Converting a 3D Printer Into an Automatic Spray Coater

by Remi_Rafael

In this instructable i will give you all the tips I know to convert a 3D printer into an automatic spray coater.

I worked for a while on the development of conductive inks based on copper particle. To carry out my tests i needed a method to reliably spray a repeatable amount of ink onto my samples. Given the budget constrains i decided to convert a 3D printer into an automatic spray coater using a simple air-brush pen. The results were fairly good in terms of layer homogeneity, so hopefully this may be useful to others.

The main parts of this tutorial will consist in:

- Making a spray head controllable with a servomotor
- Setting up the electronic to automatically control the spray
- Generating a Gcode to control the machine

I will also introduce two facultative steps consisting in:

- Adding a mechanical stirring system to keep particles in suspension in the ink
- Making an adjustable support to spray at an angle

You should read those additional steps and decide whether to follow them before beginning construction

Supplies:

Tools

- 3D printer
- Soldering iron

Supplies you will need:

- A 3D printer frame: almost any 3D printer or CNC frame can be used. Reprogramming the motherboard is facultative. If you decide to by it specifically, one build from aluminum profiles will facilitate the fixation of the various elements.
- An double action air-brush pen: Most basic types of airbrush will work. You can see in the pictures what to look for, however the models with big handles and with the pump directly attached to the intake will not be adequate.
- A pump, compressor, or a compressed air source, ideally with a pressure regulator
- An Arduino UNO or a similar programable board
- A DC to DC step down transformer to produce 5V (example: https://www.amazon.com/HiLetgo-Step-down-Converter-1-25-37V-Voltmeter/dp/B00LSEBYHU/ref=sr_1_8?keywords=dc+to+dc+buck+converter&qid=1667915974&sr=8-8)

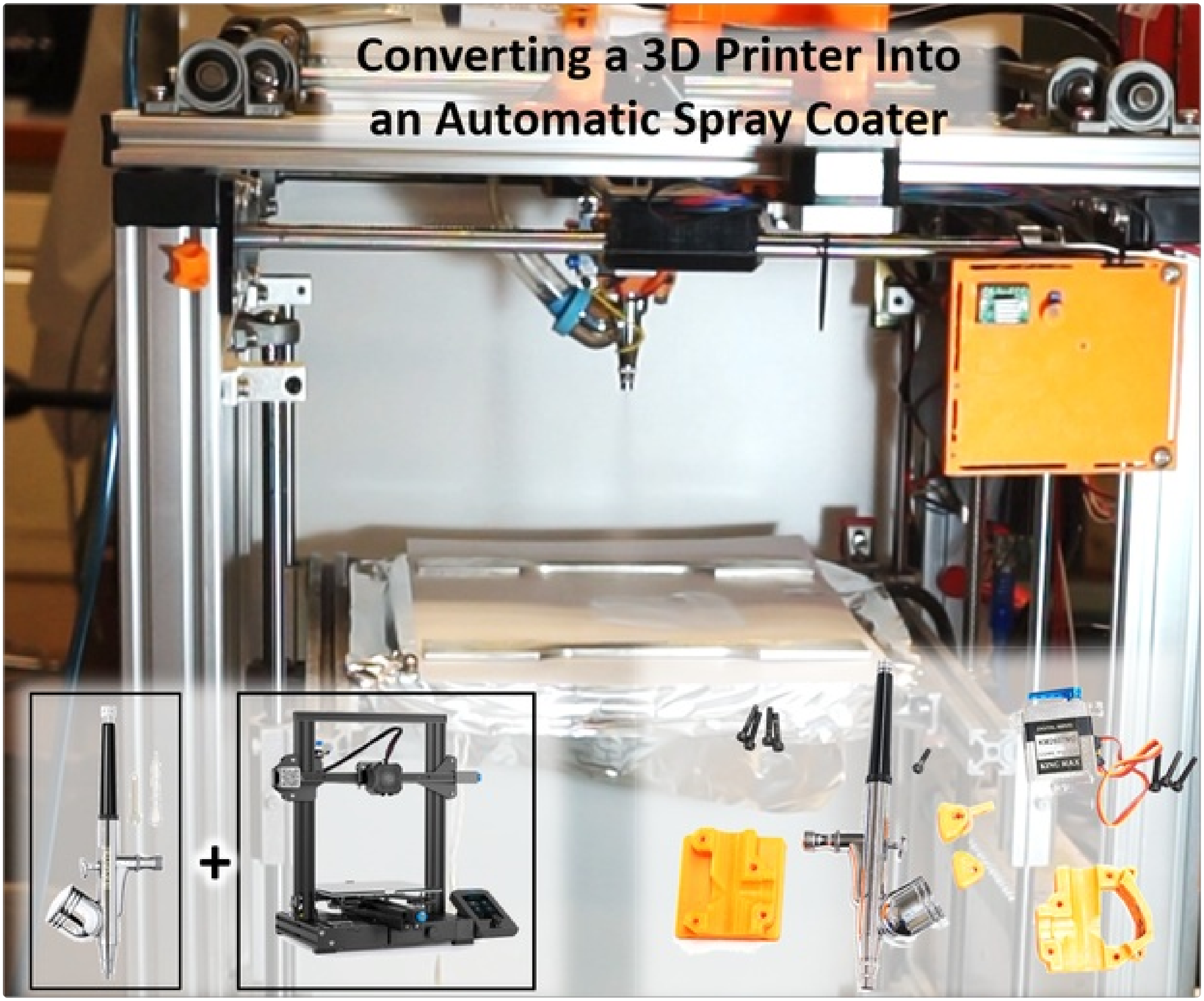
- A servomotor type "Slim Wing Servos" (i used a kingmax km2607md)
- Diverse small cables and dupont connectors
- Various M3 screws
- optional: M5 or M4 screws and T nut to attach the elements to aluminum profiles

Optional for the mechanical stirring system:

- A second arduino
- A stepper motor
- A pololu stepper driver or equivalent
- Potenciometer

3D models for printing: https://www.thingiverse.com/thing:5668363

Aduino routines for control and Python program for Gcode generation: https://github.com/RemiRafael/Spray-coating-Gcode-generation



Step 1: Understand and Adapt the General Strategy

The general idea of this build is to take a 3D printer, remove the extrusion head, replace it with a spray head built from an airbrush, and control it with the original motherboard. Depending on the exact characteristics of your printer frame and whether you can reprogram the motherboard, the optimum strategy may vary. I my case the printer had a cooling fan controlled with a 12V DC signal and I was unable to reprogram the motherboard (here cooling fan always refer to a fan aimed at the printed part to cool it down). In consequence I decided to use this signal to control the head and to use an Arduino UNO to convert it into a PWM compatible with the servomotor in (almost) real time.

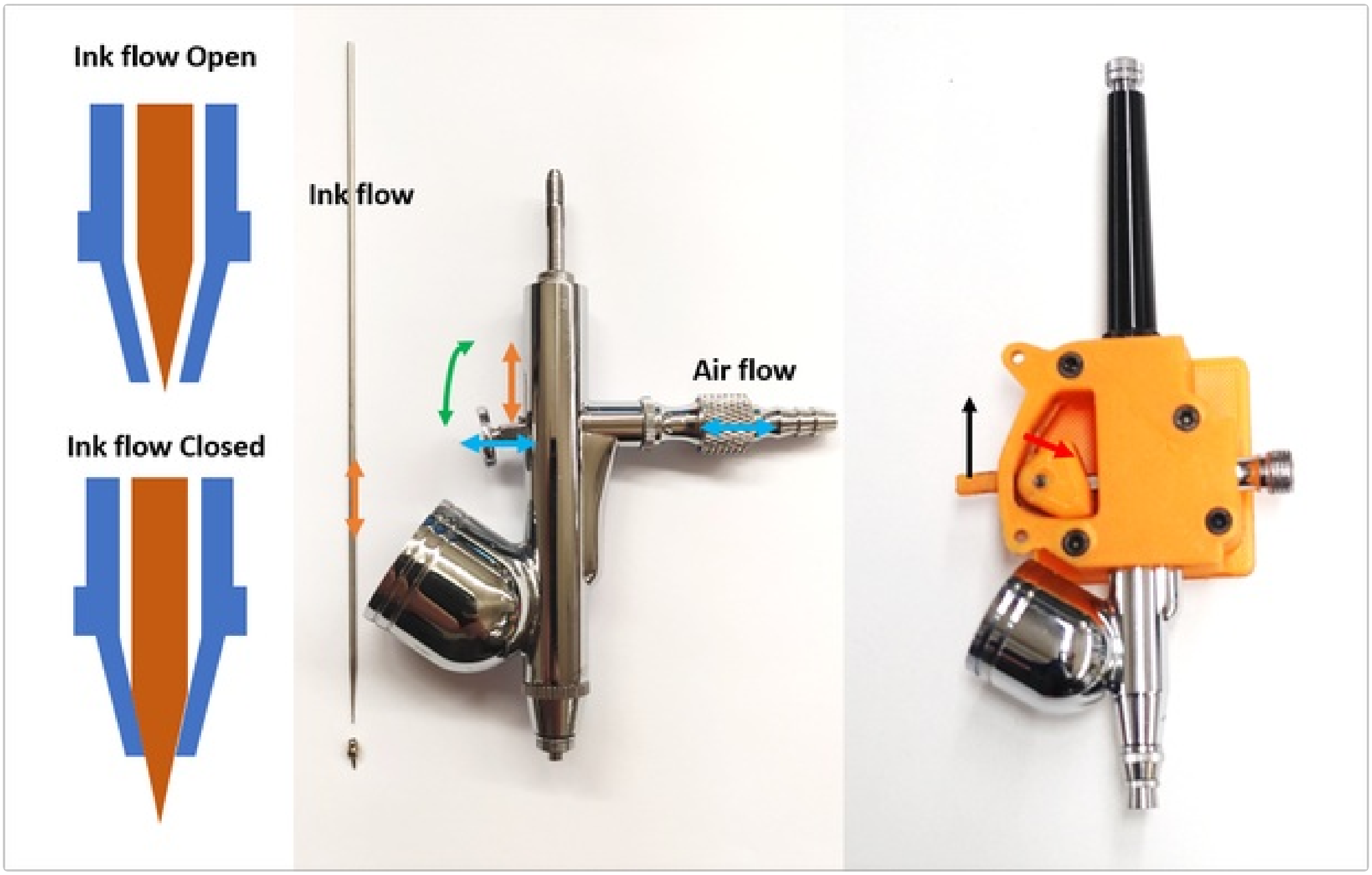
If your mainboard can produce a PWM and you have the possibility to reprogram it, it would be more efficient to directly use it to control the servomotor, but the method will not be explained in this instructable.

If your printer has no cooling fan connected you can check the mainboard documentation to identify the correct output or send the command M106 S255 (cooling fan ON) to identify the right pins.

Whatever type of frame and mainboard you decide to use, the Python routine i provide further along this tutorial generate a Gcode that rely on G1 commands for the movements and M106 commands for the control of the spray. You should check how your setup react to those commands or devise you own control strategy before beginning the built.

