredimensionar la fuente para que trabaje correctamente, si la fuente no es muy estable o genera ruido, es necesario soldar un condensador en el espacio de la placa del Módulo con una valor de 1000uF o el equivalente a 100uF por cada servomotor.

Entonces, gráficamente las conexiones serían de la siguiente forma:



Librería PCA9685 para Arduino

Usaremos la librería de Adafruit el cual lo pueden descargar aquí:

<https://github.com/adafruit/Adafruit-PWM-Servo-Driver-Library>

Después de descargar necesitan importar la librería al IDE Arduino

Explicaremos como usar la librería:

Como cualquier librería inicialmente se incluyen las librerías correspondientes y se declara la variable u objeto:

```
#include <Wire.h>
#include <Adafruit_PWMServoDriver.h>

Adafruit_PWMServoDriver servos = Adafruit_PWMServoDriver(dirección);
```

La dirección si no han modificado en su placa del módulo PCA9685 es 0x40, si desean modificar la dirección, necesitan soldar los puentes de la placa, y queda establecida por: Dirección=0|1|A5|A4|A3|A2|A1|A0, por ejemplo si no sueldan ningún puente seria 01000000=0x40, si sueldan el segundo puente seria: 01000100=0x44

El siguiente paso es inicializar el objeto creado anteriormente.

```
servos.begin();
```

Posteriormente configurar la frecuencia del PWM, que será la misma para los 16 canales PWM

```
servos.setPWMFreq(60);
```

La frecuencia que usamos es de 60Hz que equivale una señal PWM con periodo 16,6ms que se encuentra dentro del rango del ciclo de trabajo de la mayoría de los servos

Para establecer el ancho de pulso de una salida PWM usamos la siguiente función

```
servos.setPWM(n_servo, up, down);
```

En este caso n_servo es el número de servo o salida que van a configurar (valor entre 0 y 15), up es el valor de cuentas en el que se producirá el flanco de subida, y down el valor para el flanco de bajada, ambos valores deben estar entre el valor de 0 y 4096 (12bits). El ancho de pulso será la resta de down-up.

Explicado el uso de la librería vamos a ver unos ejemplos:

Realizando un Sweep, barrido de 16 servos:

Este ejemplo hace una barrido de 0 a 180° y después de 180° a 0°, el barrido se hace a todos los servos por igual.

Antes de empezar necesitamos saber el ancho de pulso (duty) para la posición 0° y el correspondiente para 180°

Trabajaremos con 172 (0.7ms) para la posición 0° y 565(2.3ms) para la posición 180°

Para otros valores de los extremos del servo pueden calcularlo con la siguiente formula:

$$nCuentas = \text{Pulso_ms} * (\text{frecuencia_Hz} / 1000) * 4096$$

Generalmente se acostumbra aumentar o disminuyen los valores mínimo (0°) y máximo (180) hasta conseguir los valores para el rango correcto,

El sketch es el siguiente:

```
#include <Wire.h>
#include <Adafruit_PWMServoDriver.h>

Adafruit_PWMServoDriver servos = Adafruit_PWMServoDriver(0x40);
```



```
unsigned int pos0=172; // ancho de pulso en cuentas para pocicion 0°
unsigned int pos180=565; // ancho de pulso en cuentas para la pocicion 180°

void setup() {
  servos.begin();
  servos.setPWMFreq(60); //Frecuecia PWM de 60Hz o T=16,66ms
}

void loop() {

  for (int duty = pos0; duty < pos180; duty=duty+10) {
    for(int n=0;n<16;n++)
    {
      servos.setPWM(n,0,duty);
    }
  }
  delay(1000);
  for (int duty = pos180; duty > pos0; duty=duty-10) {
    for(int n=0;n<16;n++)
    {
      servos.setPWM(n,0,duty);
    }
  }
  delay(1000);
}
```

Moviendo los servomotores en posiciones diferentes.

Para la mayoría de aplicaciones los servomotores van a estar en diferente posición, para este caso es mejor hacer una función que nos calcule y envíe el valor en cuentas al servo y manipular el servo desde el void loop() con con valores de 0 y 180.

