# D1 Mini 16-channel Servo Shield Documentation

*Release 1.0*

**Radomir Dopieralski**

**Apr 22, 2020**

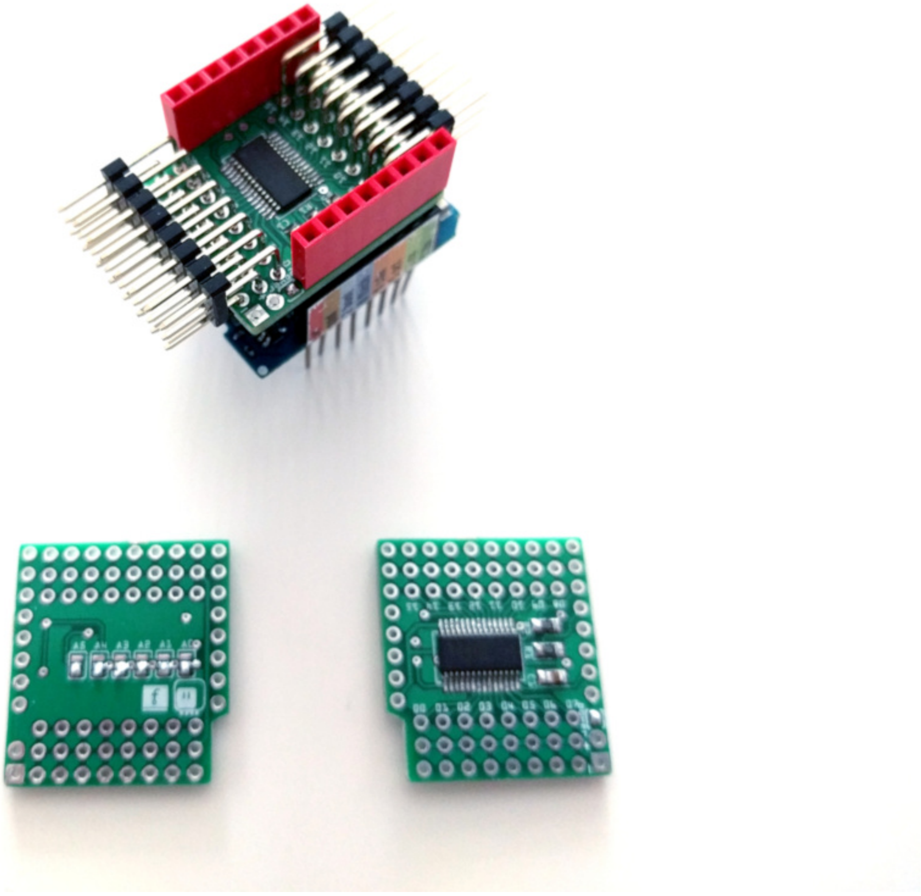
# Contents

This is an instruction manual for the 16-channel servo shield for the D1 Mini series of microcontroller boards. This shield lets you control up to 16 hobby servos, or anything else that takes a similar PWM signal. This board is designed to plug on top or bottom of the small ESP8266-based microcontroller boards produces by WeMos, such as D1 Mini, D1 Mini Pro and D1 Mini Lite.

Contents:

# CHAPTER 1

## Overview



This shield is a relatively simple breakout board for the PCA9685 chip. Apart from that chip, it only contains the bare minimum of components necessary to make it work, and all the necessary connections. The goal is to make it as convenient as possible to connect hobby servos to the ESP8266.

The board has the standard size and shape of D1 Mini shields, with the servo sockets broken out as angled headers, protruding on the sides, so that the boards can be easily stacked with each other or with other shields. Up to 64 such shields can be used with a single microcontroller, provided that they have different addresses set.

The chip takes care of continuously generating the signal that is expected by the hobby servos, freeing the microcontroller from this task. The microcontroller only needs to get involved when you want to change the position of any of the servos – it sends a command then, and from that moment the shield will generate the signal for the new position for that servo.