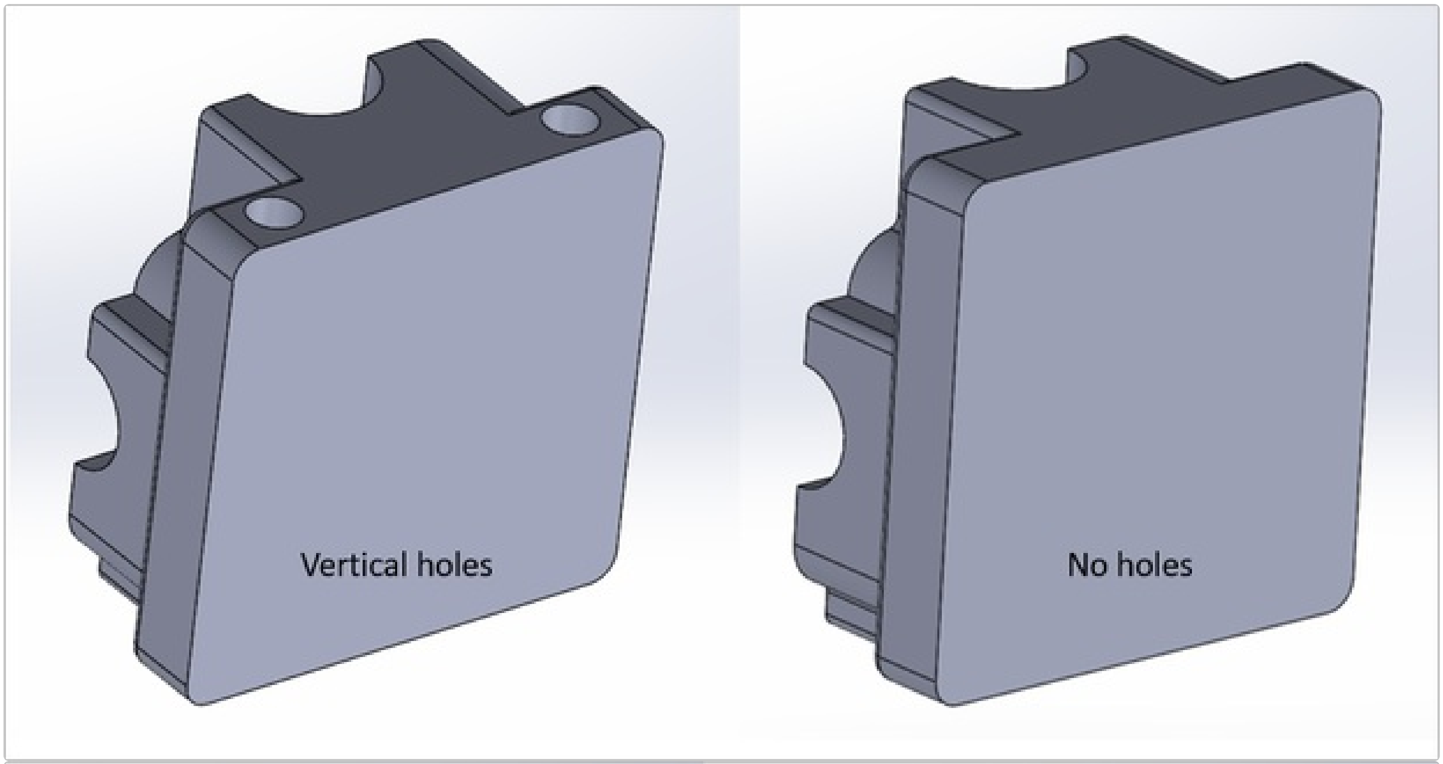
Step 2: Principles of the Spray Head

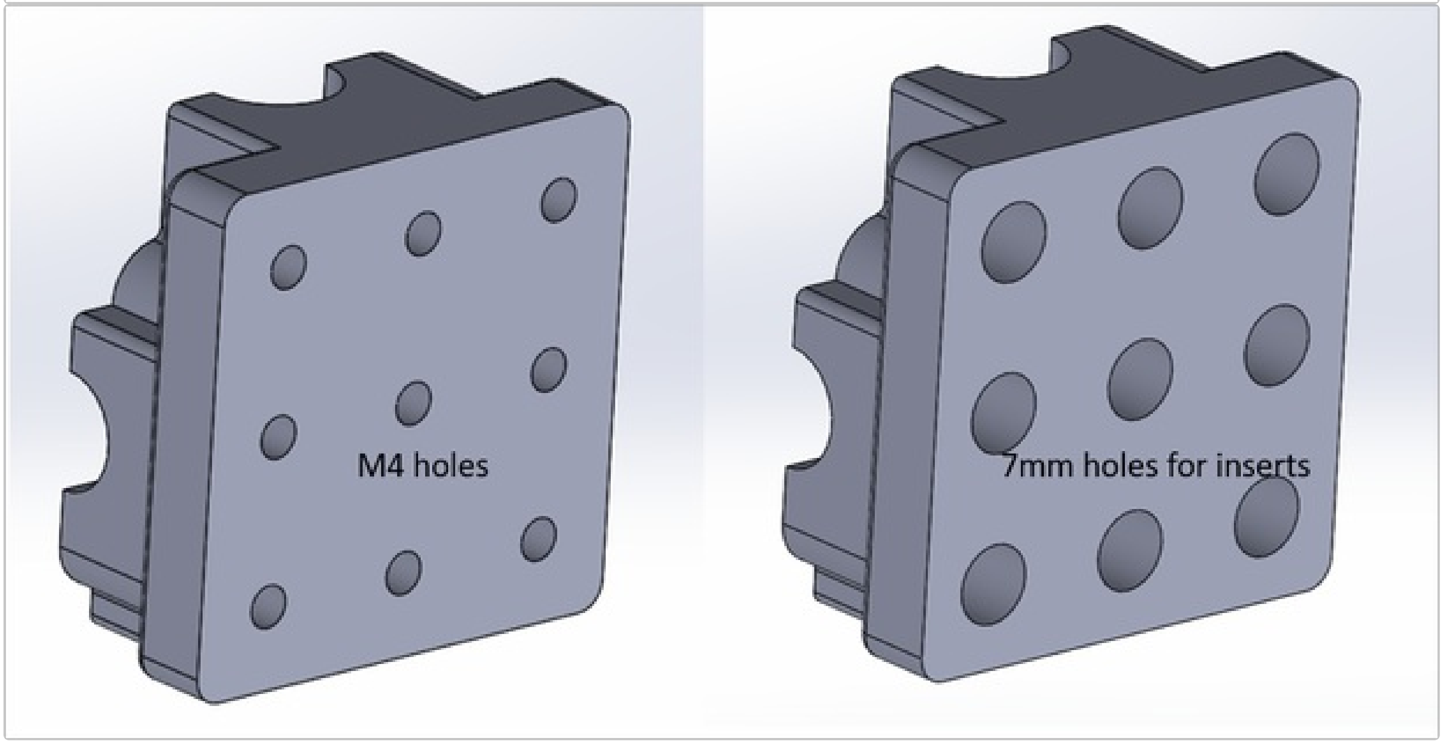
"Double action" air brush are called so because the action of the trigger has to effect: Opening the air valve integrated to the brush to release pressurized air and pushing the internal needle to the back to open the way for the ink to flow out of the nozzle. The air flow is controlled by the horizontal movement of the trigger (in blue on the figure) while the vertical movement (in red) controls the ink flow. The superposition of those two movements (in green on the schema) is neither linear nor a rotation and is actually pretty difficult to pilot with a single servomotor (actioning only one valve will not result in a spray). To force those two actions simultaneously, I decided to use a servomotor to push the trigger to the back (black arrow) and use a use a fix "wall" to force it to the side (red arrow). This system ensures the proper function of the system and allows to control the amount of ink delivered depending on the angle programmed for the servomotor. However it induce some friction and the torque of the servomotor cannot be too low.



Step 3: Choose a Spray Head Model

Before building your spray head, you need to chose how you will attach it to your printer. The junction part is called "Spray_pen_holder_part_1" on Thingiverse and i uploaded 4 models. The one called "vertical holes" is the one i used as i have two long vertical screws on my printer carrier to attach the spray head. The model "no holes" is here if you want to draw directly the holes adapted to your printer using a CAD software. Otherwise you can use the "M4 holes" or the "7mm holes for insert" models. Both have an array of 9 holes meant for M4 screws, separated by 15mm. Use the first one if you want the screws to directly form their threads in the plastic, of the second if you prefer to use heat inserts.