La función seria la siguiente:

```
void setServo(uint8_t n_servo, int angulo) {
  int duty;
  duty=map(angulo,0,180,pos0, pos180);
  servos.setPWM(n_servo, 0, duty);
}
```

Con esta función podemos enviar a cualquier servomotor el valor del ángulo al que queremos que se ubique.

El sketch que se muestra a continuación mueve los servomotores cada segundo a posiciones establecidas:

```
#include <Wire.h>
#include <Adafruit_PWMServoDriver.h>

Adafruit_PWMServoDriver servos = Adafruit_PWMServoDriver(0x40);

unsigned int pos0=172; // ancho de pulso en cuentas para pocicion 0°
unsigned int pos180=565; // ancho de pulso en cuentas para la pocicion 180°

void setup() {
```



```
servos.begin();
servos.setPWMFreq(60); //Frecuencia PWM de 60Hz o T=16,66ms
}

void setServo(uint8_t n_servo, int angulo) {
    int duty;
    duty=map(angulo,0,180,pos0, pos180);
    servos.setPWM(n_servo, 0, duty);
}

void loop() {

    setServo(0,30);
    setServo(2,90);
    setServo(4,180);
    setServo(6,120);
    setServo(8,0);
    setServo(10,30);
    setServo(12,90);
    setServo(14,170);
    delay(1000);
    setServo(1,30);
    setServo(3,90);
    setServo(5,180);
    setServo(7,120);
    setServo(9,30);
    setServo(11,90);
    setServo(13,180);
    setServo(15,120);
    delay(1000);
    setServo(0,120);
    setServo(2,180);
    setServo(4,90);
    setServo(6,60);
    setServo(8,45);
    setServo(10,160);
    setServo(12,170);
    setServo(14,30);
    delay(1000);
    setServo(1,120);
    setServo(3,0);
    setServo(5,90);
    setServo(7,60);
    setServo(9,120);
    setServo(11,180);
    setServo(13,0);
    setServo(15,30);
    delay(1000);
}
```

Usando este ejemplo como base ya pueden realizar cualquier aplicación.

En el caso que usen diferentes tipos de servomotores con diferente rango del ancho de pulso; incluso siendo del mismo fabricante en sus modelos de servos los valores mínimo y máximo pueden variar, por ejemplo para algunos casos el ancho de pulso para la posición 0° puede ser 0.7ms, 1ms u otro valor, de igual forma para la posición 180°.

Para este caso se necesita tener en cuenta que si bien la señal PWM puede ser el mismo para todos los servos debemos de tener como referencia diferentes valores del rango del ancho de pulso. Esto lo hacemos haciendo las siguientes modificaciones en el código anterior:

Al declarar las posición 0 y 180, lo debemos hacer con valores independientes para los 16 servos:

```
unsigned int pos0[16]= {172,172,172,246,246,172,246,200,200,150,160,172,172,172,200,246};  
unsigned int pos180[16]={565,256,256,492,492,565,492,550,550,600,590,565,565,550,492};
```

Y en la función para mover el servo debemos de usar estos valores a la hora de convertir de ángulo a duty

```
void setServo(uint8_t n_servo, int angulo) {  
    int duty;  
    duty=map(angulo,0,180,pos0[n_servo], pos180[n_servo]);  
    servos.setPWM(n_servo, 0, duty);  
}
```

También pueden modificar el valor 180 en caso quieran trabajar con valores de 0 a 255, valores con los que muchos traban puesto que ocupa un byte en la comunicación.

Pueden adquirir los materiales usados en este tutorial en nuestra tienda

Módulo Controlador de servos PCA9685

Arduino Uno R3

Servomotores

Tags: PWM servomotor

f 25 t ?? p 1 g 0 in ?? + Compartir 3

36 Comments



marco garcia

dic 6, 2016

Que tal buen dia gracias por el ejemplo, y si quisiera controlar motores a pasos 28byj-48 se podra? saludos gracias

Reply



Manuel

dic 7, 2016

todo quieres, por que no te pones a investigar por tu cuenta?

[Reply](#)

luis

abr 10, 2017

jvieql2fñ2h6ñlk1ñwhkwkglkokfkpojfewiejowjgoijgiorgoiirgirkdp03e

[Reply](#)

Naylamp

dic 14, 2016

Hola Marco, Las salidas del PCA9685 son PWM y son solo de control, no tienen suficiente corriente como para mover directamente cargas grandes como un motor. Para el 28byj-48 es mejor usar el ULN2003, o el 2803, este último tienen 8 salidas. Si deseas controlar varios, podrías usar multiplexores o registros con el fin de usar pocos pines desde arduino.

