

AZ-Delivery

Welcome!

Thank you very much for purchasing our AZ-Delivery PCA9685 16x Servo Driver. On the following pages, we will introduce you to how to use and setup this handy device.

Have fun!



Az-Delivery

The PCA9685 module is an I2C bus controlled 16 channel PWM controller module for Atmega328p and Raspberry Pi applications with servo motors. Each PWM servo output has a separate internal PWM controller with 12 bit resolution (4096 steps).

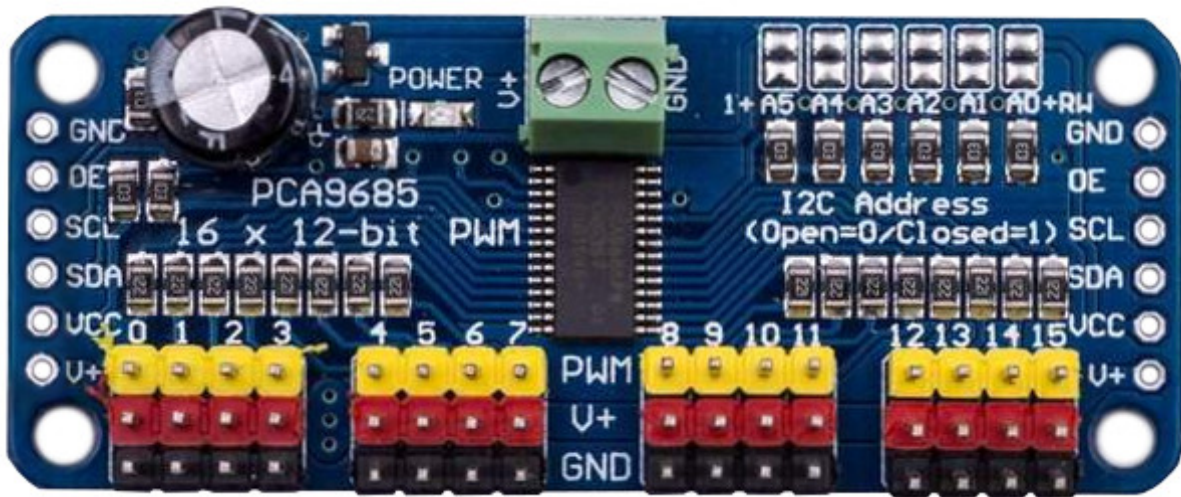
The PWM output driver can be programmed to be either a drain with 25mA current sink at 5V or a totem pole with 25mA sink and 10mA current source at 5V.

The module operates with a supply voltage range from 2.3V to 5.5V and the inputs and outputs are 5V tolerant.

The servo motors can be connected up to 25mA and 5V directly to the PWM outputs of the module or controlled with external drivers and a minimum number of discrete components with higher current or voltage.

The switch-on and switch-off times are independently programmable for each of the 16 channels. The power-on reset (POR) status of all 16 PWM output pins is LOW. Six hardware address pins theoretically allow up to 62 devices on the same bus. Which is: $62 \text{ boards} * 16 \text{ servos} = 992 \text{ servos}$ can be controlled on the same bus.

Az-Delivery



The module has a total of 6 control connections. These are occupied starting from the top left:

Pin	>	Description	Additional information:
1	>	GND	Ground
2	>	OE Input	Output Enable - LOW active (LOW by default)
3	>	SCL Input	I2C Bus - Clock line
4	>	SDA Input	I2C Bus - Data line
5	>	VCC	Power supply max. 5V (for PCA9685 chip)
6	>	V+	Power supply max. 6V (for servos)