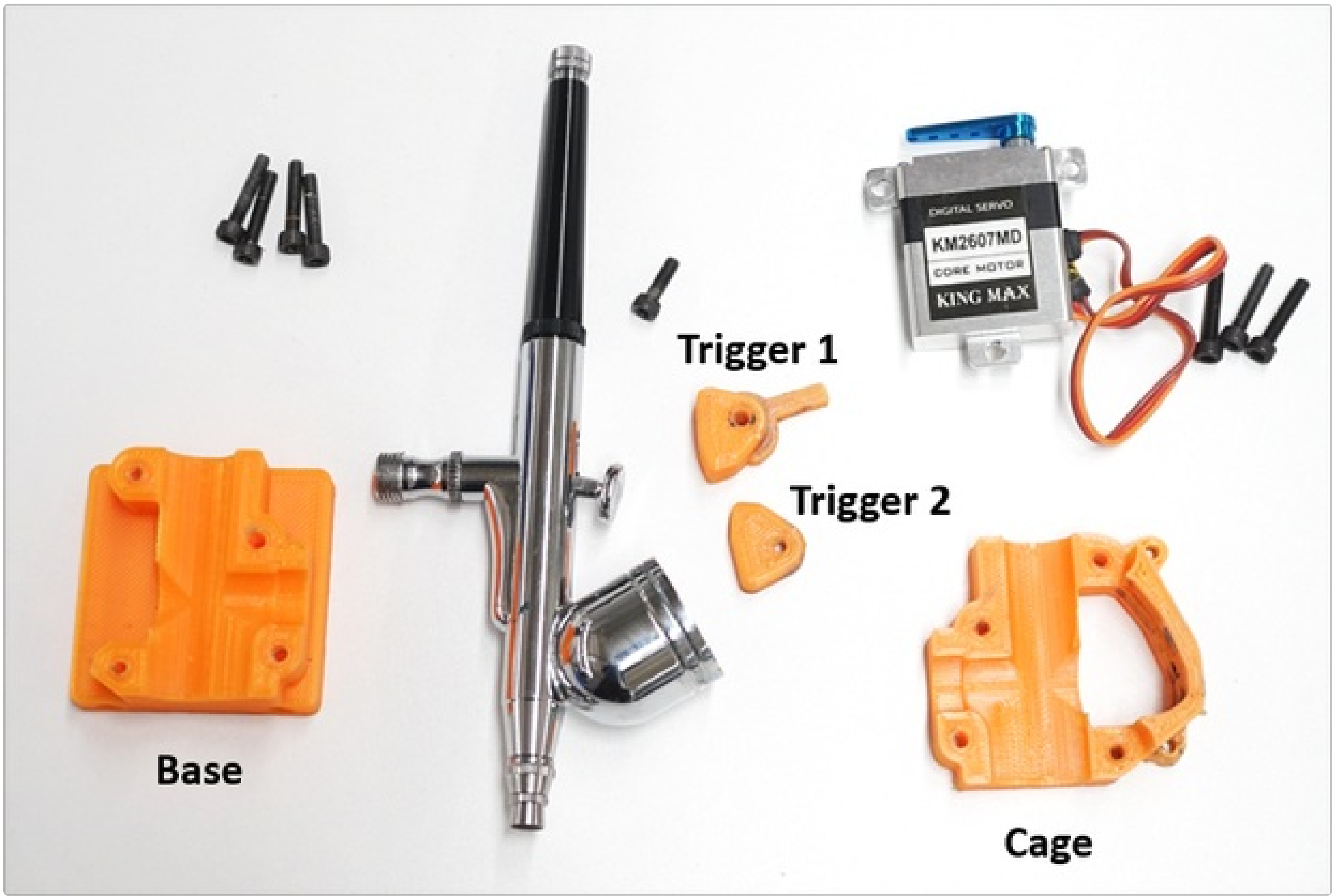


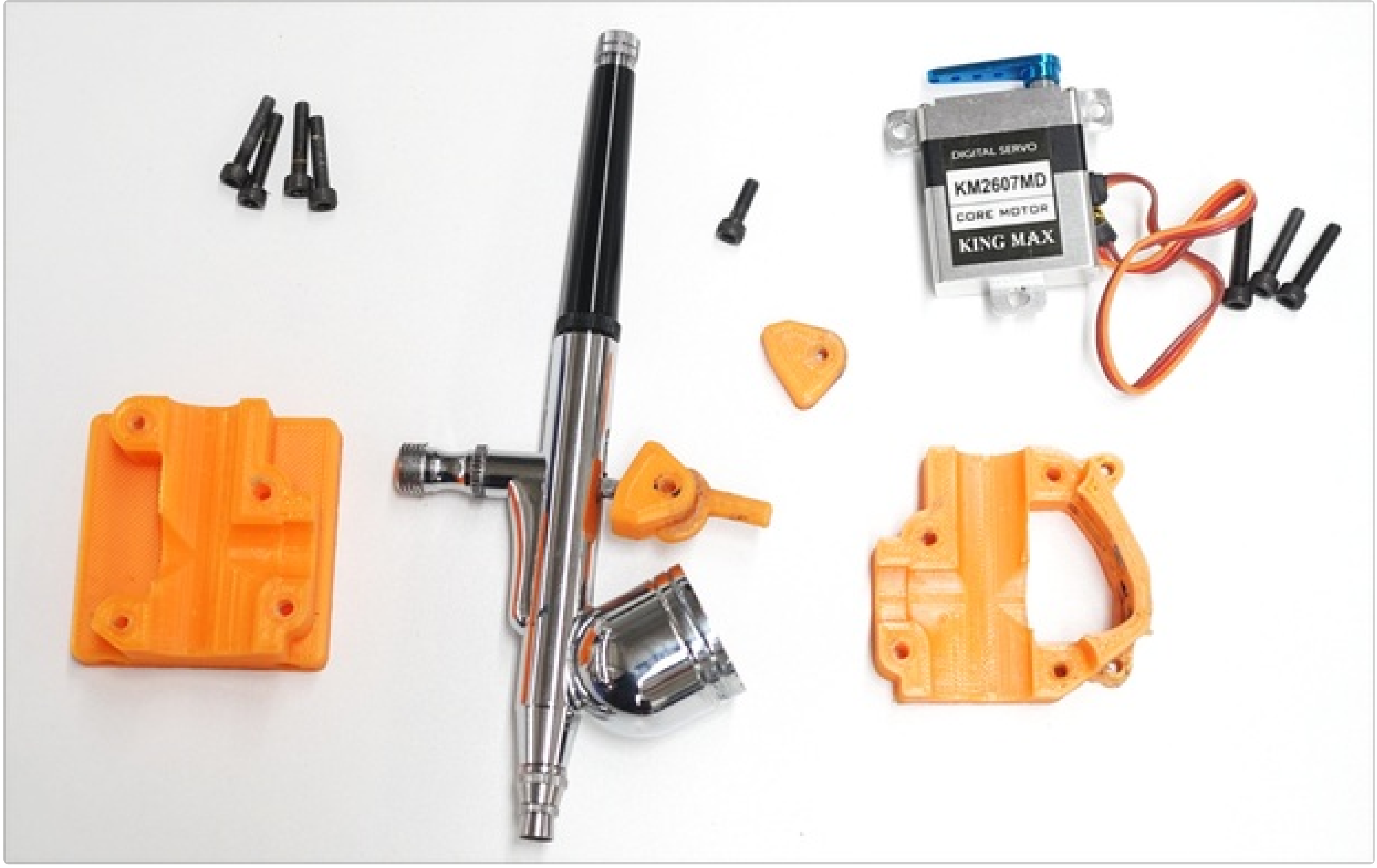


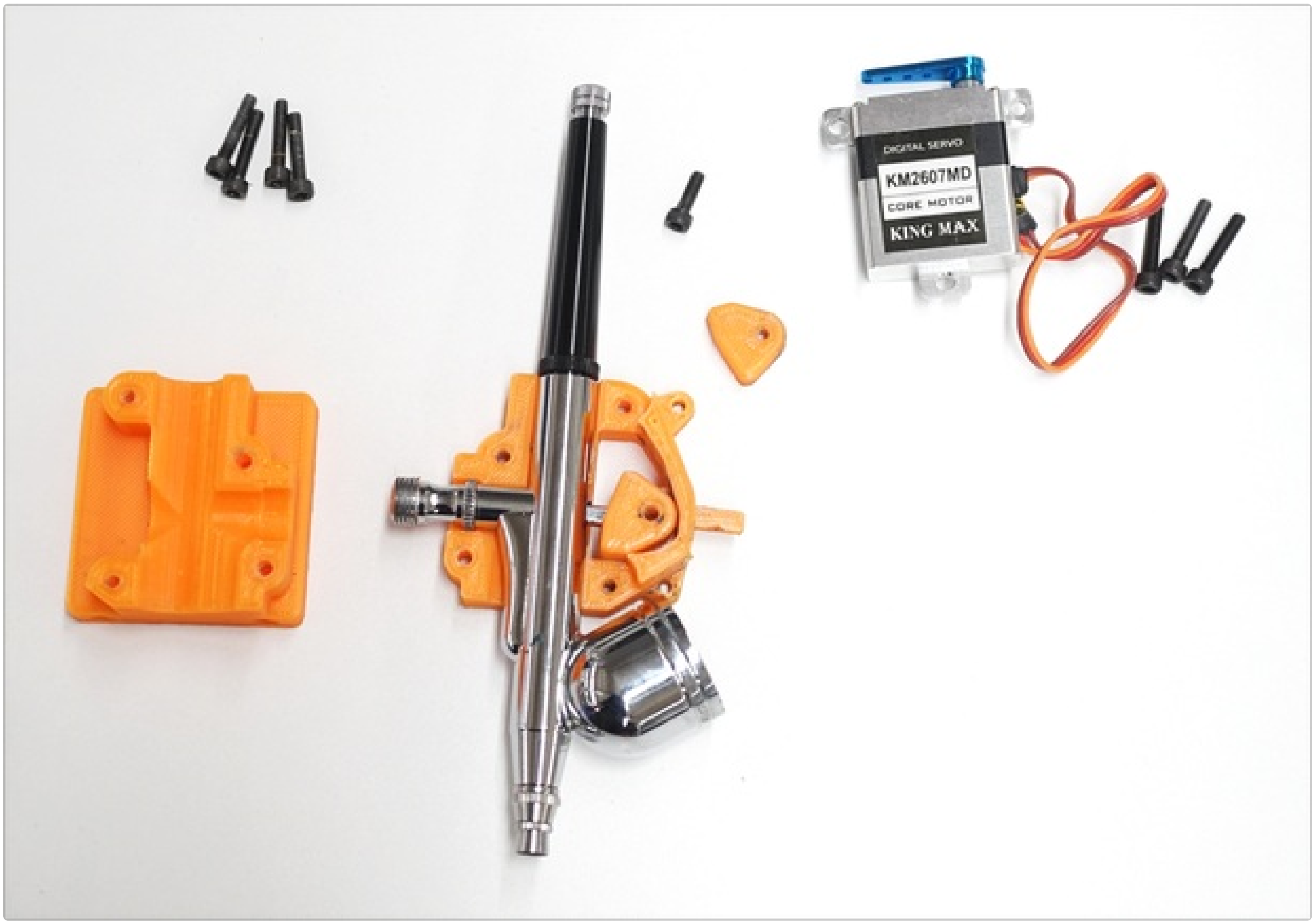
Step 4: Building the Spray Head

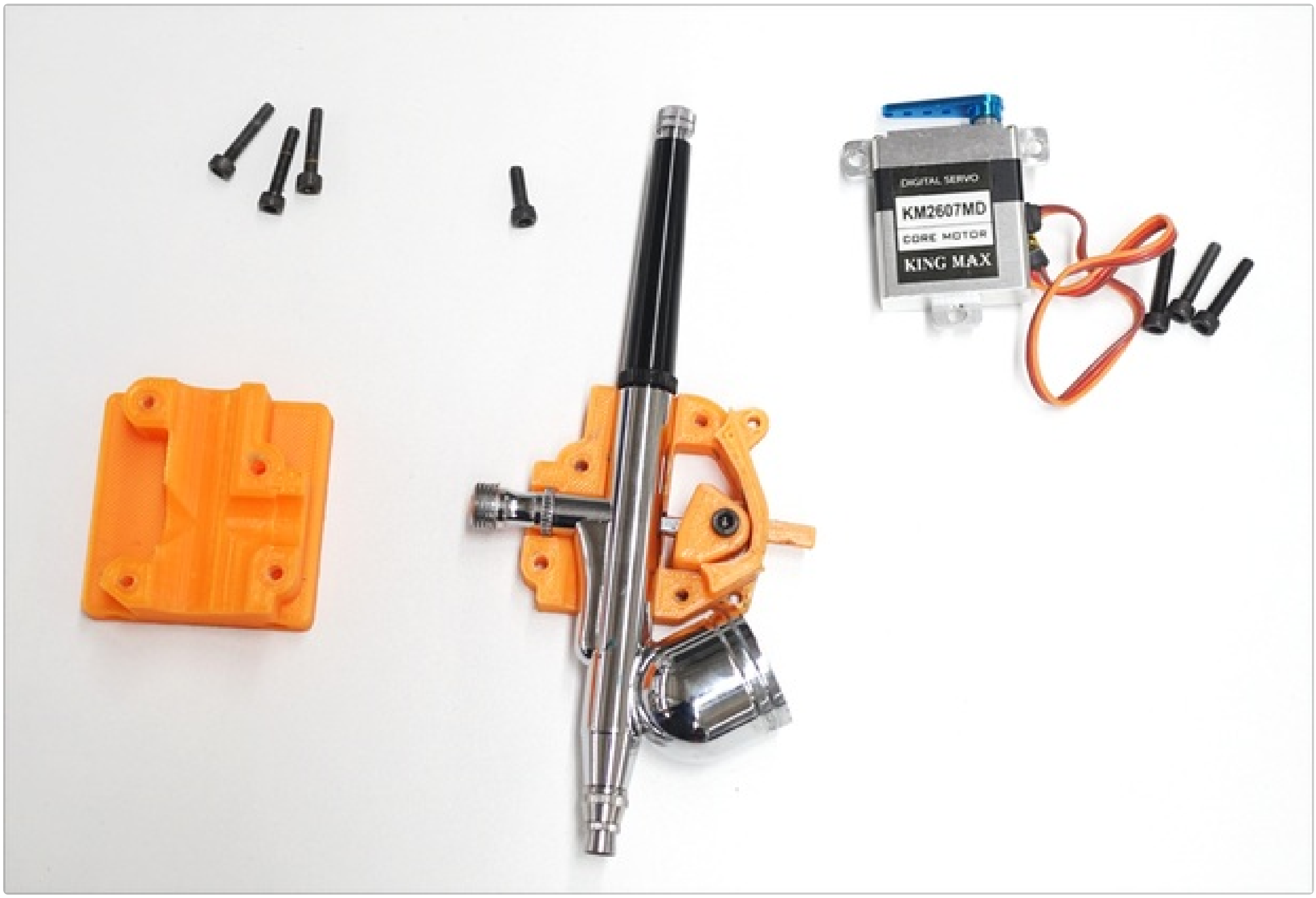
To build the spray head just print all the parts available on Thingiverse and assemble them with M3 screws according to the series of images: https://www.thingiverse.com/thing:5668363