[Reply](#)

Pedro

dic 9, 2016

Conecté todo como se indicó en una Arduino Mega y Uno (oficiales), pero al momento de cargar los ejemplos citados aquí o los ejemplos de las librerías no prenden los servos, solo reaccionan los LEDs y el puerto serial del IDE Arduino. Los servos reaccionan al conectarlos, pero no realizan la tarea del LOOP. Ayuda.

[Reply](#)

Naylamp

dic 14, 2016

Hola pedro, Podría ser tu fuente o ruido, para descartar eso, solo prueba con un solo servo, si igual no funciona, podría ser que tu módulo PCA9685 tenga otra dirección I2C, si trabajas con el mega no olvides que los pines del I2C son el 20 y 21.

[Reply](#)

jose manuel

feb 3, 2017

Buenas he copiado y he pegado el programa, añadiendo un serial begin para ver como funciona ya que el servo no se movía y cuando llega al comando servos.begin(); la placa deja de funcionar. lo mismo me pasa con los ejemplos predefinido de la librería.

[Reply](#)

Naylamp

feb 13, 2017

Hola José, intenta probar el ejemplo con el servo desconectado, posiblemente sea la fuente de alimentación del servo,

[Reply](#)**Julio**

may 6, 2017

Cómo se puede conectar dos módulos en un mismo Arduino, quiero controlar 24 servos

[Reply](#)**Naylamp**

may 14, 2017

Hola Julio, el segundo PCA9685 se conecta en el mismo bus, usa los mismo pines de conexión, pero antes en uno de ellos tienes que cambiar su dirección I2C, para modificar la dirección, necesitas soldar los puentes de la placa cuya dirección quedará definida por: Dirección=0|1|A5|A4|A3|A2|A1|A0. En el programa solo crea un nuevo objeto para el segundo PCA9685, por ejemplo:

```
Adafruit_PWMServoDriver servos2 = Adafruit_PWMServoDriver(dirección2);
```

[Reply](#)**andres**

may 19, 2017

hola al darle verificar este me marca este error que puedo hacer ayuda: Arduino:1.8.2 (Windows 10), Tarjeta:"Arduino/Genuino Uno" h:1: error: #include expects "FILENAME" or #include ^ h:2: error: #include expects "FILENAME" or #include ^ exit status 1 #include expects "FILENAME" or Este reporte podría tener más información con "Mostrar salida detallada durante la compilación" opción habilitada en Archivo -> Preferencias.

[Reply](#)**Naylamp**

may 28, 2017

Hola Andrés, verifica que estén bien escritas las dos primeras líneas, para descartar problemas de librería compila los ejemplo que vienen junto con la librería, si estos también tienen errores al verificar, instala nuevamente la librería, instálalo desde el IDE de arduino usando Añadir Librería .ZIP

[Reply](#)**antonio g**

jun 29, 2017

Hola, una pregunta. Se puede conseguir con esta placa, que los servos se muevan a las ordenes de un teclado. con confirmación alfanumérica del número de servo a mover, de posición de reposo a final de carrera, y vuelta ? Me explico, por ejemplo, pulsar el número uno en el teclado (o nueve, o 11, etc.), y al apretar el asterisco, que inicie el movimiento. Luego al repetir la pulsación del mismo número pero al confirmar con almohadilla, que el servo retorne a la posición inicial. Gracias de antemano.

[Reply](#)**Naylamp**

jul 16, 2017

Hola Antonio, la tarjeta del PCA9685 es solo para controlar los servomotores, el que se va a encargar de ejecutar el algoritmo para mover los motores es el arduino y es allí donde tienes que programar de acuerdo a tu necesidad, el PCA9685 va a funcionar independientemente si el arduino se comunicación serial con un celular o la PC o cualquier otro periférico.