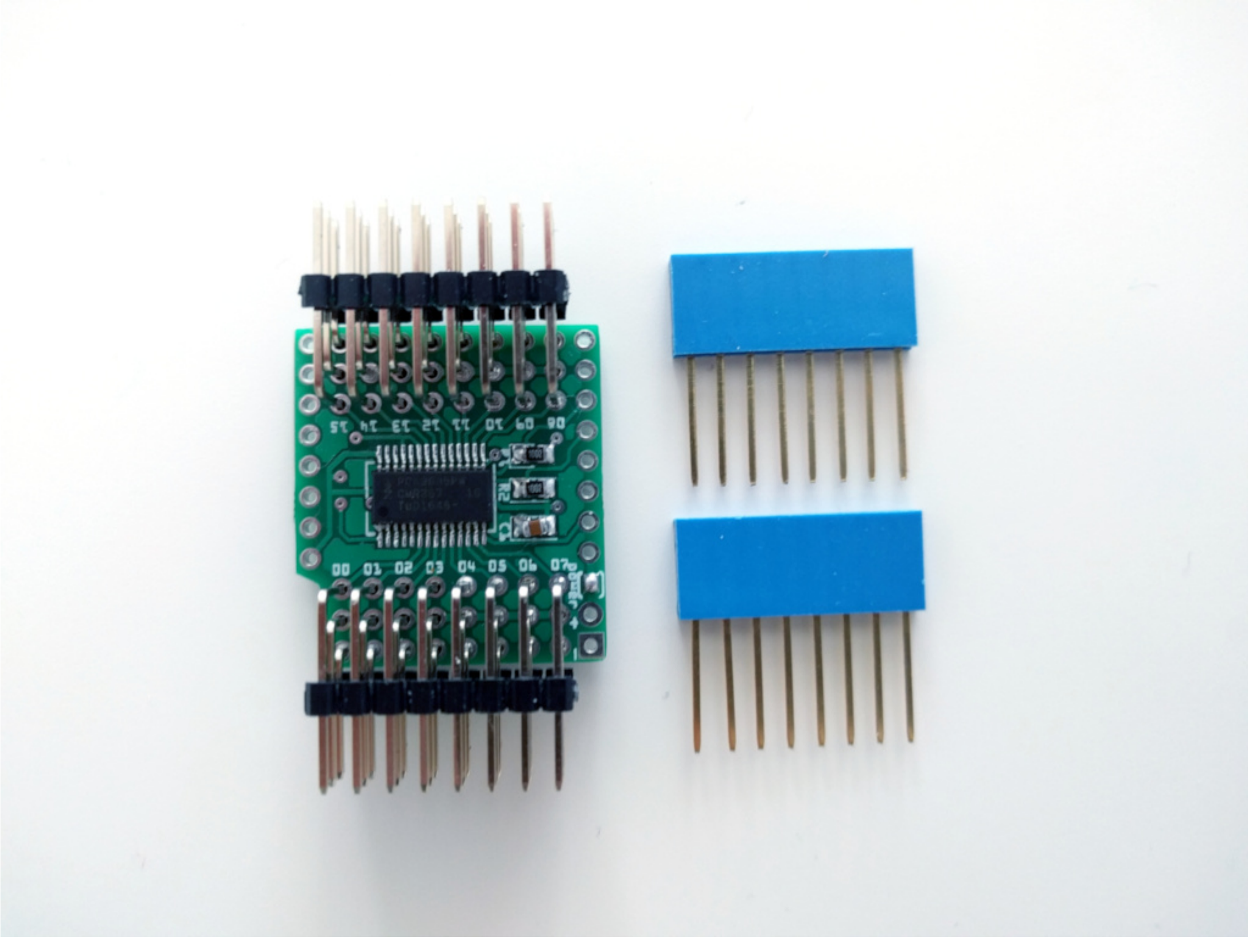
This shield is 100% compatible with other servo controllers using PCA9685 chip, and can be used as a drop-in replacement for them. All the available libraries and examples should work exactly the same.

# CHAPTER 2

## Assembly

To protect the boards during transport, they ship only partially assembled. Before you can use them, you need to solder the stacking headers on them.

**Warning:** Soldering the headers is a necessary step, and can't be skipped. Just inserting the headers without soldering will not provide a robust electrical connection that is required for this shield to work properly.