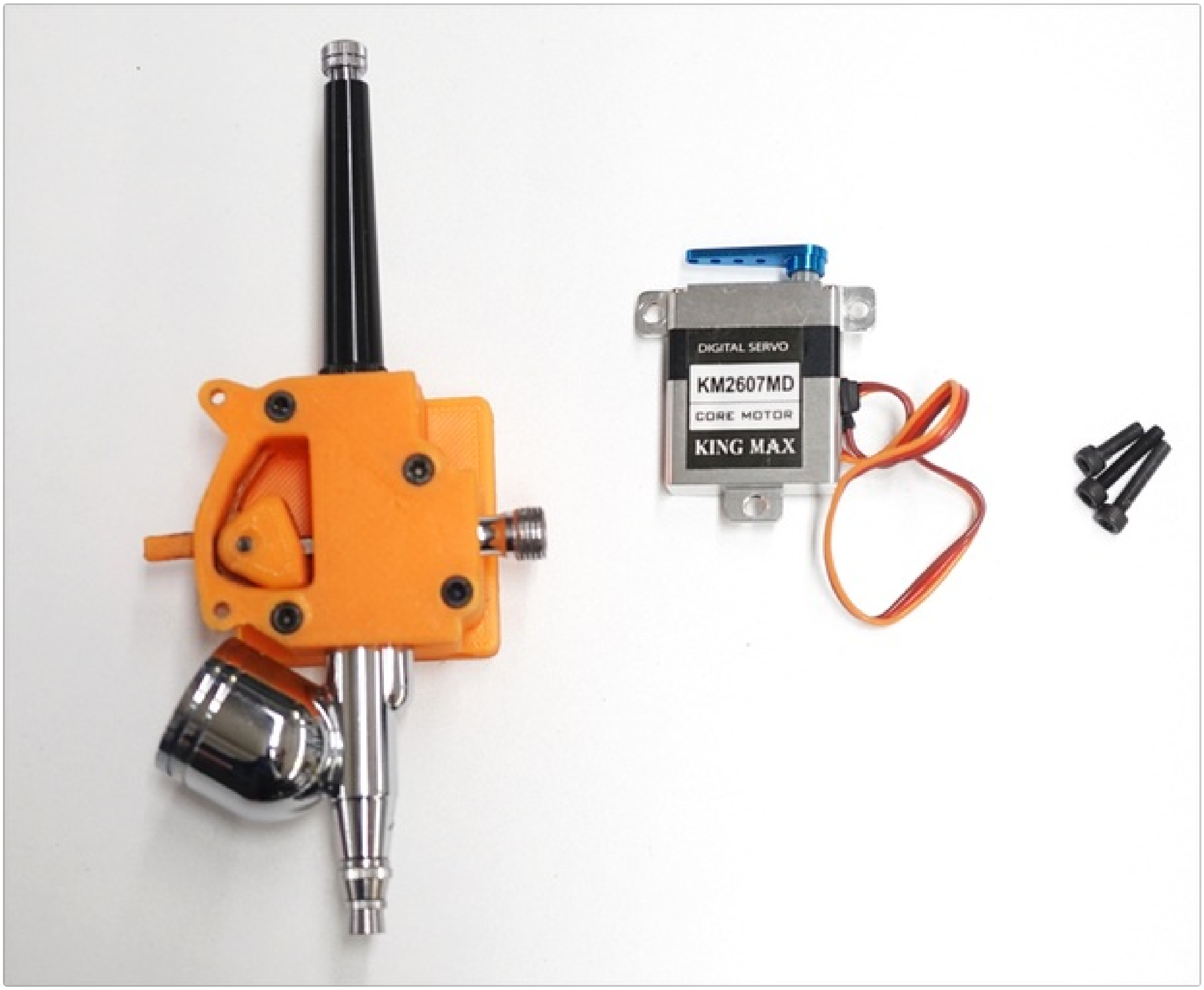
If the friction is to high in your assembly and the trigger doesn't correctly comes back in position you can use graphite powder as lubricant on the mechanism or add a rubber band according to the last to pictures presented here. Both solutions worked for me but the graphite tends to stain everything around. The rubber also slightly adds to the efforts required of the servomotor so only use it if you need.

