Tapster Robotics, Inc.

Quick Start with Tapster 3 *and* Tapster 3+

Thank you for your purchase of a **Tapster 3** or **3+ robot**

TAPSTER ROBOTICS:

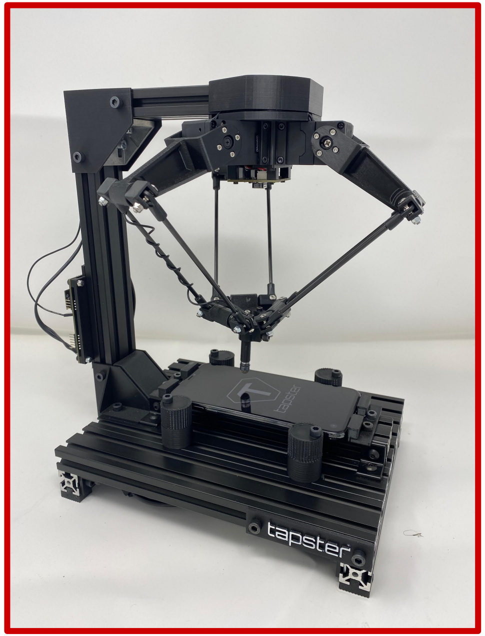
The intersection of software, electronic, and mechanical engineering

Tapster 3 and 3+ Robots

Tapster 3 and 3+ are delta robots that tap and swipe the touchscreen of a device just like a human finger does. They work and are programmed the same way.

Tapster 3 works with mobile phone-sized devices.

Tapster 3+ is large enough to be used with many tablet-sized devices as well as those the size of standard and plus-sized mobile phones.

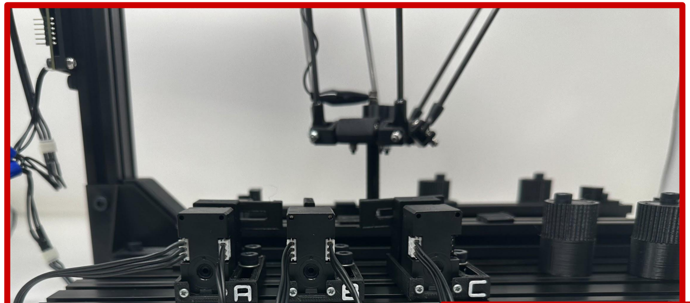


Tapster 3 shown in photo

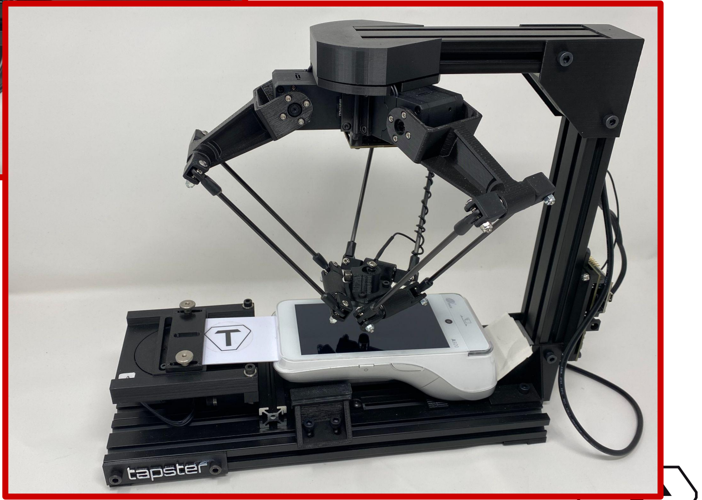
Add-on Modules

If your robot includes an add-on module like Push Button Module or Payment Pack, please refer to the **Addendum** for specific instructions for those modules.