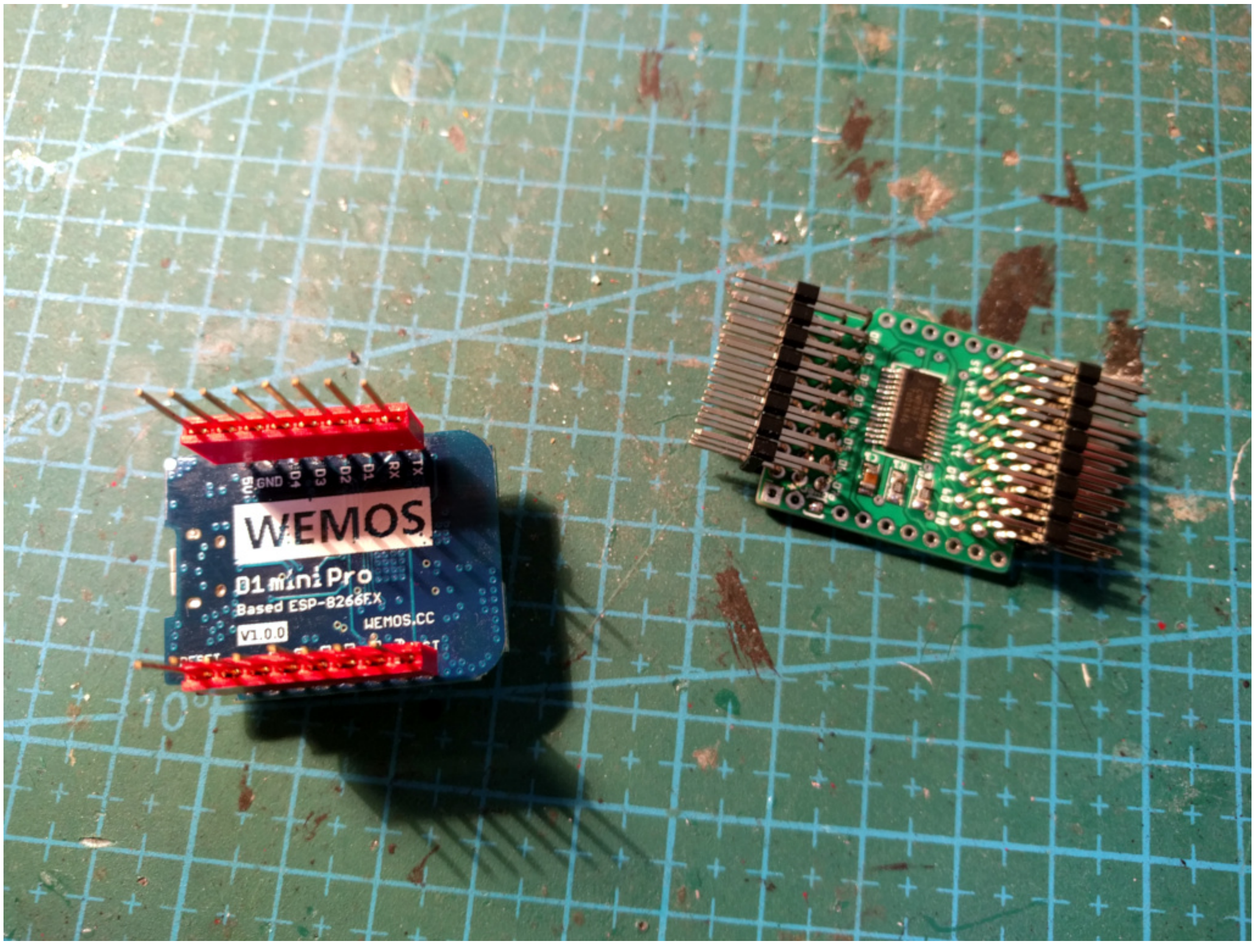


## 2.1 Stacking Headers

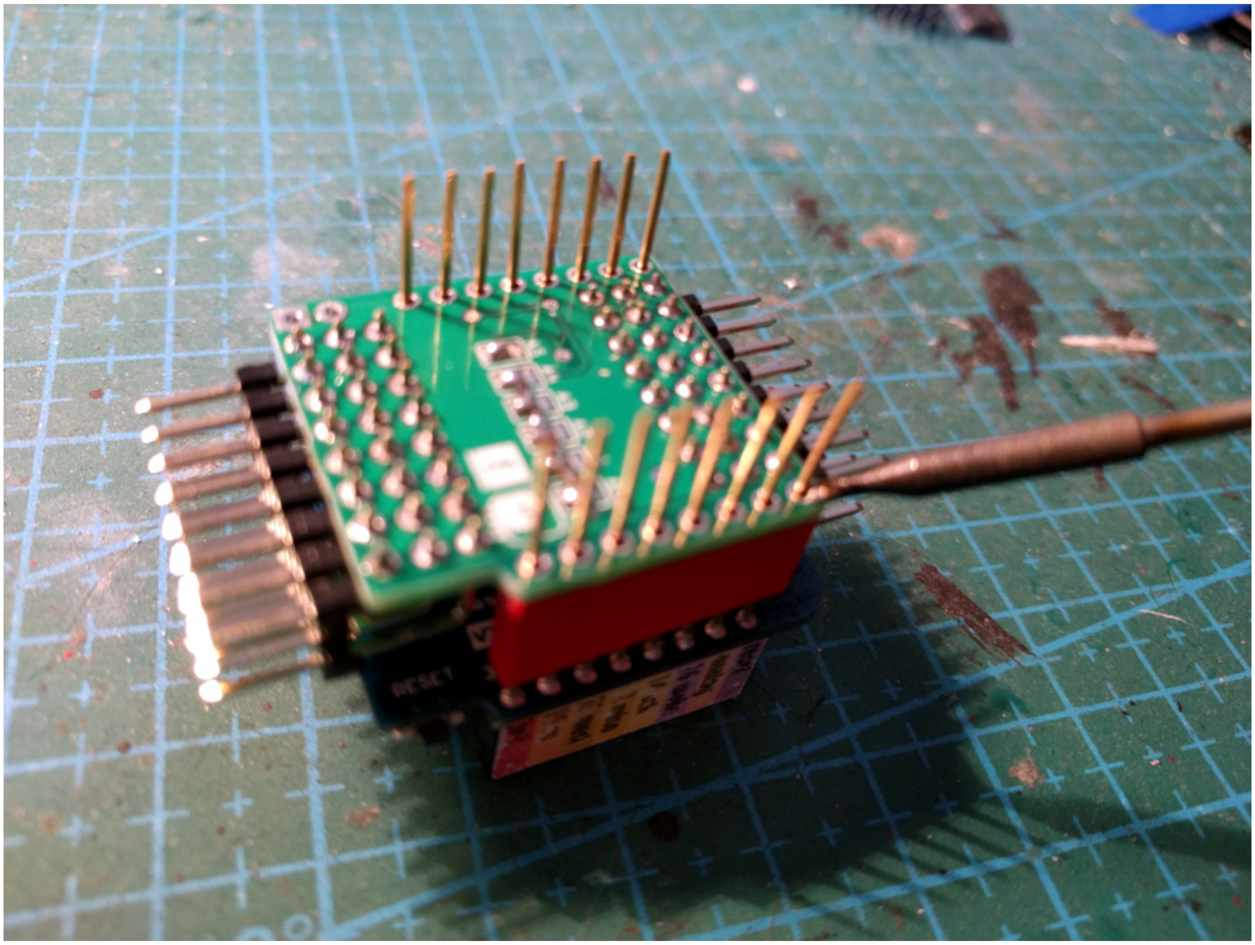
You should have received a pair of 8-pin stacking headers with your board. They let you put this shield either on top or on the bottom of the D1 Mini, and stack more shields on top (or bottom) of that.

**Note:** If you want to, you can instead use normal male or female headers, or even connect the shield to the microcontroller board by soldering the wires or male headers to both boards directly. In either case, the procedure is similar.