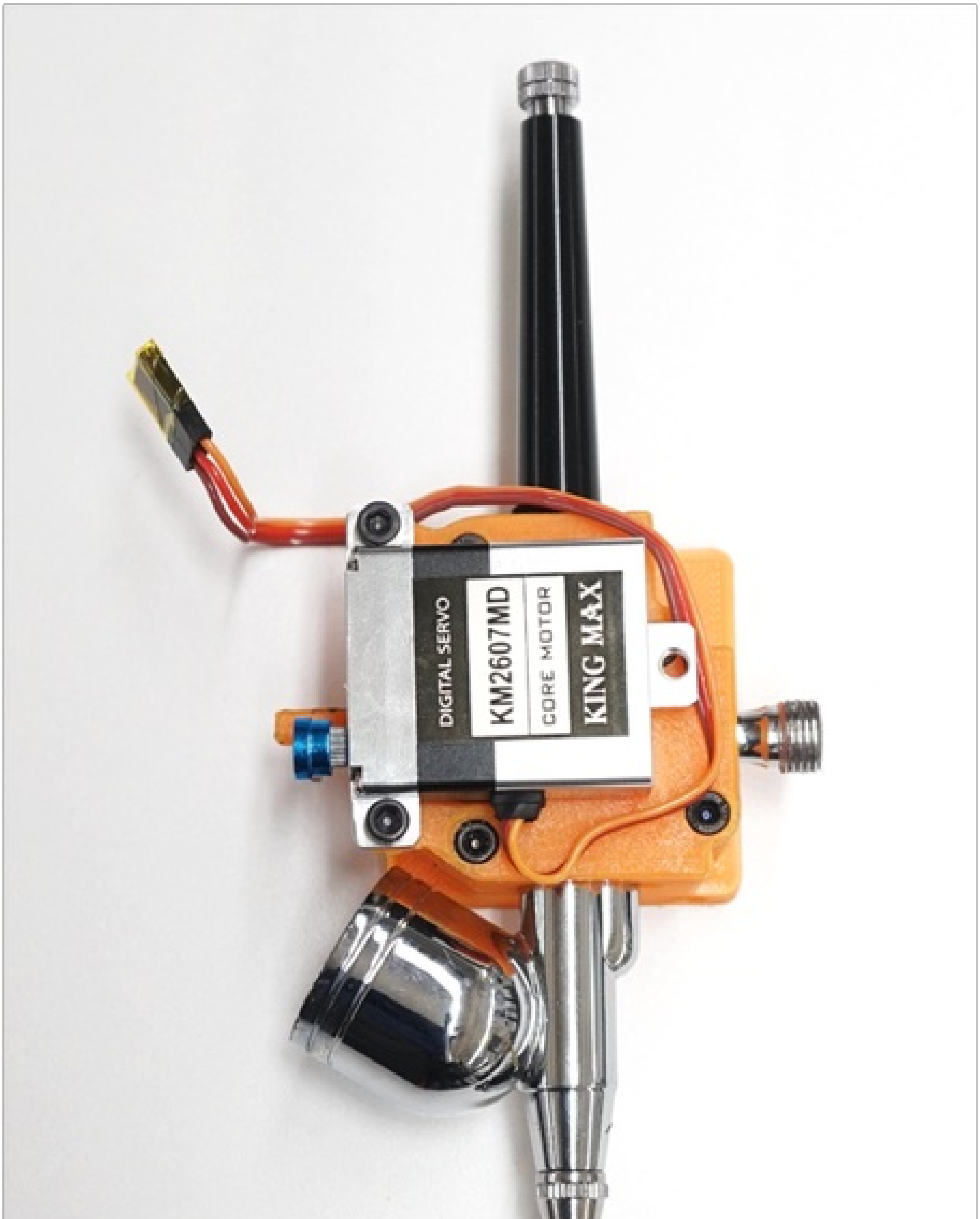

DIGITAL SERVO
KM2607MD
CORE MOTOR
KING MAX
Trigger 1
Trigger 2
Base
Cage

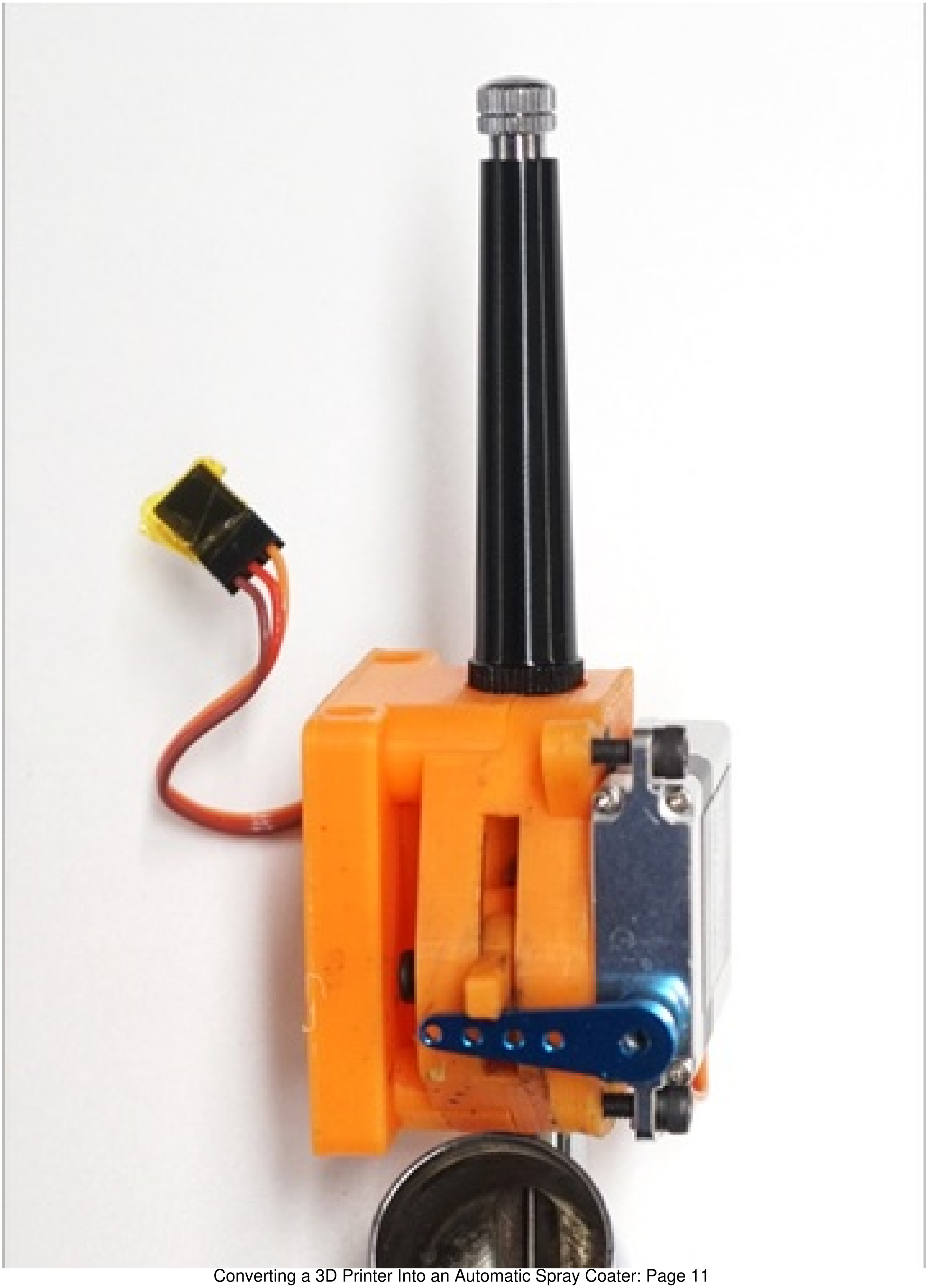

DIGITAL SERVO
KM2607MD
CORE MOTOR
KING MAX

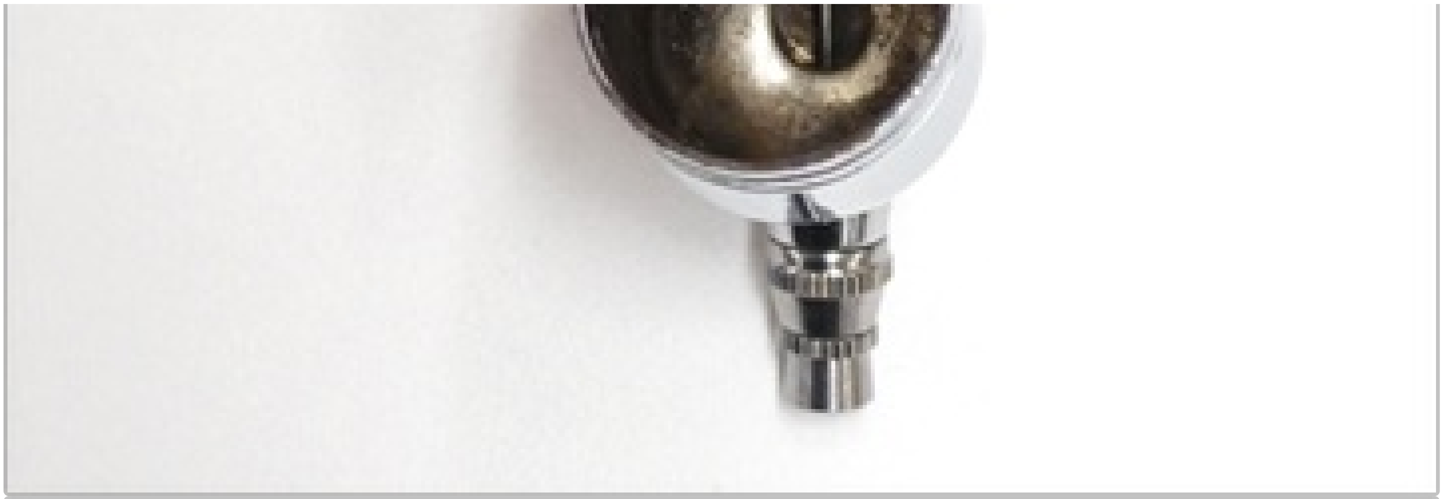

DIGITAL SERVO
KM2607MD
CORE MOTOR
KING MAX


DIGITAL SERVO
KM2607MD
CORE MOTOR
KING MAX

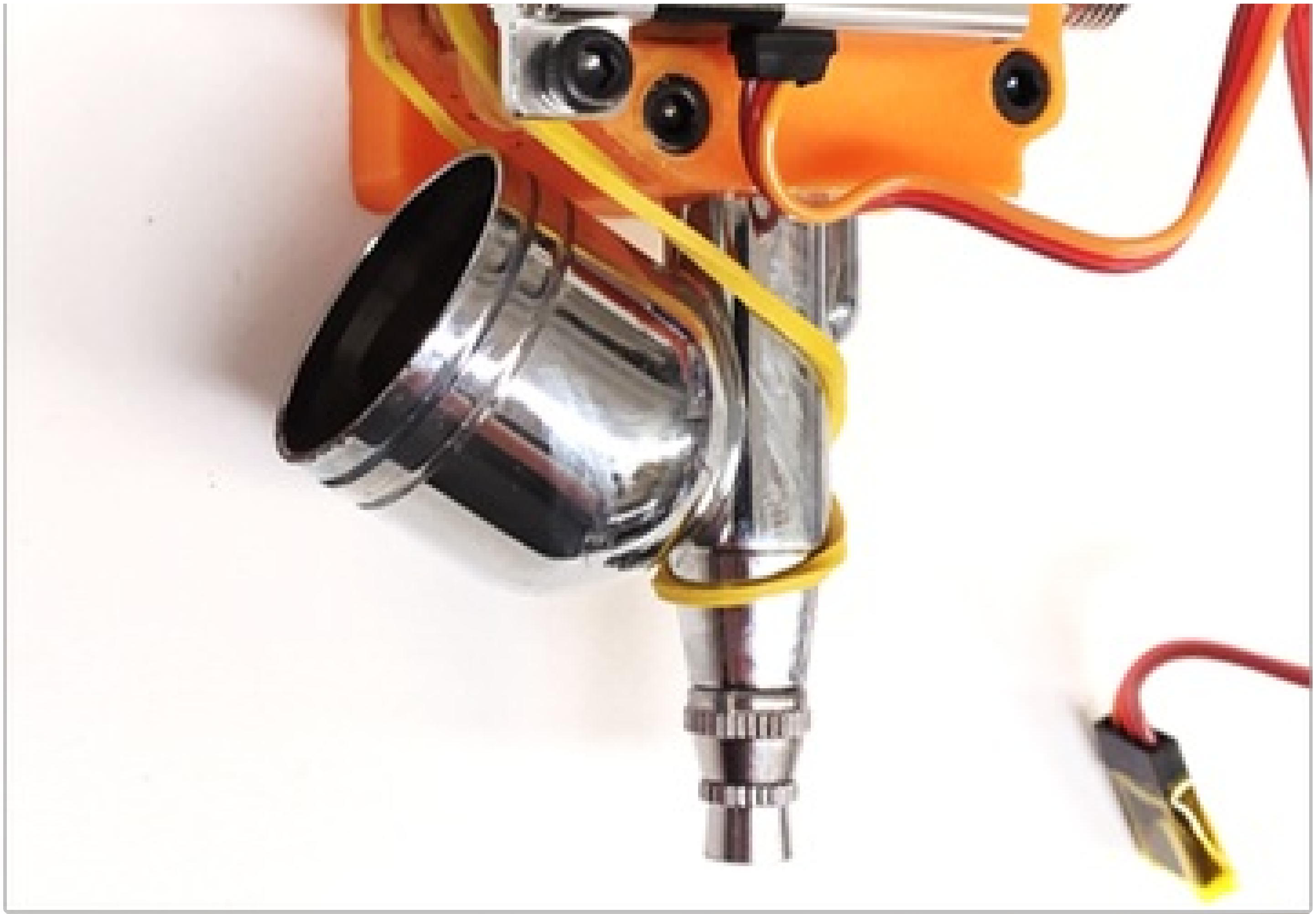

DIGITAL SERVO
KM2607MD
CORE MOTOR
KING MAX

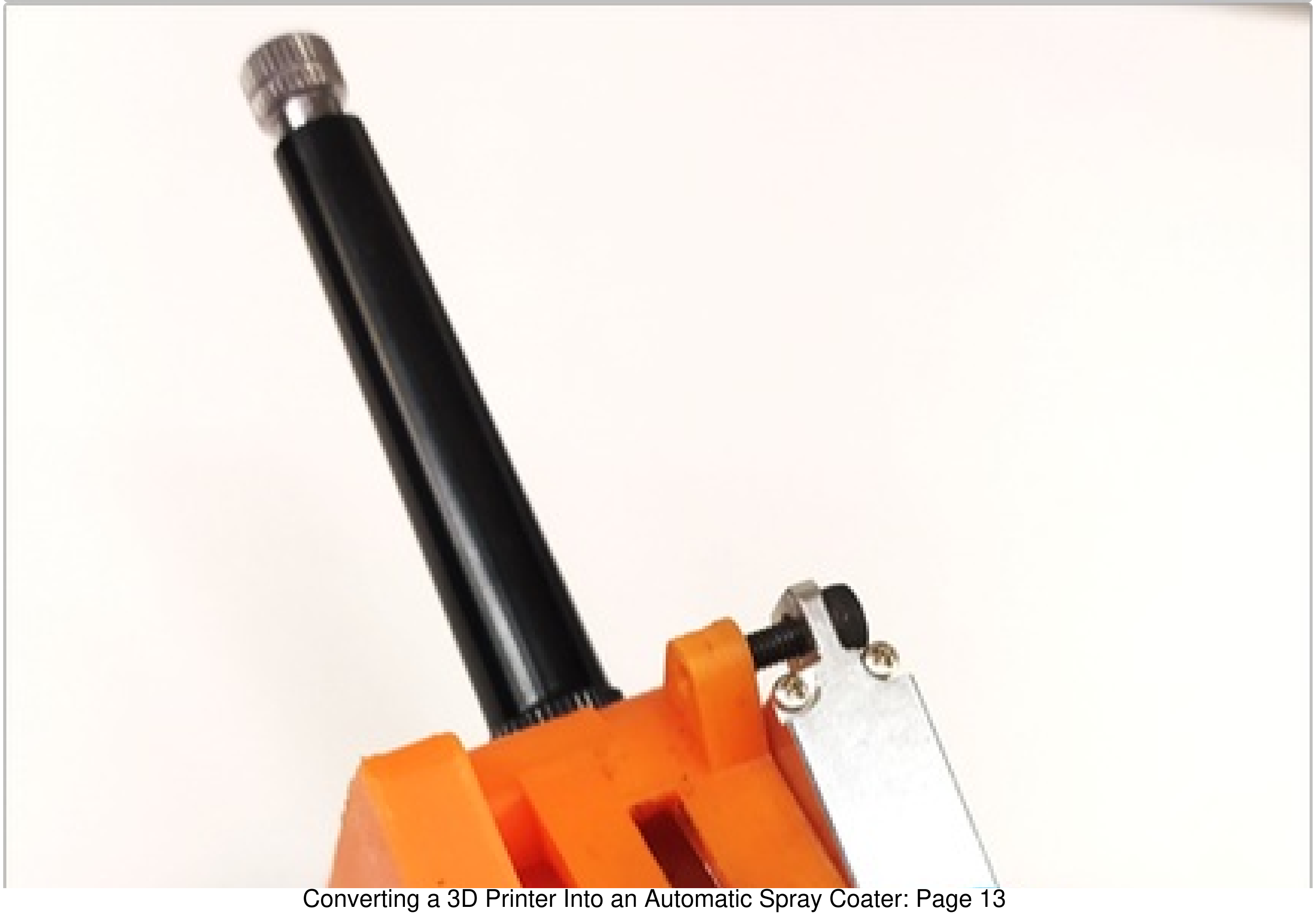

DIGITAL SERVO
KM2607MD
CORE MOTOR
KING MAX





DIGITAL SERVO
KM2607MD
CORE MOTOR
KING MAX





Step 5: Pre-Connecting the Spray Head

After building and and attaching the spray head to your frame, the next step is the electrical interconnection. This image illustrates the electrical circuit used.

ATTENTION: This circuit is valid without modification only if your main alimentation is 12V, you decide to use the same strategy as me to control the spray head and your main board uses a DC signal >5V to control the cooling fan.

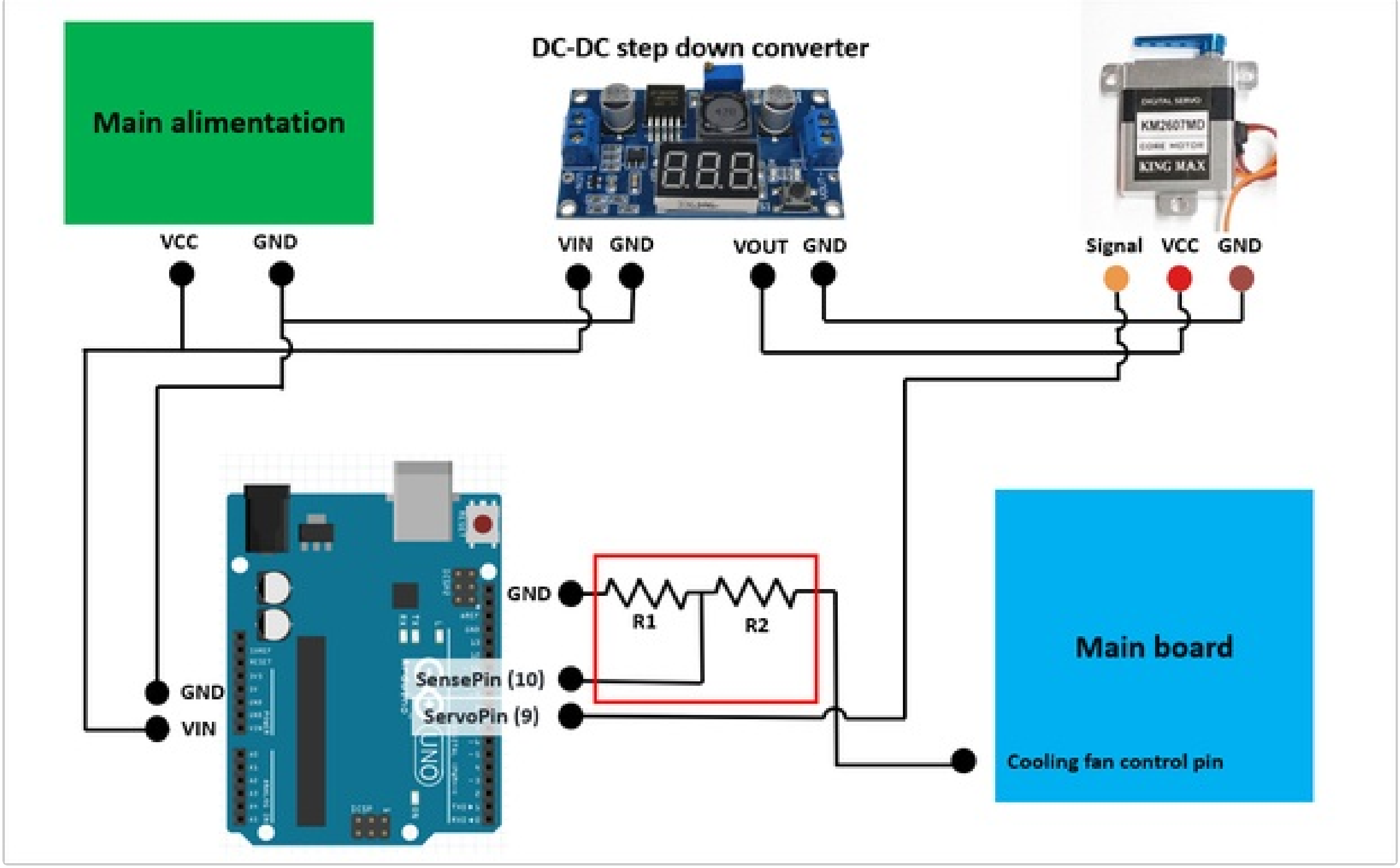
Before assembling the circuit, identify the cooling fan control pin on your printer's mainboard and check the voltage on the main alimentation. This voltage should be 12 or 24V. All the components can be alimented from the main alimentation but the required voltage may vary. The arduino UNO accepts 6 to 20V between the pins VIN and GND. If your alimentation is 12V you can plug it directly.