Push Button Module (PBM)



Payment Pack



What’s in the box:

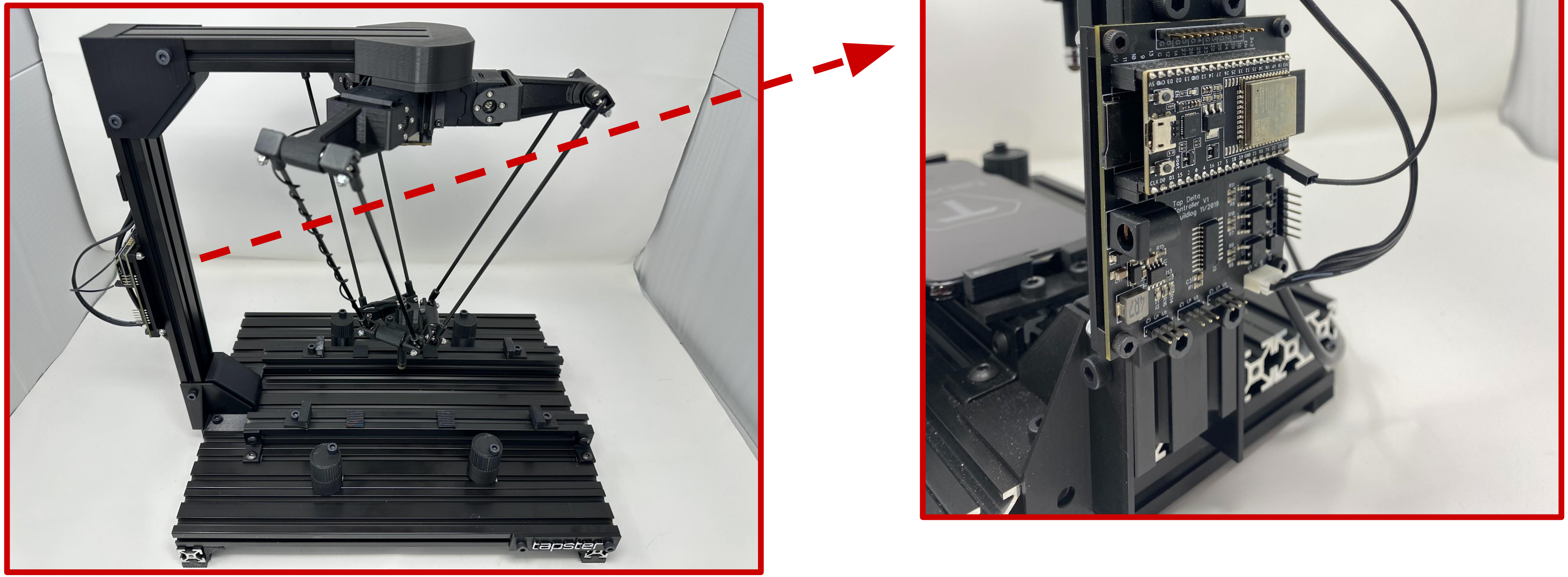
- Tapster 3 or 3+ robot
- 12V 5A Power Supply
- USB cable: A to micro B
- Hand tools: 4mm and 3mm hex drivers
- Cross check level
- Additional stylus tip, nuts and bolts
- *500mm cable included with add-ons*

Requirements:

- Computer running Mac, Windows, or Linux
- USB port

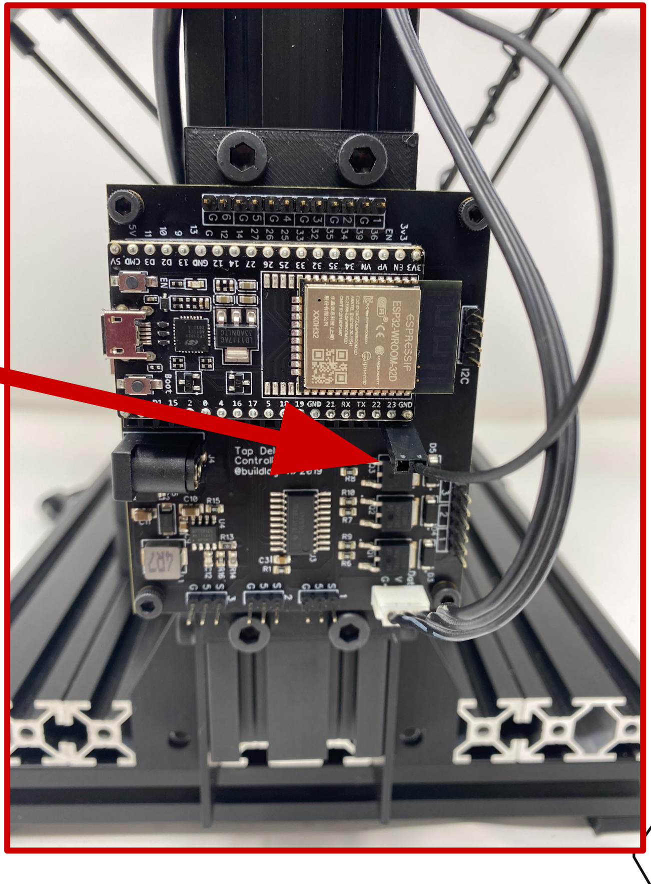
Electronics

The Tapster 3 and 3+ circuit board is attached to the back post of the robot.



Electronics (cont.)

1) Make sure the black wire is plugged into the GND header pin on the circuit board.

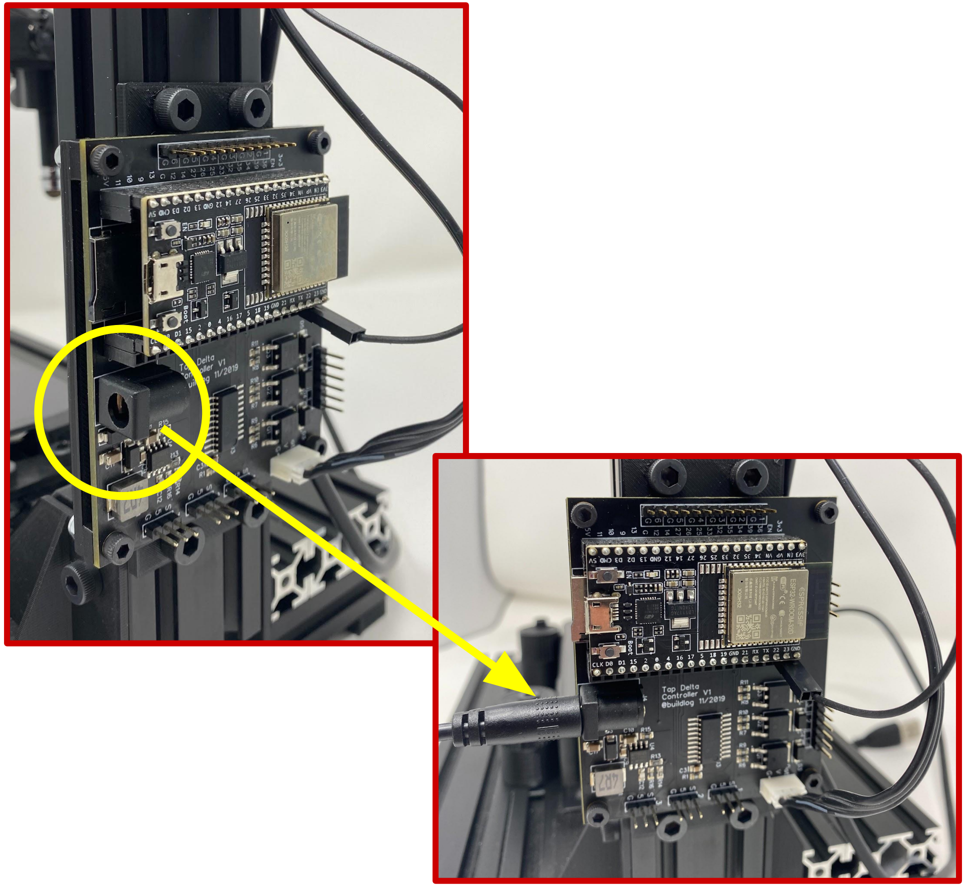


Electronics (cont.)

2) Insert the included 12V 5a power cable into the circuit board.