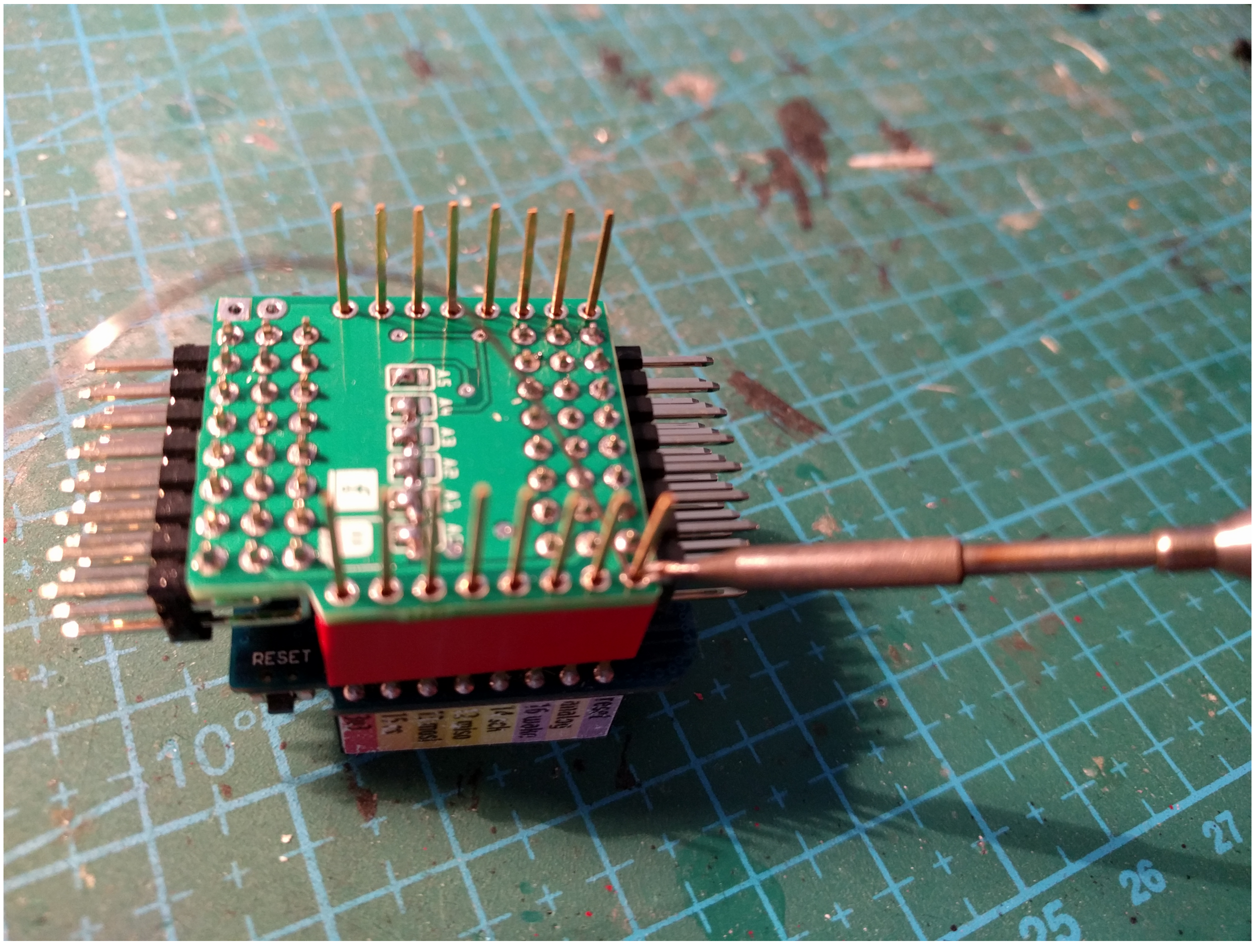
Start by putting the headers on your board, or any other already assembled shield. This will hold them in place straight and make it easier to work.



Then place the shield on the headers, making sure that every pin goes into its hole. You might need to use tweezers to help you insert all the pins.



Next, connect your soldering iron, wait for it to get hot, and put it on one of the pins, so that it touches both the pin and the board.