[Reply](#)**Jose**

abr 23, 2019

Hola Antonio Conseguiste tu propósito de manejar ellos servís individualmente por el teclado? Estoy haciendo algo parecido con un teclado midi y estoy interesado en tu Program ;)

[Reply](#)**alan**

jul 6, 2017

a mi me paso lo siguiente hice un programa basado en el ejemplo y primero use una fuente de 2 ampers 5 v que noo podia alimentar los 12 servos de 15kg, luego use una fuente de computadora de 5.5v a 20A y se quemó un transistor cerca del condensador y del led mis dudas son: se podra reparar? y cuanto amperaje resiste maximo ? a y si puede usarse la placa sin ese transistor ? en lo personal ya no me funiona la placa pero el transistor solo se quemó cuando el programa corria aunque ya tubiera el amperaje conectado alguna idea?

[Reply](#)**Naylamp**

jul 16, 2017

Hola Alan, existen varias variaciones de esta placa, comprueba si en realidad es un transistor, en algunos modelos viene un diodo para proteger la palca en caso se conecte al revés la fuente. Este diodo por lo general está cerca a la bornera y del condensador, y por el tamaño es de poca corriente, en tu caso seguramente se quemó porque 12 servos de 15Kg excedieron la corriente que el diodo soporta, Verifica y si es un diodo solo puentea con un cable el diodo, algunos modelos de este módulo no traen ese diodo.

[Reply](#)**Andres**

ago 11, 2017

Hola, muy buen trabajo y me funciona, pero si quisiera bajarle la velocidad de giro de los motores, ¿¿ como puedo hacerlo??

[Reply](#)

**Naylamp**

ago 22, 2017

Hola Andres, lamentablemente en estos servos no se pueden controlar la velocidad, salvo en los de rotación continua. Una forma de simular el cambio de velocidad es en no mover directamente de un punto a otro, sino en llevarlo de grado en grado haciendo pausas, similar al primer ejemplo pero con pequeñas pausas.

[Reply](#)**Víctor**

sep 22, 2017

Exelente trabajo muy bien explicado ahora como se conectaría con un Bluetooth

[Reply](#)**Lanthi**

oct 9, 2017

Y si quiero conectar dos o mas de estas placas como seria el codigo?

[Reply](#)**MATT**

mar 16, 2018

```
Adafruit_PWMServoDriver servos1 =  
Adafruit_PWMServoDriver(DIRECCION_PLACA1); //Por default 0x40  
Adafruit_PWMServoDriver servos2 =  
Adafruit_PWMServoDriver(DIRECCION_PLACA2); . . . Adafruit_PWMServoDriver  
servosN = Adafruit_PWMServoDriver(DIRECCION_PLACA_N); servos1.begin();  
servos2.begin(); . . servosN.begin();
```

[Reply](#)**viveba**

dic 29, 2017

Hola, buenas noches, muchas gracias por este tutorial. Tu pones el pin OE a masa (GND), ¿esto lo haces para evitar ruido? ¿o hay algún motivo más significativo? Desde ya, gracias por responder.

[Reply](#)**MATT**

mar 16, 2018

OE es para habilitar o deshabilitar las salidas. Conectado a GND quedan habilitadas por default, sino se puede conectar a una salida del microcontrolador y controlar por software.

[Reply](#)

**CHRISTIAN**

mar 17, 2018

Hola me gustaria saber por que cuando conecto mi fuente de alimentacion al modulo mis servos accionan si razon aparente no hace caso a las intrucciones que le pongo al arduino llevo todo el dia intentando saber por que pero no he encontrado nada... me podrian ayudar

[Reply](#)**Gonzalo**

abr 8, 2018

Hola qué tal Me gustaría saber cómo podría aplicar este ejemplo para controlar leds. Dar un tiempo de encendido a cada salida. Podrías apoyar con eso. De antemano, Gracias.

[Reply](#)**Gonzalo**

abr 8, 2018

Qué tal. Me gustaría aplicar este ejemplo para el control de leds. Podria apoyarme en eso. De antemano, Gracias.