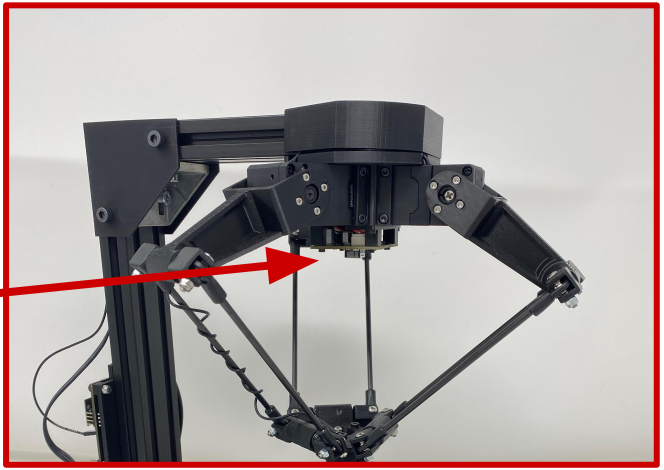
You *must connect power* to the circuit board *before* inserting the USB cable in Step 4.

3) Connect power cable to power supply.



USB Camera

The 16MP USB camera is pre-mounted and will connect to your computer through the integrated USB cable.

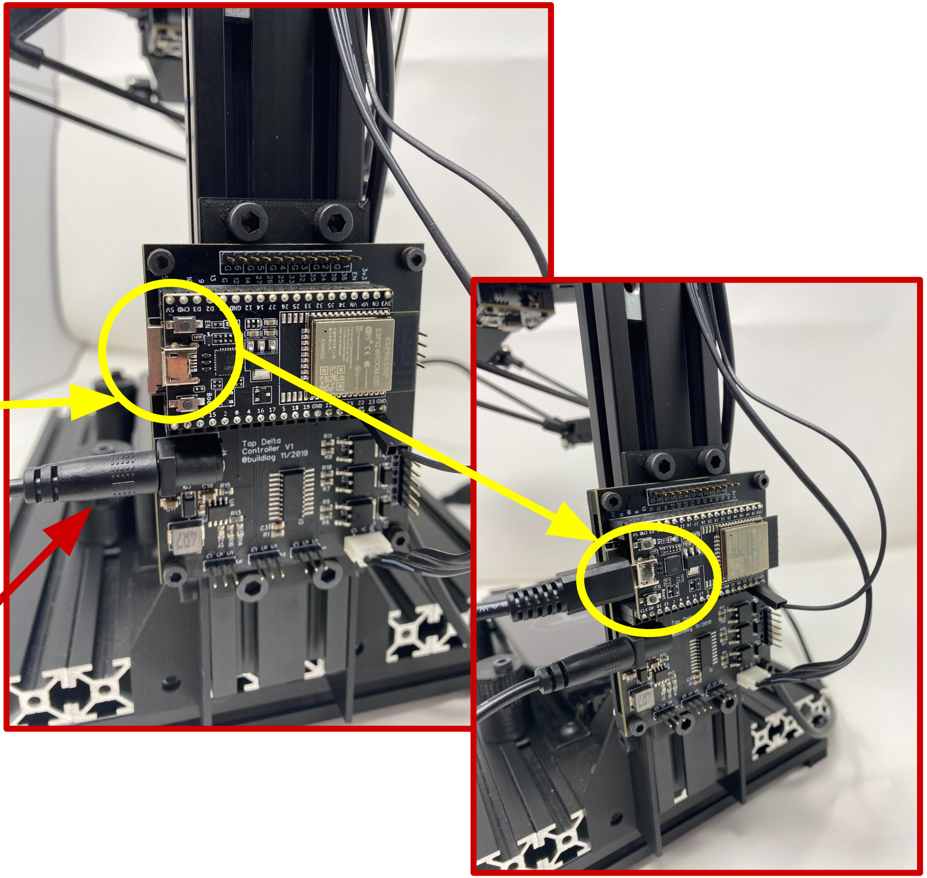


Electronics (cont.)

4) Confirm that you have connected power in Step 2.