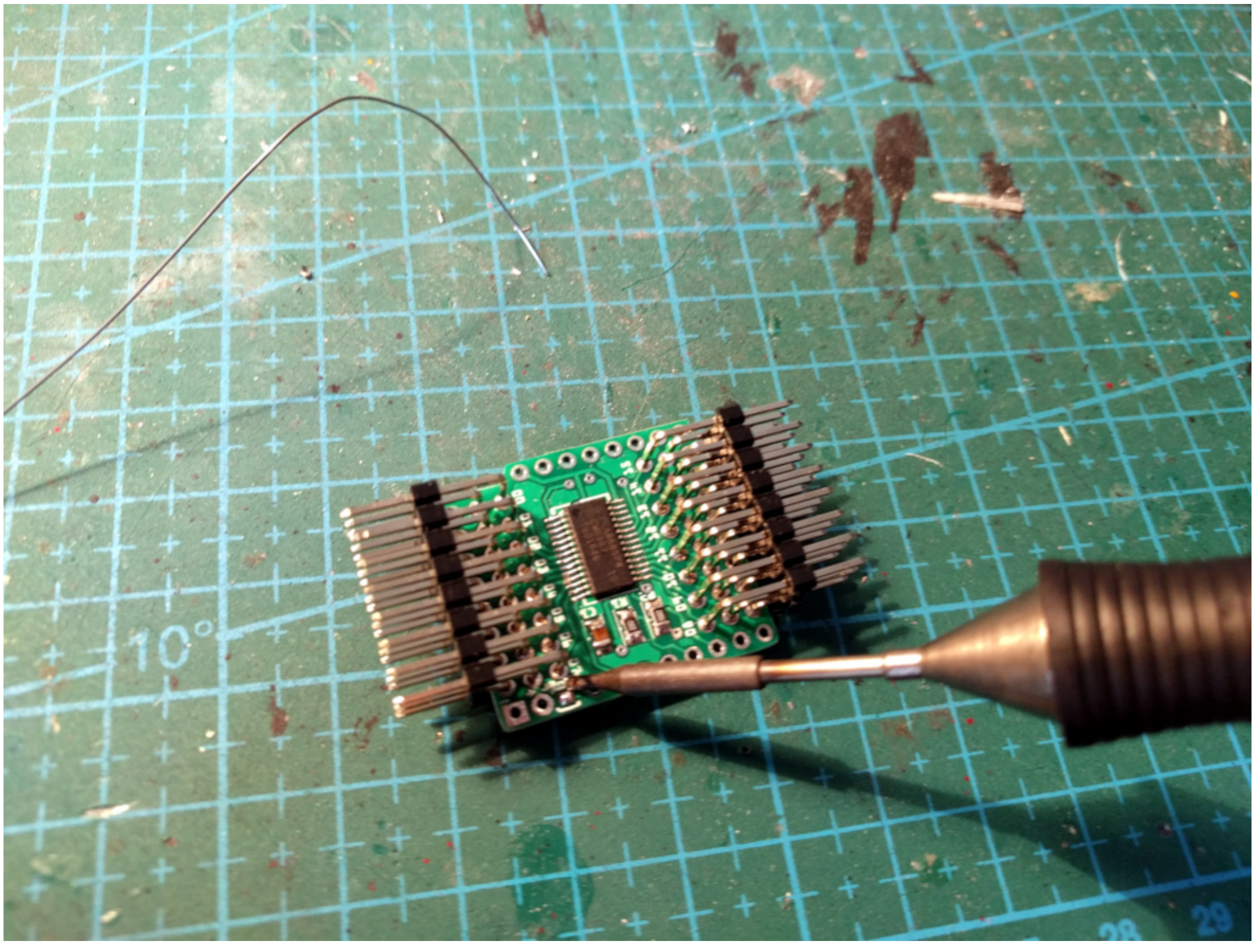


Wait a couple of seconds for the pin to get hot, and then touch the pin and the board with the solder wire. Let the solder flow into the hole around the pin. Then remove the solder wire and the soldering iron, and move to the next pin.

Repeat that for each of the 16 pins. Done.

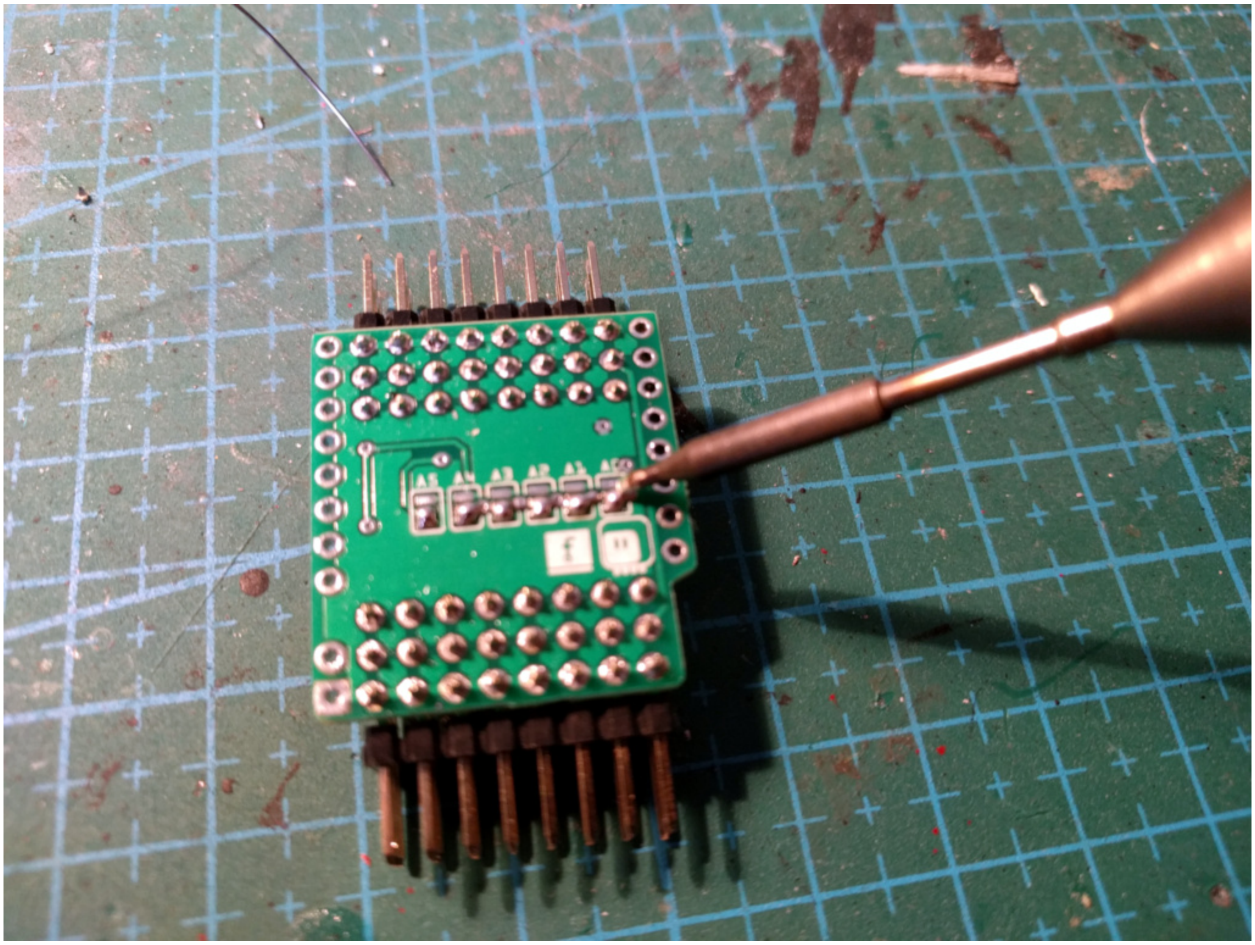
## 2.2 External Power Source

If you want to power your servos from a different power source than the microcontroller board, you need to remove the drop of solder from the jumper next to the power connector, and solder the connector or wires for the power to the two additional holes. The square hole is minus, the round one is plus.