The servomotor is typically alimented in 5V which can be produced by the Arduino's internal transformer. However the servomotor's current consumption is too high for the arduino and another step down converter is required. I used a LM2596 Adjustable DC-DC Step Down Buck Power Convert Module but more simple fixed voltage converter can work just fine. Connect your converter input to the main alimentation, set it to 5V and connect to the VCC and GND pins of the servomotor. For optimum performances (fast trigger movements), check the optimal alimentation voltage of the servo and set your converter accordingly.

Finally you can connect the pin 9 of the Arduino to the signal pin on the servomotor: You are now ready to charge a program on the Arduino.

If your main alimentation uses 24V you cannot connect it to directly to the Arduino. Instead check the maximum voltage supported by the servomotor and set the step down converter to that value. It must be at least 6V. You can then use the converter output to aliment both the Arduino and the servo.

The connection between the Arduino and the motherboard (red rectangle) is described in a next step.



Step 6: Testing the Correct Servo Angles

The next step consist in loading a control program in the Arduino to test the correct angles to send the servomotor to control the spray head. If you are not familiar with Arduinos and how to charge a program you can look here: https://www.instructables.com/A-Beginners-Guide-to-Arduino/

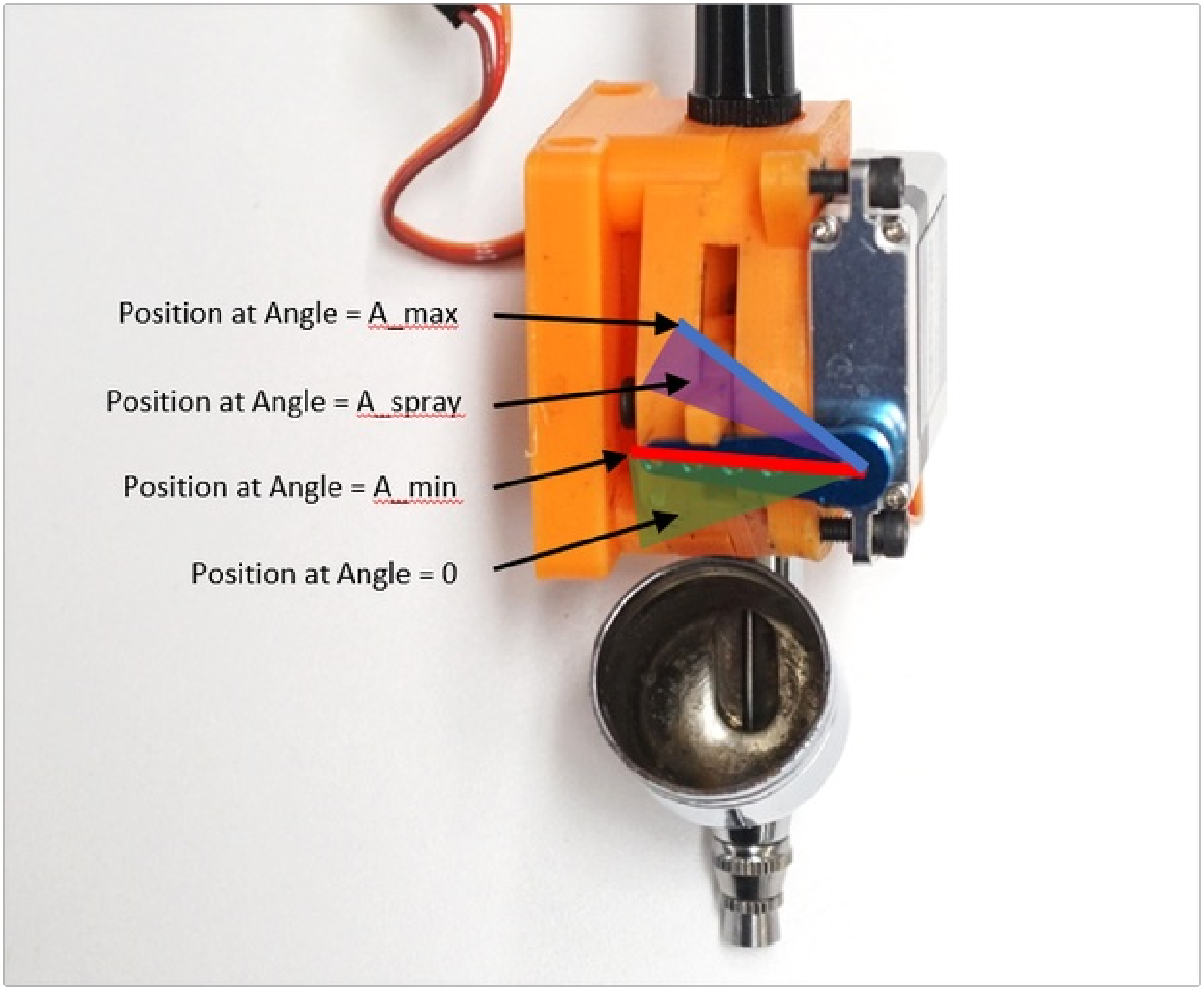
Go to my GitHub repository (https://github.com/RemiRafael/Spray-coating-Gcode-generation) and download or copy the routine Test_angle.ino With this basic routine you can program the position of the servomotor by changing the value of the variable Taget_angle and loading the routine.