[Reply](#)**Oscar Franco**

abr 13, 2018

Hola, estoy trabajando en un proyecto de feria de mi instituto. Es un robot humanoide de 17 servomotores, estoy utilizando Arduino UNO y el modulo PCA9685. El problema es el siguiente, cuando estoy programando todo va bien, las posiciones responden bien, pero el problema viene cuando ya dejó de trabajar con el robot y desconectó y lo guardo. Cuando vuelvo a conectar todo para continuar con la programación de la secuencia que quiero lograr, pues las posiciones no cogen el ángulo en qué estaba la otra vez, y vuelvo y subo el programa a Arduino y sigue con esa posición diferente, es como si el ángulo 50 cambiara y no es el mismo que el de la otra vez. Con el robot lo que quiero lograr es una secuencia de movimientos para hacer una demostración de éste, ya sea que camine y salude, etc. El código que utilizo es este. Nota: es solo un ejemplo

```
#include Adafruit_PWMServoDriver servos = Adafruit_PWMServoDriver(0x40); unsigned int pos0=172; // ancho de pulso en cuentas para pocicion 0° unsigned int pos180=565; // ancho de pulso en cuentas para la pocicion 180° void setup() { servos.begin(); servos.setPWMPFreq(60); //Frecuecia PWM de 60Hz o T=16,66ms } void setServo(uint8_t n_servo, int angulo) { int duty; duty=map(angulo,0,180,pos0, pos180); servos.setPWM(n_servo, 0, duty); } void loop() { setServo(0,30); setServo(2,90); setServo(4,180); setServo(6,120); setServo(8,0); setServo(10,30); setServo(12,90); setServo(14,170); delay(1000); setServo(1,30); setServo(3,90); setServo(5,180); setServo(7,120); setServo(9,30); setServo(11,90); setServo(13,180); setServo(15,120); delay(1000); setServo(0,120); setServo(2,180); setServo(4,90); setServo(6,60); setServo(8,45); setServo(10,160); setServo(12,170); setServo(14,30); delay(1000); setServo(1,120); setServo(3,0); setServo(5,90); setServo(7,60); setServo(9,120); setServo(11,180); setServo(13,0); setServo(15,30); delay(1000); }
```

[Reply](#)**Ronaldo Fernay Hernandez Gomez**

mar 3, 2019

buenas noches como seria el codigo para poder modificar la velocidad si tengo servos de rotacion continua

[Reply](#)**Akram**

abr 3, 2019

Me quedo loco a leer, que pena que todo esto me suene a chino, pero le estot poniendo interes. Asi que Gracias a ti y a los que estan comentando

[Reply](#)**Mariana**

may 6, 2019

Hola esto me marca como error #include

[Reply](#)**Mariano**

may 31, 2019

hola, me gustaría usar placa arduino uno + pca9685 para accionar 8 servos con pulsadores, alguien podría facilitarme el sketch así como un esquema de las conexiones? perdonad si pido mucho pero es que no se nada de arduino pero me parece muy interesante para mover los desvíos un saludo y gracias por adelantado

[Reply](#)**Anonim0w0**

ago 15, 2019

Che capo, puse los programas del ejemplo, conecté todo como decía ahí, y lo probé conectando un solo servo al controlador pero el servo no se movía a pesar de que la placa recibía voltaje/corriente. **AXUDAME**

[Reply](#)**gustavo**

ago 18, 2019

Disculpa la.placa puede con 16 servos mg995? y me gustaria saber si una bateria lipo con 25c (es decir que suministra 25 veces mas la corriente establecida) es capaz de alimentarlos

[Reply](#)**José Pablo**

feb 27, 2020

El programa me funcionó a la perfección, pero quiero saber si es posible cambiar los pines de SDA y SCL a otros pines analógicos debido a que también deseo conectar un acelerometro MPU6050 y también usa esos mismos pines A4 y A5. ¿O no hay problema si uso los mismos pines para ambos módulos?

[Reply](#)

Leave a Reply

* **Name:**

* **E-mail:**

(Not Published)

Website:

(Site url withhttp://)

* **Comment:**



Type Code

Submit

Boletín

Introduzca su dirección de correo electróni



Información



Contacte con nosotros

Nosotros

Envíos y Entregas

Condiciones de uso

Tutoriales y Proyectos con Arduino

Mapa del sitio

Mi cuenta



Mis compras

Mis vales descuento


Mis direcciones

Mis datos personales


Mis vales

Información sobre la tienda



 Naylamp Mechatronics SAC., Almacén: Mz. E Lote 31 Urb. Ingeniería, Trujillo - Perú //// Horario de atención:

Lun. a Vie: 10am-7pm. Sab: 10am-6pm

 Llámanos ahora: 997646858

 Email: naylamp.mechatronics@gmail.com