Now, insert the micro B end of the included USB cable into the circuit board.

5) Insert other end of USB cable into your computer.



Connected power

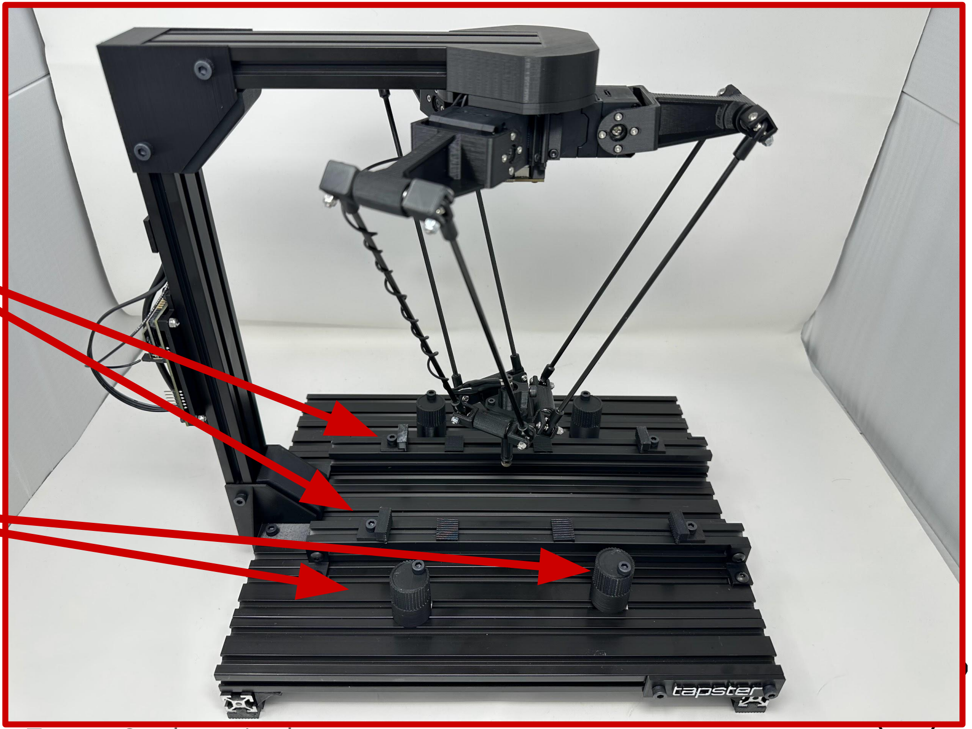
Insert your device into the base of the robot

Insert your device into the base of the robot.

Use the hand tools to adjust the four end brackets …

… and the four round side brackets …

Keep the brackets loose until you level the robot and device in the next step.



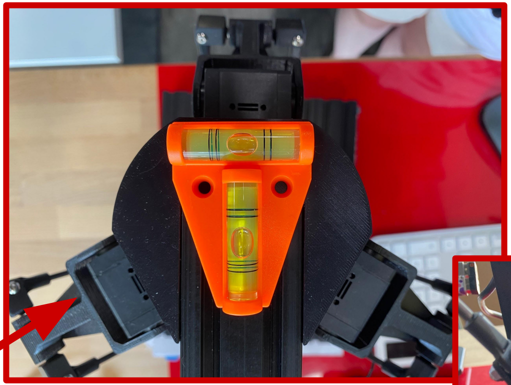
Tapster 3+ shown in photo

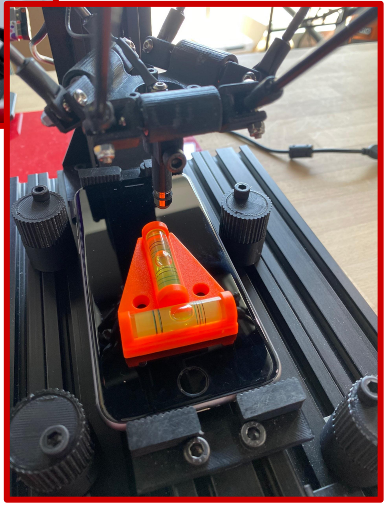
Level your robot and your device.