First go to the position 0 and fix the arm of your servomotor to the shaft below the position of the trigger. You can also fix the spray head to your printer frame.

At that point you need to test and register the value of 3 angles: A_min, A_max and A_valve:

Depending on the position of the trigger the spray will behave differently: When you begin to action the triger the needle inside the spray brush will move to slowly open a path for the ink at the nozzle. The angle where the needle begins to move is A_min. However initially the air valve in the brush will remain close and no spray will be released. The angle at which the air valve oppen is A_valve. At that point the spray begin but the needle is still partially blocking the ink path and the spray will be thin. When continuing to increase the angle you will eventually reach the end of the trigger play A_max. At this point the ink path is fully open and the intensity of the spray is maximum.

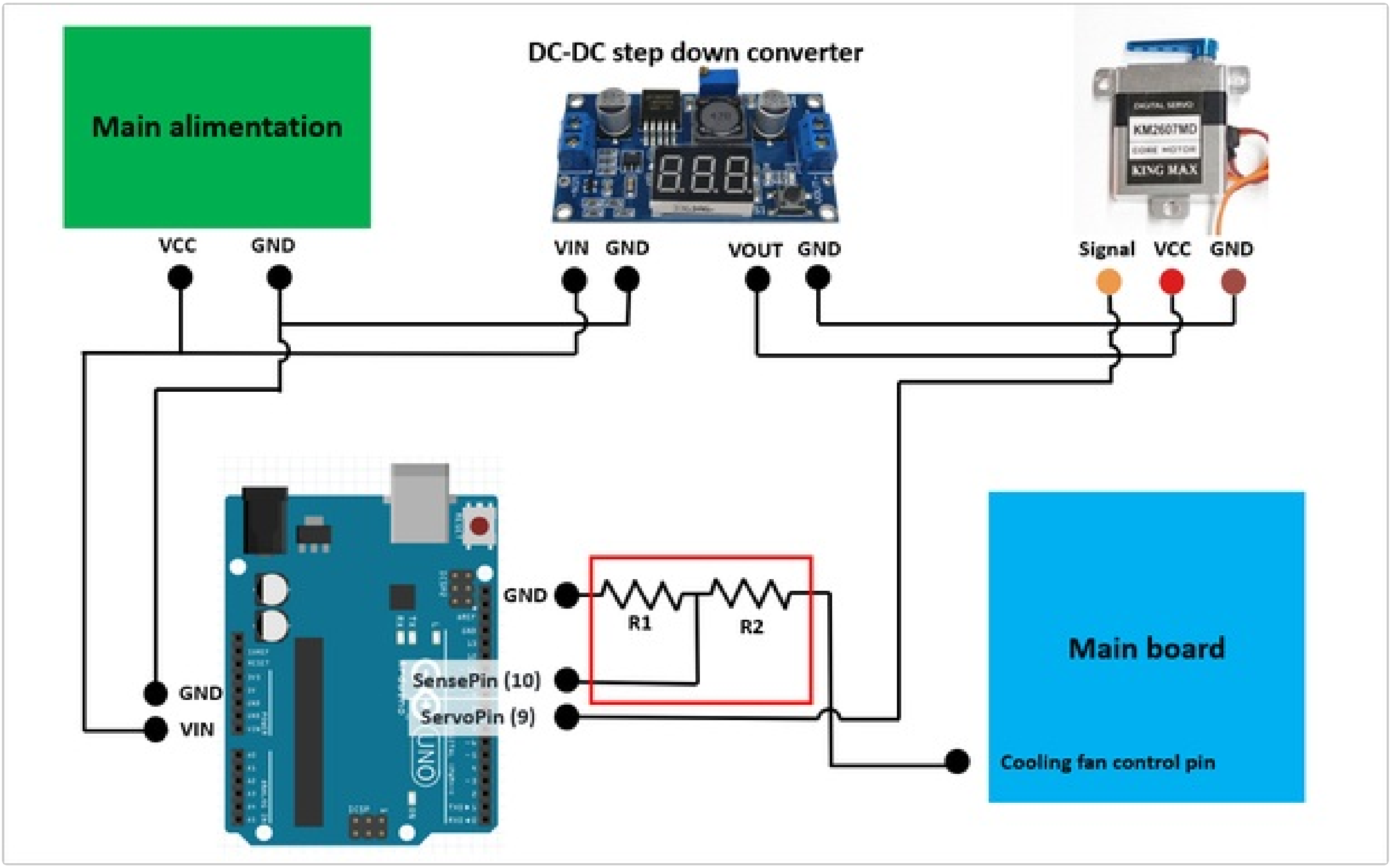
You need to test and record the position of each of those angles. For spraying you will use a 4th angle, A_spray, contained between A_valve and A_max. You can change the value of this angle to modify the intensity of the spray.


Position at Angle = A_max
Position at Angle = A_spray
Position at Angle = A_min
Position at Angle = 0

Step 7: Connecting the Arduino to the Motherboard

At that point you need to connect the arduino to the motherboard to detect and convert automatically the signal sent to the printer's cooling fan and use it to control the servomotor. The method i implemented uses the digitalRead() function of Arduino to detect the binary state (HIGH or LOW) of the cooling fan control pin and send a preprogrammed target angle to the servomotor. However the mainboard signal is 0/12V (or 24V depending or your model) and the arduino doesn't support inputs higher than 5V. In fact the arduino UNO will read an input between 0 to 1.5V as low and an input between 3 to 5V as high. In consequence it is necessary to transform the signal from the mainboard. As the Arduino is only reading a value and the pin 10 is defined as an input, it is in a high impedance state and doesn't draw current. For this reason we can lower the signal voltage with a simple voltage divisor bridge composed of two resistance R1 and R2. Folowing the connection diagram presented in this image, the tension on pin 10 V_{10} is linked to the tension at the cooling fan control pin V_{mb} by the equation: $V_{10} = R1/(R1+R2) \times V_{mb}$

In my case, with a 12V control signal i used R1=47kΩ and R2=100kΩ to obtain a signal around 3.8V compatible with the Arduino.



Step 8: Loading the Control Program and Test

When have connected the mainboard control pin and you know what angles to use to command your spray you can download the routine Control_spray_head.ino, change the angle variables to your personalized ones and load it to the Arduino.

At that point connect your computer to the printer main board and test the command "M106 S255" to turn on the cooling fan. The spray head should react by going to the A_spray position.

If this isn't the case it is debug time... Go over the precedent instructions until you understand and correct what went wrong and leave a comment if it was my fault...

Step 9: Generating a Control File

If your system is connected and correctly react to the commands, it is time to begin generating some control files. If you want to use this setup to make drawings, you may look into using laser cutter Gcode generation software and change the distance between the spray head and your sample to affect the line thickness. In my case, the objective was to produce uniform and homogeneous coatings on the surface of my samples and i wrote a python routine to generate the Gcodes. This routine is available here: Gcode generation routine.py. The wiki page associated to this code gives a lot of information on how to proceed: https://github.com/RemiRafael/Spray-coating-Gcode-generation/wiki/Gcode-generation-python-routine

If you are not familiar with the use of python scripts you can take a look here: https://www.instructables.com/Intro-to-Python/

To plainly use this code there are two things you need to measure: the offset off the spray head and the dimensions of the spray. To measure the offset, first go into the manual control of your printer. Move the head until the tip of the spray pen is located at the origin point of the printer. The coordinates of the printer are then directly the opposite of the tool offset value. To measure the diameter of the spray you can put a piece of paper on the printer bed and some ink or water in the pen. Use the printer manual mode to go at different positions over the paper (10 to 80 mm by 10mm steps for example) and spray for half a second of so. To do that you can use the command:

```
M106 S255

G4 P500

M106 S0
```

Measure the diameter of the sprayed area and record it. The type of ink, the nozzle and needle diameters in the spray pen and the air pressure can all influence this value.

Step 10: Spray and Enjoy

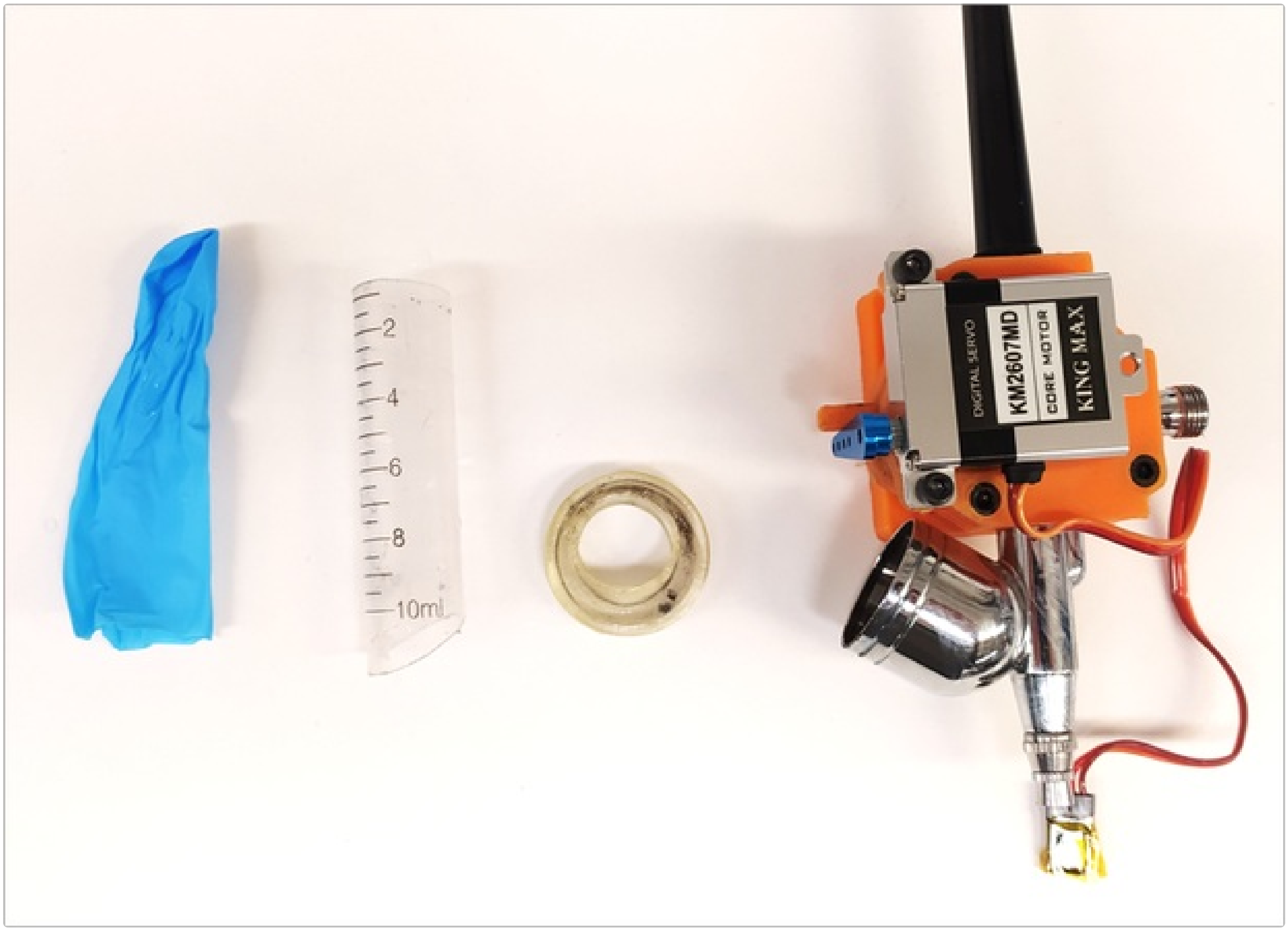
At that point you have done quite a lot of efforts and you spray coater works! Lay back and enjoy for a minute ✌