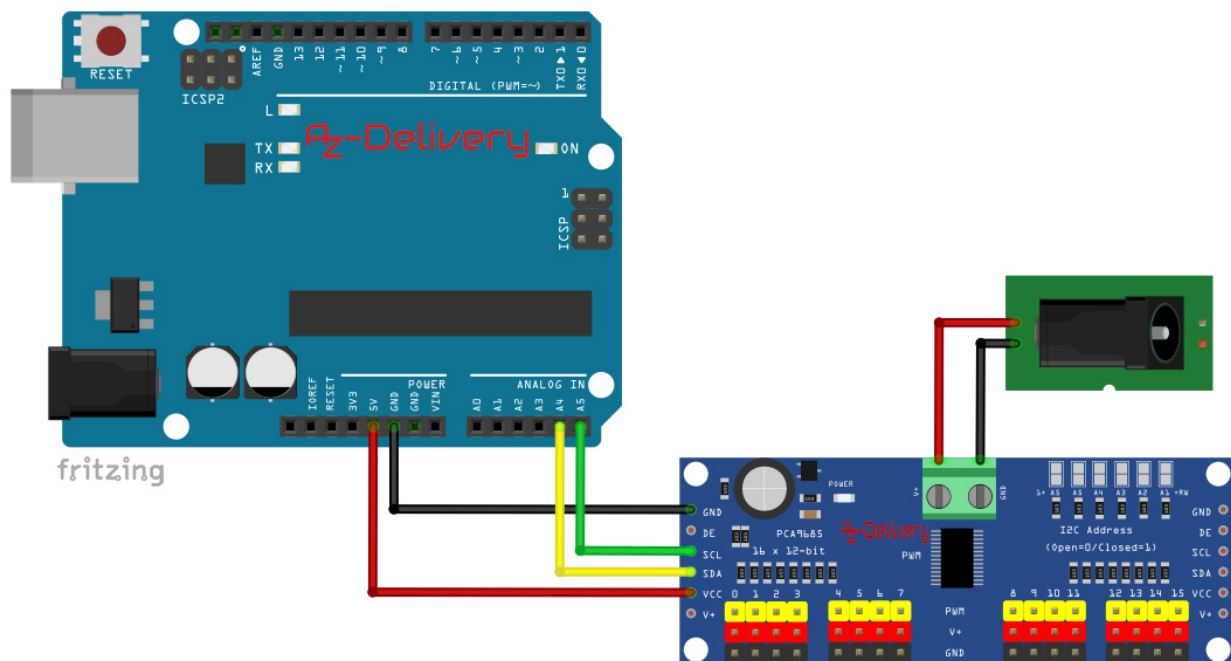
Pin 6 should remain unconnected. Instead use green screw terminal on the top of the board for power supply for servos, max. 6V.

Az-Delivery

Output Enable

The active LOW Output Enable input pin (OE) allows asynchronous control of the PWM outputs and can be used to set all outputs to a defined, programmable I2C bus logic state. The OE can also be used to externally "pulse width modulate" the outputs. This is useful when multiple devices need to be dimmed or flashed together using software control. By default, if left unconnected, it is in LOW state.

The basic circuit with the Atmega328p is as follows:



The pin connection is:

A4 > SDA

A5 > SCL

+5V > VCC

GND > GND

Yellow wire

Green wire

Red wire

Black wire



PWM - basic frequency

The PCA9685 has an adjustable PWM base frequency from 24Hz to 1526Hz, whereby the accuracy of the PWM base frequency is not very high, since the internal oscillator is used as clock generator. The PWM base frequency is determined using an internal programmable prescaler value written to register *0xFEh*. All outputs always work with the same PWM frequency, whereby the duty cycle can be set between 0% and 100%.

Here are a few Prescaler values to try out for yourself:

Prescaler value:	PWM base frequency:
0x03h	1,526 kHz
0x06h	1 kHz
0x0Ch	500 Hz
0x1Eh	200 Hz
0x7Ah	50 Hz
0xFFh	24 Hz

The prescaler value can be adjusted in the code in the *driver.setPWMFrequency()* function. The PWM base frequency with which your servo motor works best, can either be taken from the servo datasheet or simply tried out for yourself.



Power supply of the module/output current limitation

On the left connector strip of the module there are two positive supply pins and one ground pin. The positive supply pins are described with Vcc and V+. Vcc is the supply for the PWM chip itself and not for the servos!

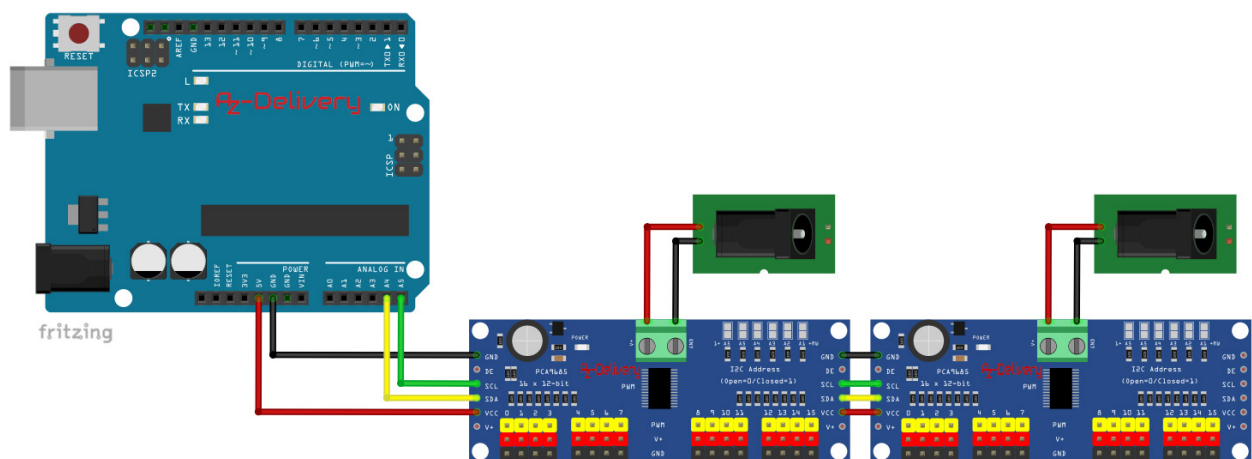
V+ should be left unconnected, but the power supply of the servos should be done via the green power supply terminal on the top of board, with power supply of the appropriate size.