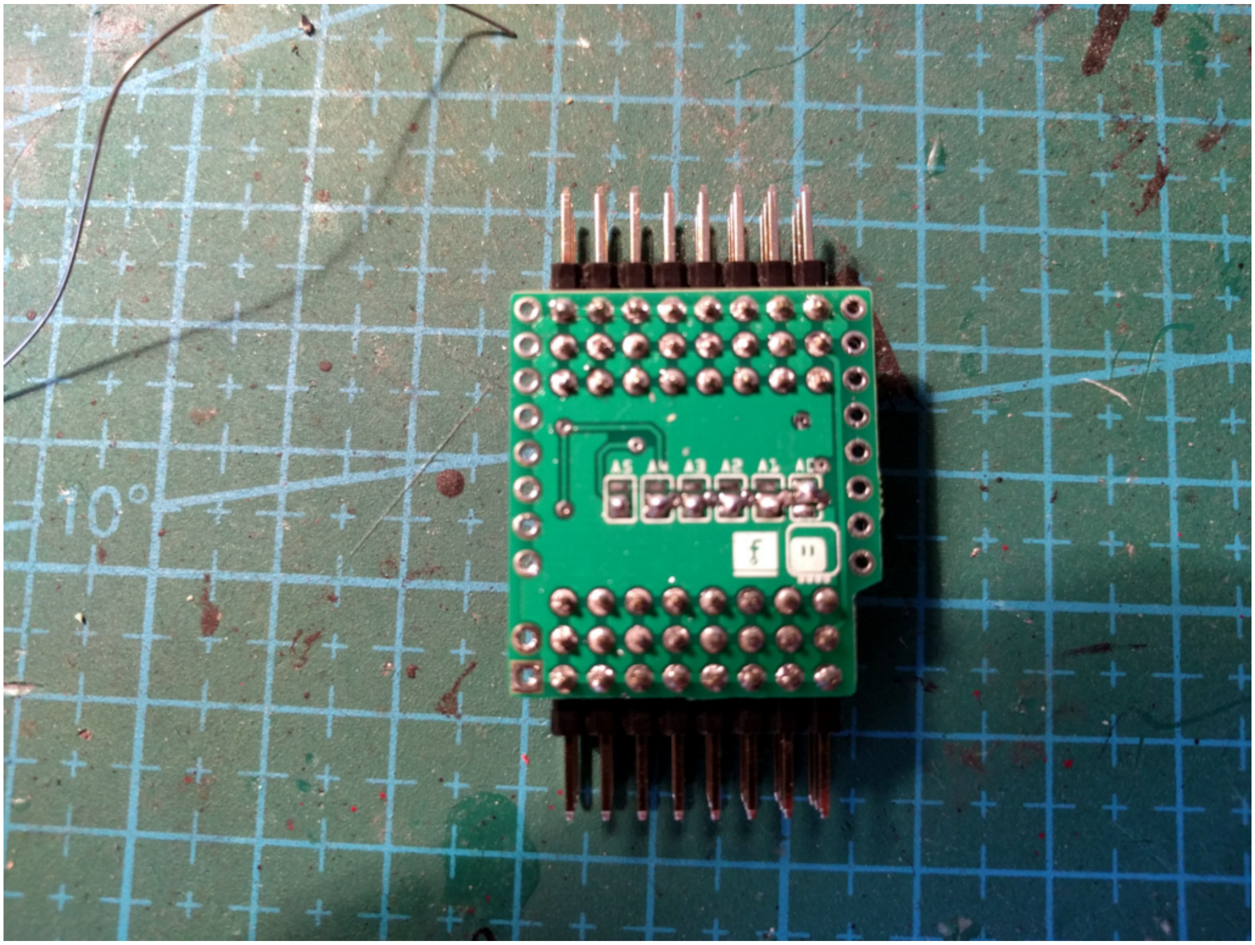
## 2.3 Address Selection Jumpers

On the back of the board, there are six jumpers, labeled A0 to A5, which let you change the default I$^2$C address of the board. By default, they are all set to 0, which sets the address to the value of 0x40. If you plan to stack several of those boards together on the same microcontroller board, you need to change their addresses, so that every one of them has a different one. There are 64 possible combinations.



**Warning:** Be careful to not connect a jumper both to the top and bottom pads, as that will result in an electric short, a lot of smoke and possibly permanently damaged shield, microcontroller board, battery or any other power source used (including your computer, if you powered it from USB).

It is recommended to check the connections with a multimeter before applying power to the device.