For best performance, it is important that your robot and the device are level to each other.

Use the cross check level included in the accessory box to level both the robot and the device.

- Level the robot by placing the cross check level on the top of the robot.
- Level the device by no placing the level on the surface of the device. Adjust the device as needed and secure with the round side brackets.





Software

Grbl for ESP32 firmware is what powers Tapster 3 and 3+. It is already pre-installed.:

https://github.com/bdring/Grbl_Esp32

https://github.com/bdring/Grbl_Esp32/wiki

Grbl for ESP32 (The specific fork running on Tapster 3 & 3+ for ESP32 microcontroller and Dynamixel servo support):

https://github.com/tapsterbot/Grbl_Esp32/tree/dynamixel_support

To **program Tapster 3 & 3+** from any language:

Doc: https://github.com/gnea/grbl/wiki/Grbl-v1.1-Interface#streaming-protocol-simple-send-response-recommended

Sample Code: https://github.com/gnea/grbl/blob/master/doc/script/simple_stream.py

Software (cont.)

Prerequisites:

1) Install Python from https://www.python.org/

2) Install pyserial library (https://pypi.org/project/pyserial/)

```
pip install pyserial
```

Software (cont.)

A minimal Python script to drive your robot:

```
import serial
import time

port = "/dev/cu.usbserial-1420"
serial = serial.Serial(port, 115200)
serial.write(b"\r\n\r\n")
time.sleep(2)                  # Wait for grbl to initialize
serial.flushInput()            # Flush startup text in serial input
serial.write(b"G0 X0 Y0 Z0\n") # Send G-code for moving to home position
serial.read_all()              # Wait for grbl response
```

Software (cont.)

To move the end effector, try these G-code commands:

Examples:

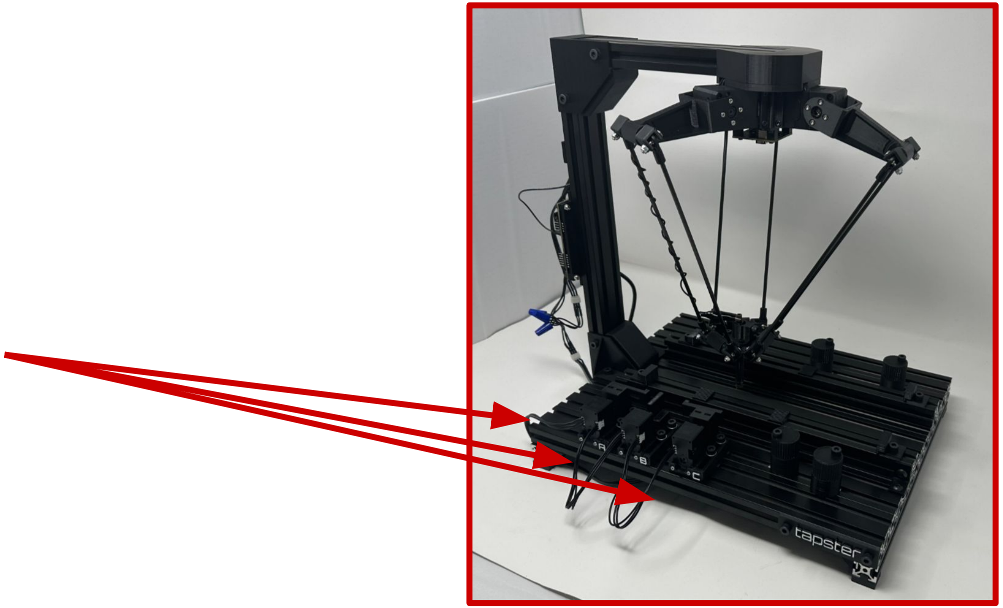
```
G0 X30
G0 X-30
G0 Y30
G0 Y-30
G0 X10 Y10
G0 X-10 Y-10
G0 Z0
G0 Z-20
G0 X-30 Y-30 Z-30
```

Addendum: Add-on Modules

tapster

Push Button Module v3

PBMv3 (Push Button Module) enables the robot to push the physical side buttons of a device.