Please note that the maximum supply voltage of the servos may be at 6V.

Each PWM output has a maximum positive current carrying capacity of 10mA at 5V.

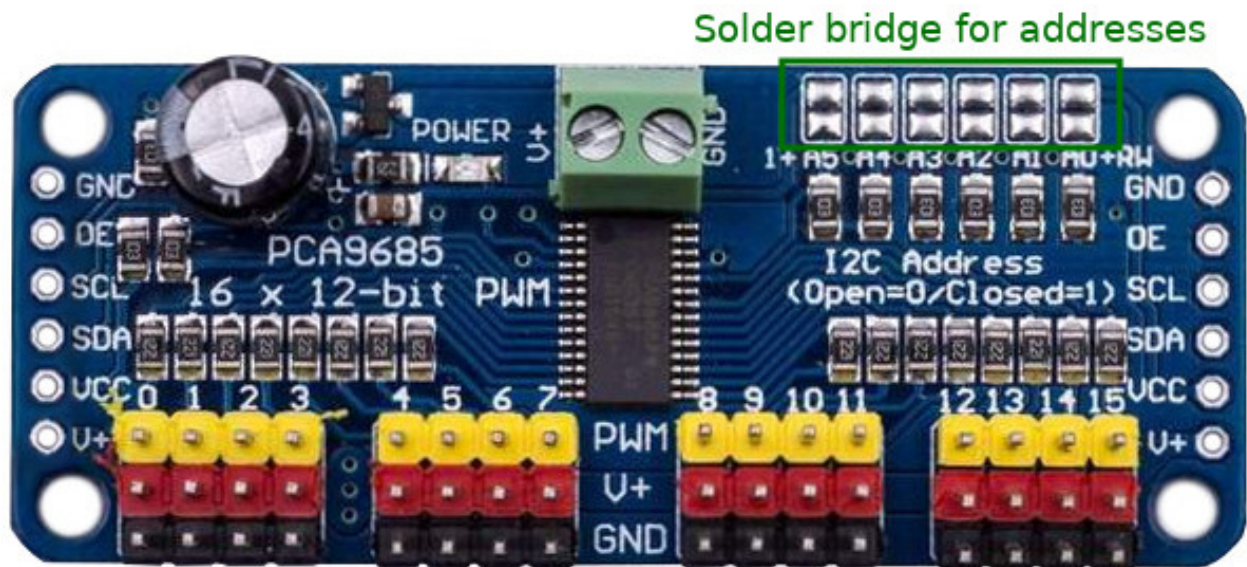
Addressing several modules

If 16 outputs are not sufficient for the one project, further modules can be connected in series to the I2C bus. The wiring required for this is shown below:



It should be noted that the supply of the individual servos can no longer be taken from Atmega328p here at the latest, but that it is imperative that the supply voltage is externally supplied via the green terminals. One external power supply per module must be supplied.

Az-Delivery



The addressing solder bridges A0 to A5 on the module should then be connected differently for each module according to the binary code counting upwards:

PWM Module:	Soldering bridge for:	I2C address of the module:
0	No connection	0x40
1	A0	0x41
2	A1	0x42
3	A0 and A1	0x43
4	A2	0x44
5	A2 and A1	0x45

The control register addresses start anew with each module. This means that the basic control of channels 0 - 15 can remain the same, only the I2C address of the module must be changed. In the code this address is set by `driver.init(B000000);`.