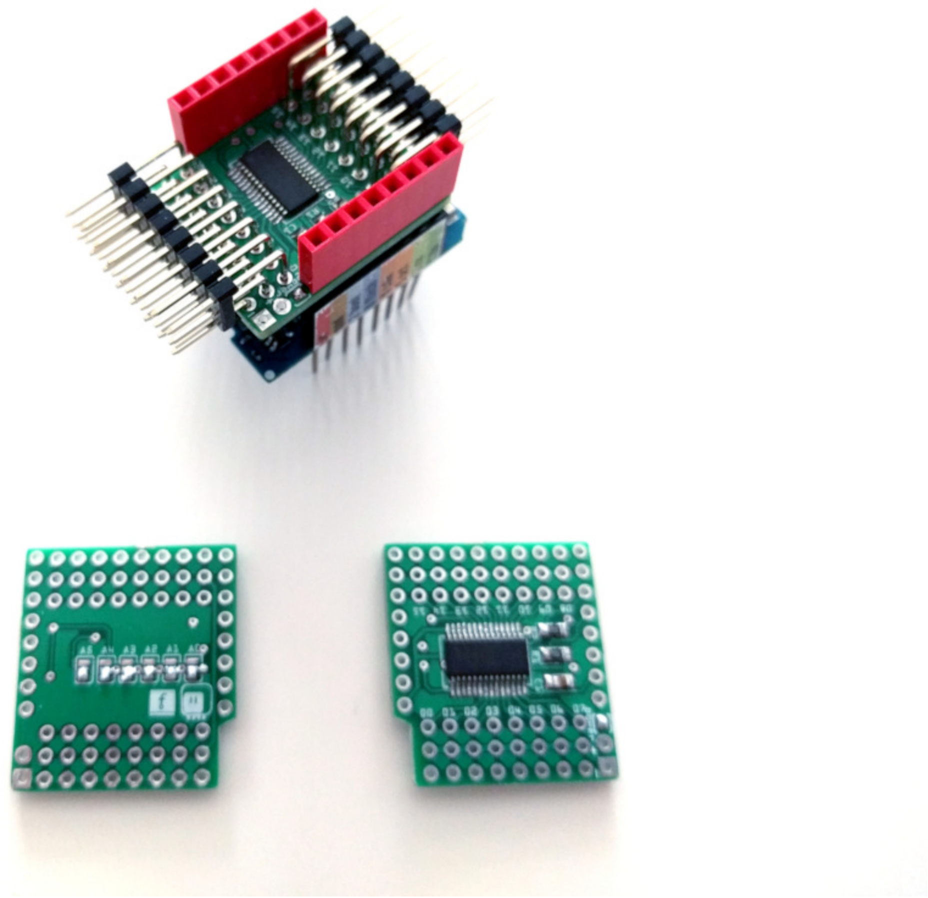

A5
A4
A3
A2
A1
A0
10°

# CHAPTER 3

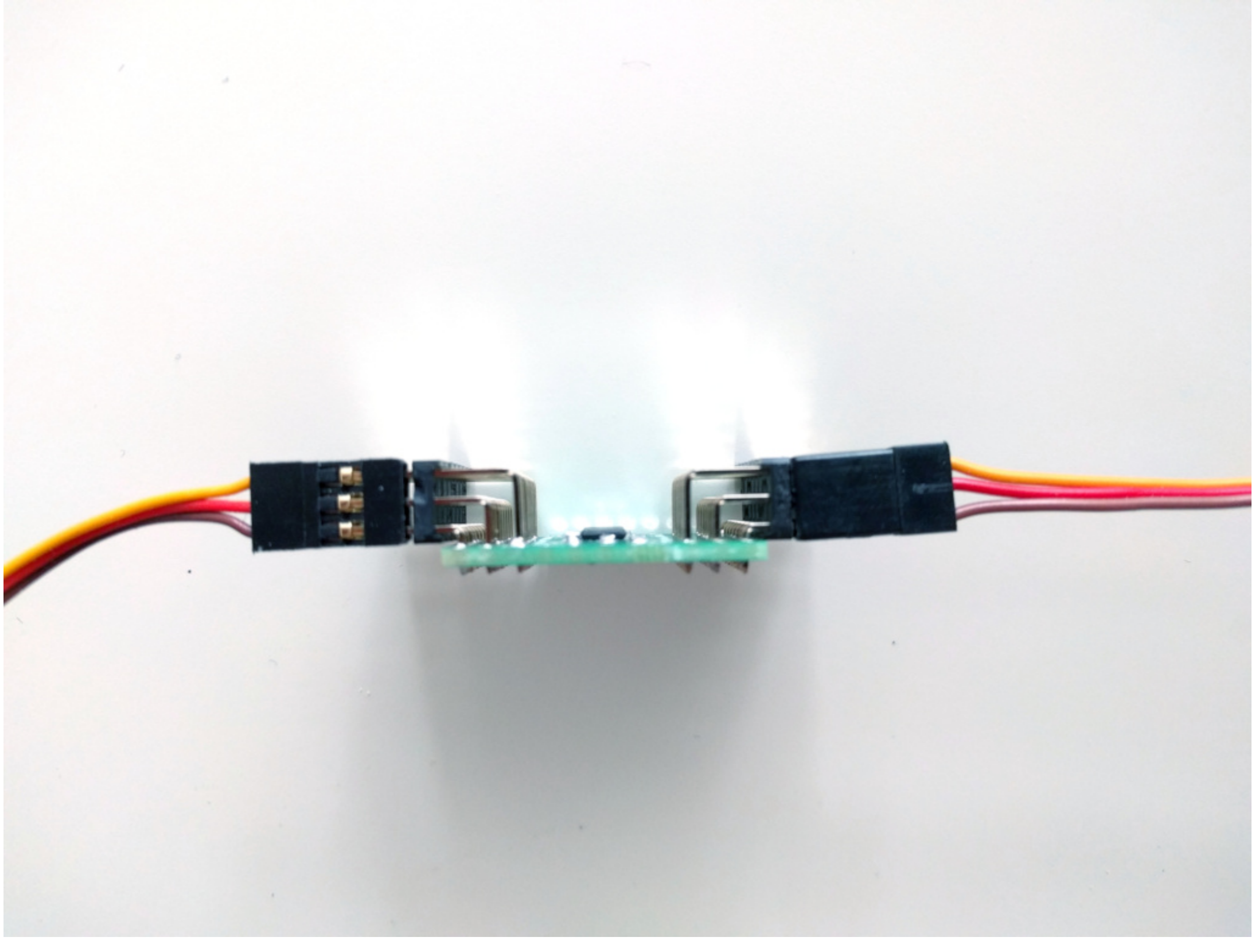
## Connecting

To use the shield, simply plug it on top or bottom of the D1 Mini, making sure that the notched corners of both boards align.

**Note:** It is physically possible to plug the shield up-side-down, but it will not work then. Fortunately, since the pins that provide power will then not be connected to the shield, it shouldn't result in any kind of damage.



Next, connect the servo plugs vertically into the servo sockets, paying attention to the order of wires: the brown or black wire needs to go to the bottom pin, and the white or yellow wire to the top.



You can stack this shield with any other shields that either don't use the pins D1 and D2, or use them for I$^2$C and have a different address than set on this shield. If you want to stack more than one servo shield on the same board, you have to change their addresses with the jumpers on the back.

**Note:** This means that you can't use it with the RGB and Relay shields, as they use those pins for other purposes than I$^2$C communication.

CHAPTER 4

# Programming

## 4.1 Arduino

To use this shield with the Arduino environment, you can use any of the libraries for PCA9685-based controllers, such as the Adafruit PWM library.