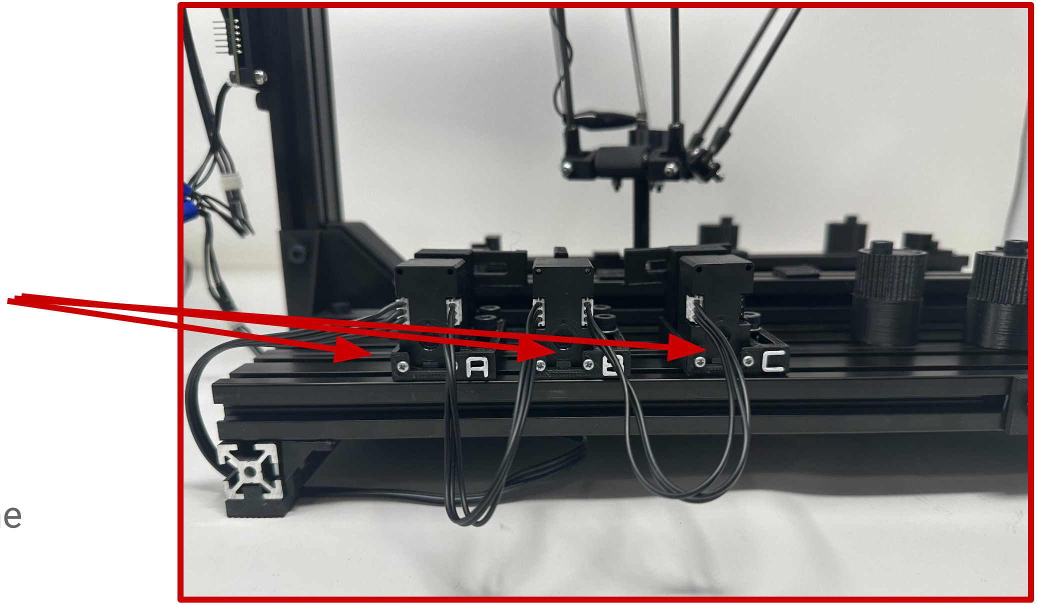
Push Button Module v3

Push Button Module v3 uses adjustable actuators (A,B,C) to push the physical buttons.

Your robot is currently set up to work with all three actuators on one side.

Use the enclosed hand tools to adjust the position of the actuators as needed. Use the extra 500mm cable to use actuators on both sides of the device.

Note: additional cable length is tucked between the actuators.



Push Button Module v3: G-code

For the side button actuators, valid values are anything from 100 to 2000, physically adjust the position of the servo mounting brackets with the included hex wrench drivers as needed to stay within this range.

To press the side buttons, try these G-code commands:

Power button - press:

```
G0 A100
```

Power button - release:

```
G0 A2000
```

Volume up button - press:

```
G0 B100
```

Volume up button - release:

```
G0 B2000
```

Volume down button - press:

```
G0 C100
```

Volume down button - release:

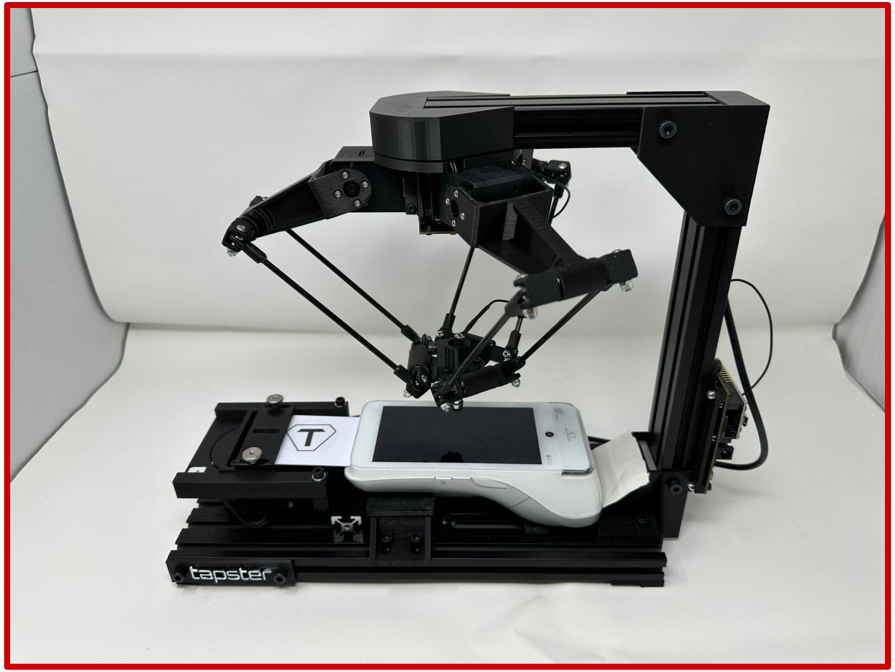
```
G0 2000
```

Note: The values (800, 2000, etc.) are not in millimeters, they are encoder positions specific to the servos.

Addendum: Payment Pack

Product Overview

The **Payment Pack** includes two card mechanisms that can each be used to insert a card with a chip or bring a card within range to trigger NFC.

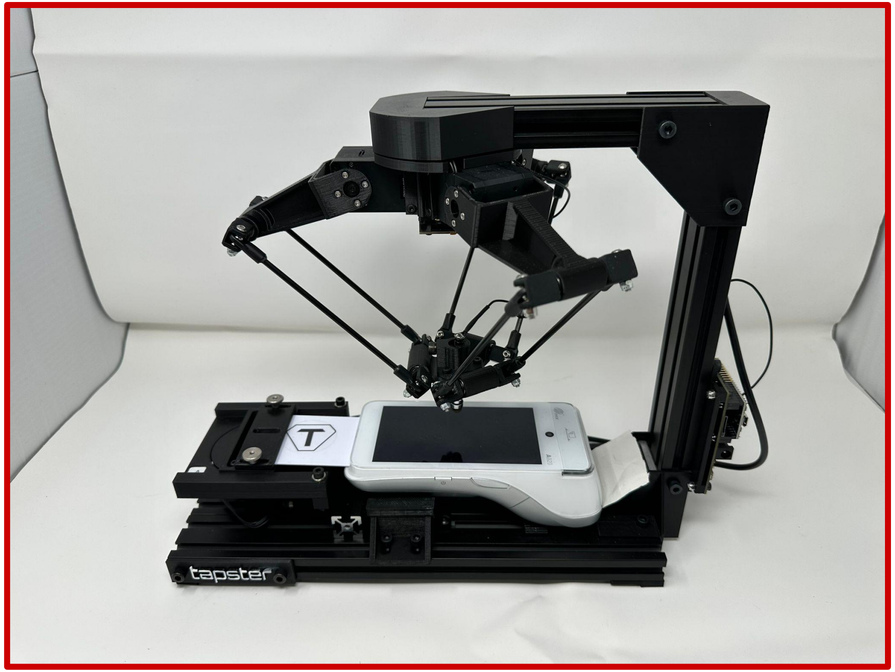


Addendum: Payment Pack

Product Overview

Your robot is currently set up to work with a PAX A920 payment terminal.

Use the enclosed drivers to adjust brackets as necessary. Use the extra 500mm cable when both card mechanisms are engaged.



Payment Pack: G-code

For the card insert actuators, valid values are anything from 100 to 2000, physically adjust the position of the mounting brackets with the included hex wrench drivers as needed to stay within this range.

Try these G-code commands:

Insert Card 1:

```
G0 A100
```

Release Card 1:

```
G0 A2000
```

Insert Card 2:

```
G0 B100
```

Release Card 2:

```
G0 B2000
```

Note: The values (800, 2000, etc.) are not in millimeters, they are encoder positions specific to the servos.

Thank you!

Keep in touch with
Team Tapster - we look
forward to hearing from you!

hello@tapster.io

www.tapster.io