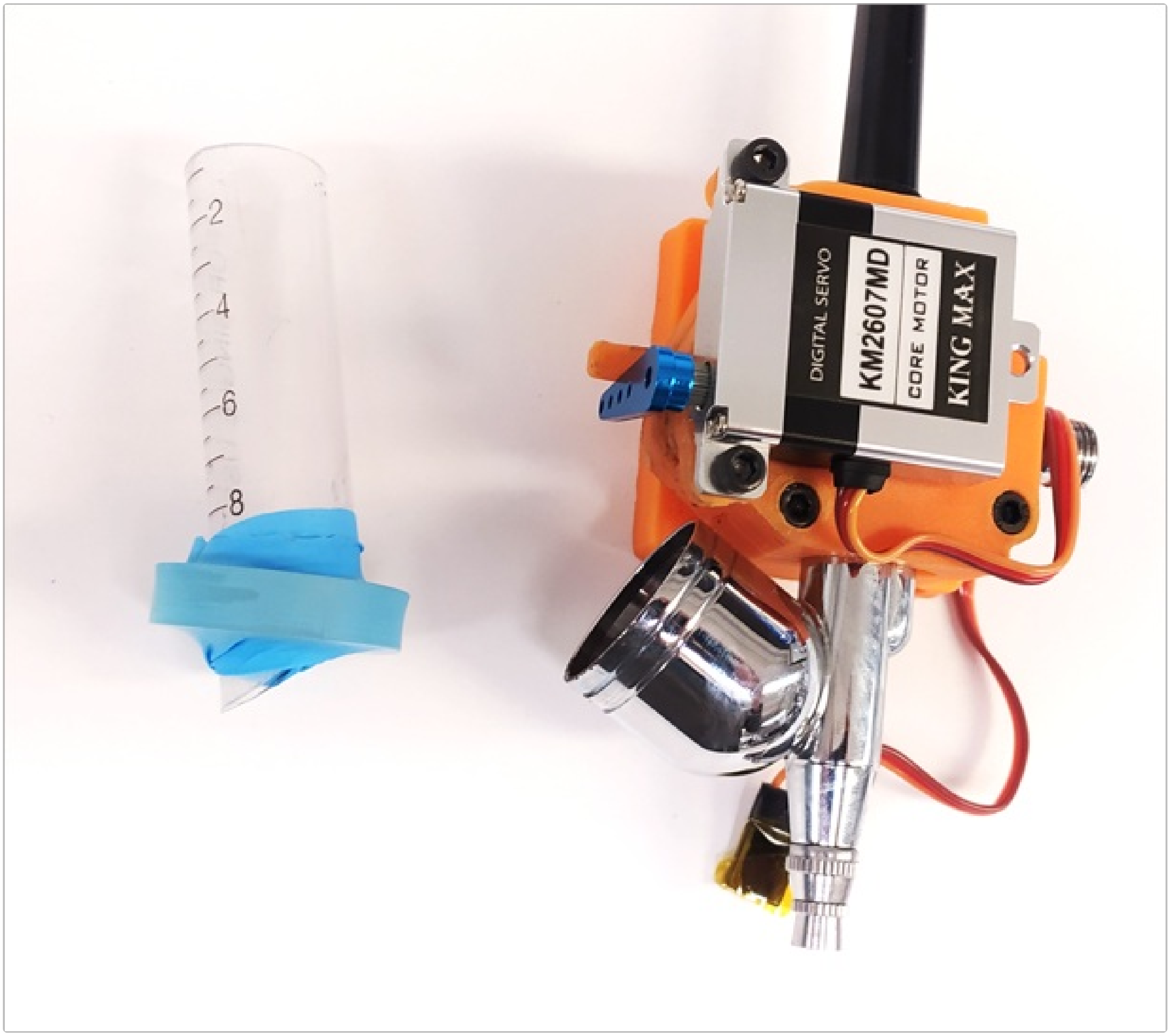
https://youtu.be/Ixd39U7TLSI

Step 11: Optional: Extend the Ink Tank

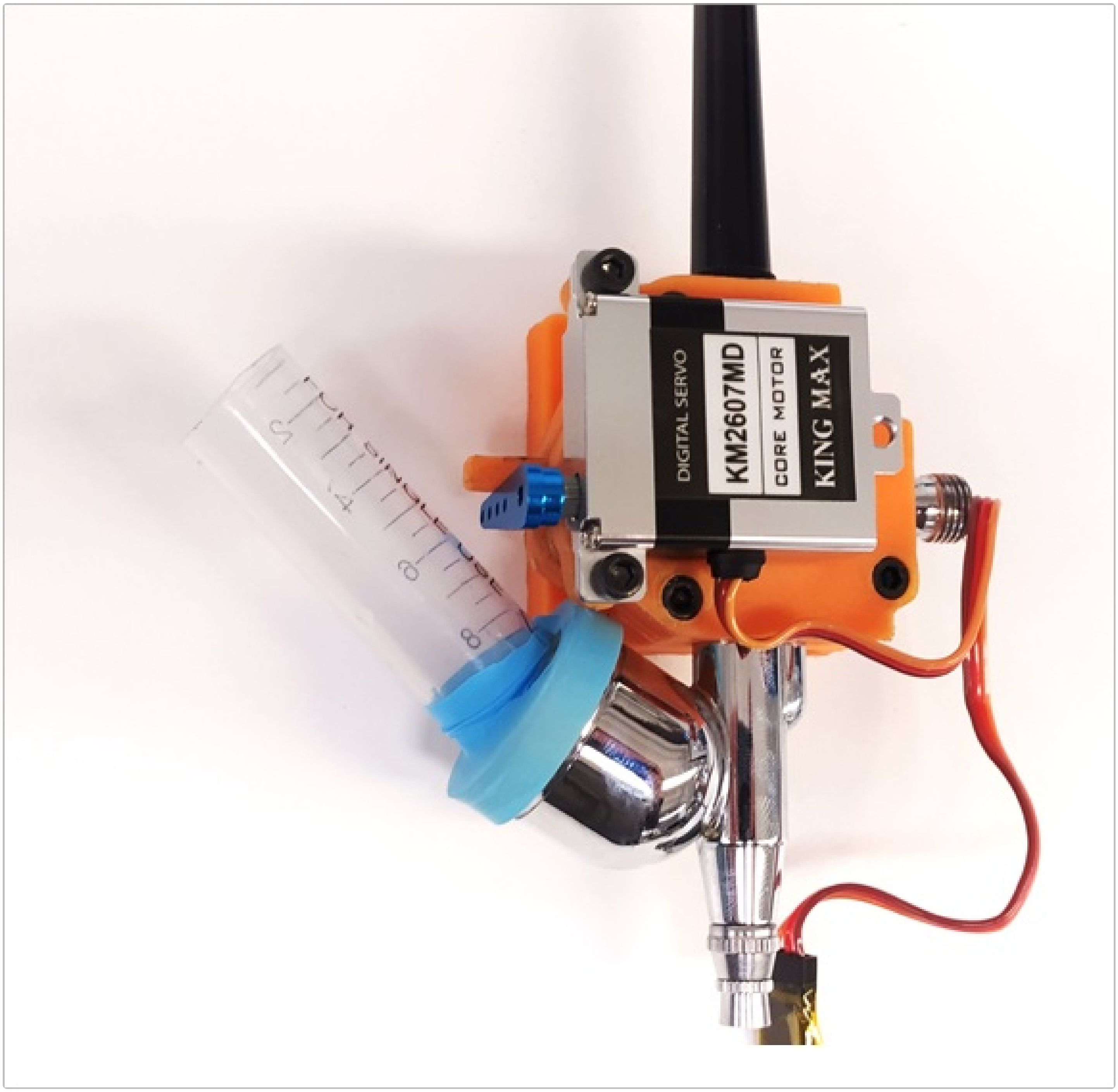
The ink tank of spray pens might be too small when you work with diluted inks and holing it vertically. To extend it you may use a 10ml syringe body and fix it to the tank opening according to those pictures. I designed a 3D printed part meant to be printed in TPU and hold the syringe in place but it can also be printed in PLA using the finger of a glove to ensure water-tightness.

3D model: https://www.thingiverse.com/thing:5668363


2
4
6
8
10ml
DIGITAL SERVO
KM2607MD
CORE MOTOR
KING MAX



2
4
6
8
DIGITAL SERVO
KM2607MD
CORE MOTOR
KING MAX



Step 12: Optional: Add a Stirring System

As is have said somewhere in this tutorial, i used this spraying setup to work with inks containing particles in suspensions. This type of inks can be prone to particle aggregation which tends to clog the nozzle and make the sprayed layer uneven. To prevent that I designed a stirring system to agitate the ink inside the reservoir. I initially used a small DC motor to drive a stirrer directly inside the reservoir but with some ink it creates foam and the inks spills everywhere. In the end I decided to use a syringe linked to a tube terminated by a nozzle. The nozzle is placed inside the reservoir and alternatively aspire and reject the ink to keep it agitated. The syringe is automatically driven by a stepper motor controlled via an Arduino and an pololu stepper driver. As this system is for a niche application (even more than a spray coater ^^) i will not go into much details but the CAD files are available here: and the arduino control routine here: Control_Strirrer_Stepper.ino

3D models: https://www.thingiverse.com/thing:5668363

https://youtu.be/er7cRqY1UCs