Az-Delivery

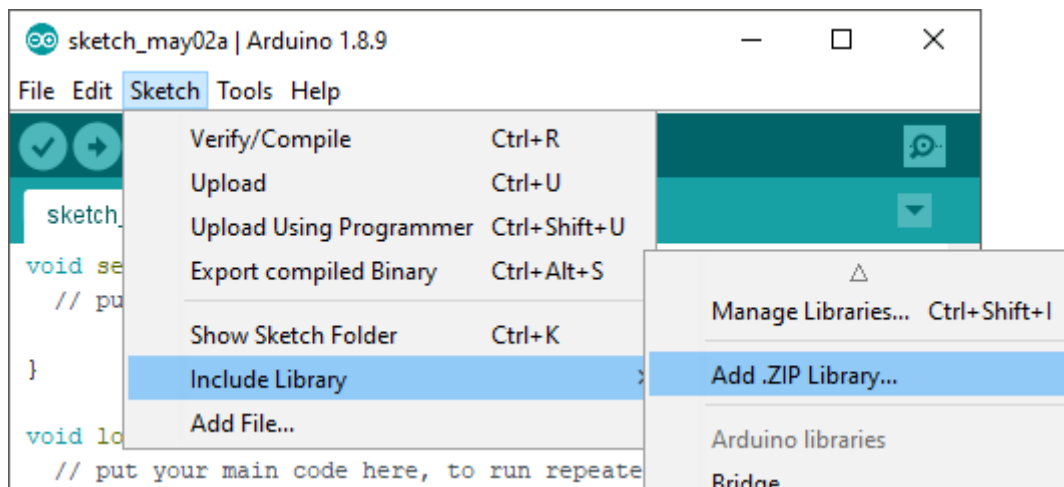
The library and code

We use this library: <https://github.com/NachtRaveVL/PCA9685-Arduino>

To include it in your Arduino IDE, download it from the link above. The file you download will be .zip, so in your Arduino IDE go to:

Sketch > Include Library > Add .ZIP Library

and find and add .zip file you've just downloaded.



Az-Delivery

We use:

```
PCA9685_ServoEvaluator pwmServo(102, 470); // (-90deg, +90deg)
```

to define end positions of servo.

And there is another way to do it:

```
PCA9685_ServoEvaluator pwmServo2(102, 310, 505); // (0deg, 90deg, 180deg)
```

With this:

```
Wire.setClock(400000);
```

we define I2C baud rate at 400kHz.

With this:

```
driver.init(B000000);
```

we set I2C address of our PCA9685 to 0x40.

With this:

```
driver.setPWMPFrequency(50);
```

we set PWM frequency at 50Hz.

And to move servo shaft we use:

```
driver.setChannelPWM(0, pwmServo.pwmForAngle(-90));
```

But be careful to use angle between -90 and 90 if you used first method to define end positions for servo, or 0 and 180 for second method.

Az-Delivery

```
#include <Wire.h>
#include "PCA9685.h"
PCA9685 driver;
// PCA9685 outputs = 12-bit = 4096 steps
// 2.5% of 20ms = 0.5ms ; 12.5% of 20ms = 2.5ms
// 2.5% of 4096 = 102 steps; 12.5% of 4096 = 512 steps
PCA9685_ServoEvaluator pwmServo(102, 470); // (-90deg, +90deg)
// Second Servo
// PCA9685_ServoEvaluator pwmServo2(102, 310, 505); // (0deg, 90deg, 180deg)
void setup() {
    Wire.begin();           // Wire must be started first
    Wire.setClock(400000); // Supported baud rates are 100kHz, 400kHz, and 1000kHz
    driver.resetDevices(); // Software resets all PCA9685 devices on Wire line
    driver.init(B0000000);  // Address pins A5-A0 set to B0000000
    driver.setPWMPFrequency(50); // Set frequency to 50Hz
}

void loop() {
    driver.setChannelPWM(0, pwmServo.pwmForAngle(-90));
    delay(1000);
    driver.setChannelPWM(0, pwmServo.pwmForAngle(0));
    delay(1000);
    driver.setChannelPWM(0, pwmServo.pwmForAngle(90));
    delay(1000);
    driver.setChannelPWM(0, pwmServo.pwmForAngle(0));
    delay(1000);
}
```

You've done it, you can now use and program your module for your projects.



Now it is time to learn and make the Projects on your own. You can do that with the help of many example scripts and other tutorials, which you can find on the internet.

If you are looking for the high quality microelectronics and accessories, AZ-Delivery Vertriebs GmbH is the right company to get them from. You will be provided with numerous application examples, full installation guides, eBooks, libraries and assistance from our technical experts.

<https://az-delivery.de>

Have Fun!

Impressum

<https://az-delivery.de/pages/about-us>