Refer to the examples and documentation provided with that library.

## 4.2 MicroPython

If you are using the ESP8266 port of MicroPython, there is also a MicroPython library you can use with this shield.

## 4.3 Nodemcu

I don't know about any libraries available for Nodemcu for this chip at the moment. However, you can use direct $I^2C$ communication to control it. Refer to the datasheet for description of the commands available.

CHAPTER 5

# Power

## 5.1 Servo Power Requirements

Powering hobby servos is trickier than with most other electronic components (except maybe for electric motors), mostly because the high power required. An average hobby servo can easily draw more than 500mA of current at 5V – which is the limit that you can get with a USB connection. If you connect 16 such servos to your computer's USB port, it will either shut down instantly or get fried, depending on how good the protection circuits are in your computer.

> **Warning:** If you try to power hobby servos from your computer's USB port, you risk permanently damaging your computer.

This also makes the common alkaline batteries unsuitable: not only their voltage will drop with such a high current draw, but they will also get discharged and very quick and possibly damaged.

You have several other options for powering your servos, but before you decide, you need to check how much current they draw. You can either refer to the data provided by the manufacturer (the maximum current draw is usually called "stall current"), or you can use a multimeter to measure the it. In either case, you have to add up all the power requirements of all your servos together, and look for a solution that can provide that much power.

## 5.2 Effects of Inadequate Power

As has been said already, in the extreme case, drawing too much current can damage whatever source of power you are using. In less extreme cases it can lead to it getting hot, and to voltage drops. Voltage drops will make your microcontroller board work unreliably and possibly restart. They can also make the servos behave pretty much randomly – buzzing, moving to random positions, or refusing to work at all.

If you are experiencing problems with your project that magically go away when you disconnect the servos, make sure your are powering everything properly.

## 5.3 Alkaline Batteries

This type of batteries is usually completely inadequate for powering hobby servos or other electric motors. With the high current draw necessary, their voltage will drop a lot and they will get discharged and/or damaged very quickly. Since they are not rechargeable, this is an expensive prospect.

You can get away with powering a small hobby servo (like the 9g ones made of transparent blue plastic) from alkaline batteries, but it's not the best way.

## 5.4 Nickel Batteries

The NiCd, NiMH and NiMg batteries, commonly available as "rechargeable batteries" for things such as digital cameras, are good for powering a couple of servos. A single cell has the voltage of 1.2V, and servos typically accept from 4.8V to 6V, so using four such cells directly powering the servos is right. They are rechargeable, so even though the initial cost is higher, you will save in the long run.

## 5.5 Lithium Batteries

The LiPo batteries are commonly used in radio-controlled toys, and they can provide very high current. They are also quite light compared to their capacity, so they are perfect for anything that walks or flies. Unfortunately, a single cell has voltage between 3.6V and 4.2V, which may be not enough for powering the servos directly (though it may still work). The two-cell (also called 2S) batteries have twice that voltage: between 7.2V and 8.4V – which usually is too much for the hobby servos. The solution is to use a buck converter for the higher voltage, or a boost converter for the lower. See below.

Lithium batteries are a bit tricky in handling, and can be dangerous when mishandled (remember the exploding cellphones?). They require a special charger, and if you discharge them below 3.6V, you risk damaging them permanently. They are definitely an advanced device.

## 5.6 Voltage Converters

If the battery pack that you are using doesn't have the correct voltage for your servos, you can use a voltage converter. Those come in two kinds: buck converters take higher voltage and drop it to the required output, and boost converters take lower voltage and "boost" it to the set value. There are also converters that have both a buck and a boost converter inside, and they can deal with any input voltage (within their range, of course). They are commonly called "BEC" or "UBEC" in the radio control hobby communities.

Using a voltage converter is a good way to get exactly the voltage you need, and to get the most out of the batteries – it will give you the same voltage no matter how discharged the battery is. Some even come with a battery protection circuit, that shuts off the battery when it gets discharged too much – which is very handy for LiPo batteries.

However, voltage converters also have their limits for the current they can handle. When choosing a converter for your project, make sure that it can handle the current that your servos are going to draw. (Of course, the battery also has to be able to handle that.)

## 5.7 Power Banks

The portable battery packs commonly used to recharge USB devices such as smartphones usually contain inside a LiPo battery, a boost converter and a recharging circuit. They can often provide as much as 2A of current, which should be enough for powering a couple of servos moving at once. They are also relatively easy to handle, and the built-in protection makes them safer than raw LiPo batteries.

## 5.8 Phone and Laptop Chargers

If your project is not mobile, it is convenient to simply power it from a wall socket. You will need a power unit for that.

The phone chargers are very easy to obtain, but you have to be careful to check how much current they can provide. The cheap chargers often don't have any kind of protection, and will simply melt or burn if you try to draw too much. The better ones have a protection circuit that will shut them off.

If you need more power, a laptop power unit, or a power unit from any other appliance may be suitable, provided that they give the right voltage (between 4.8V and 6V) and enough current. Connecting a unit with higher voltage than your servo (and microcontroller board) accepts will result in damaging them. Even if the effect is not instant, powering them with too high voltage can damage them over time, so make sure you get the right model.

## 5.9 Tips and Tricks

If you can, try not to move all the servos at once in your program, but instead move a few at a time. This way you can lower the amount of current necessary for them and avoid current spikes and voltage drops. In particular, at startup, don't just switch all the servos on and move them to their home positions, but instead switch them on one by one, with some delay between them.

If you have some current spikes, but otherwise your current needs are lower, it may help to add a large electrolytic capacitor across the servo power – in the range of 10-100µF. It will act as a buffer, providing extra current when there is a spike, and will help avoid voltage drops.

# CHAPTER 6

## Indices and tables

- genindex
- modindex
- search