Opentrons OT-2
Liquid Handler

Opentrons Labworks, Inc.

PUBLISHED: NOVEMBER 2022 VERSION 1.2

The information in this guide is subject to change without notice. Changes or modifications to the liquid handler not expressly approved by Opentrons Labworks, Inc., could void the user's authority to operate the equipment.

Failure to follow the warnings, precautions, and instructions for proper operation found in this manual will void the product's warranty. It may also result in injury to the operator and cause safety, reliability, and performance hazards within the Opentrons OT-2 system. The company does not assume responsibility.

Chapter 1 | Preface

1.1 About This Manual

The Opentrons OT-2 Liquid Handler User Manual contains important information regarding the safe use of the model OT-2R Liquid Handler. Read and understand the safety information thoroughly before operating the Opentrons Liquid Handler.

The Opentrons Liquid Handler OT-2R is referred to as the Opentrons Liquid Handler or the OT-2. For other Opentrons Liquid Handler models, please refer to the appropriate guide.

This chapter points out the purpose of the manual, specifies the product detailed within, and the audience the manual is intended for. It also explains the symbols, conventions, and abbreviations used and offers other general information.

1.2 Intended Audience

The intended audience for this guide is laboratory personnel. Researchers can use the Opentrons OT-2 Liquid Handler to transfer samples and reagents to conduct a wide range of laboratory experiments or assays. Opentrons OT-2 Liquid Handler operators should read this manual before using the instrument. Throughout this manual, "you" or "the user" refers to the Opentrons OT-2 Liquid Handler operator.

1.3 Safety Warnings and Precautions

The Opentrons OT-2 Liquid Handler has been designed for safe operation. Please refer to these specifications and compliance guidelines to ensure safe usage of the Opentrons OT-2 Liquid Handler. These include safe use specifications for all input and output connections for the product including the power supply, electrical input, and PC connection, and the warning labels bearing this information on the module:
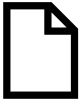
- Do NOT replace the IEC cord unless at the direction of Opentrons Support.

- AC IEC power connection to the power supply with earth grounding. Make sure that the outlet or circuit has earth/protective grounding.
- USB: User can connect Opentrons Modules to the USB port in the back of the liquid handler.
- Ethernet Network Connection: The user connects the OT-2 to a computer over Ethernet.
- Do not lift or move the OT-2 without the help of another person.

Safety Notation Marks

Warning labels posted on the Opentrons OT-2 Liquid Handler and in this manual warn you about sources of potential injury or harm. This manual uses the following symbols in the left margin to draw attention to the specified type of information.

Symbol	Meaning
	Warning - Warning alerts all users to the following. - Potentially hazardous conditions - Actions that may result in personal injury or death
	Caution - Cautions alert all users to the following - Damage to the equipment - Lost or corrupted data - Unrecoverable interruption of the operation being performed
	Electrical Shock - This symbol alerts users to instrument components that might pose risk of electrical shock if instrument is handled improperly.
	Hot Surface - This symbol identifies instrument components that pose a risk of personal injury due to excessively high heat temperature if handled improperly.

	Pinch Point - This symbol identifies instrument components which can pose risk of personal injury when moving.
	Note - Notes emphasize or expand upon the surrounding information.

Electrical Safety Warning

Total power consumption depends on the specifics of the protocols run on the robot. The Opentrons OT-2 Liquid Handler generally consumes around 90-120 W when idle. Power consumption may range from approximately 100 to 180 W when running a protocol. Exact power consumption is determined by the amount of movement executed during a protocol or the amount of time the robot spends idle. Power consumption may also be affected by the status of the lights on the Opentrons OT-2 Liquid Handler and how many pipettes are attached. The *power supply* maximum consumption is 220 W (6.1 A, 36 V).

Always observe the following electrical safety warnings:

⚠️ **Warning:** Plug the Opentrons OT-2 Liquid Handler into a grounded circuit Class 1.

📄 **Note**: If you are unable to insert the AC plug into your AC receptacle, contact an electrician to correct the situation.

⚠️ **Warning:** The Opentrons OT-2 Liquid Handler contains AC power fuses. If a fuse requires replacement, contact Opentrons support at *support@Opentrons.com.*

⚠️ **Warning**: Do not use AC power cords if the following conditions exist:
- The power cords are frayed or damaged.
- Other attached cords, cables, or receptacles are frayed or damaged.

The use of damaged power cords can cause an electric shock hazard resulting in serious injury or damage to the Opentrons OT-2 Liquid Handler.

⚠️ **Warning:** Do not connect (plug in) or disconnect (unplug) AC power cords if they are damaged. The use of damaged power cords can cause an electric shock hazard resulting in serious injury or damage to the Opentrons OT-2 Liquid Handler.

Additional Safety Warnings

⚠️ **Warning:** The Opentrons OT-2 Liquid Handler has not been certified for use with explosive or flammable liquids. Do not load plates, tubes, or vials containing explosive or flammable liquids into the OT-2 or otherwise operate the instrument with explosive or flammable liquids in the interior.

⚠️ **Warning:** Use good laboratory practices and follow the manufacturer's precautions when working with chemicals. Opentrons is not responsible or liable for any damages because of, or as a result of, the use of hazardous chemicals.

⚠️ **Warning:** The Opentrons OT-2 Liquid Handler weighs approximately 40 kg (90 lb). Moving or lifting the Opentrons OT-2 Liquid Handler incorrectly can result in injury. Never attempt to move or lift the Opentrons OT-2 Liquid Handler without using proper equipment and proper safety techniques. Contact the safety coordinator at your company or review the *Opentrons Getting Started Guide* for more information.

⚠️ **Warning:** The Opentrons Liquid Handler can emit vibration while in operation. For this reason, the OT-2 system should be placed on a surface that is sturdy, level, and water-resistant with cross-bracing or welded joints.

⚠️ **Warning**: The Opentrons Liquid Handler should be placed on a surface capable of supporting a minimum of 40 kg (90 lb) weight with sufficient surface area to accommodate the Opentrons Liquid Handler in the recommended operating envelope (see Opentrons Liquid Handler Dimensions).

Safety Cautions

To protect the Opentrons OT-2 Liquid Handler from damage, follow these precautions:

⚠️ **Caution:** Use labware that is ANSI/SLAS-compliant or approved by Opentrons.

⚠️ **Caution:** Keep corrosive materials, agents, or otherwise damaging materials away from the Opentrons OT-2 Liquid Handler.

Biological Safety

Specimens and reagents containing materials taken from humans should be treated as potentially infectious agents. Opentrons recommends using safe laboratory procedures as explained in *Biosafety in Microbiological and Biomedical Laboratories (BMBL) 6th Edition*. Under normal circumstances, the Opentrons OT-2 Liquid Handler does not create detectable aerosols from source liquids. However, under certain conditions, it is possible to generate aerosols from source liquids. When operating with BSL2 or above source liquids, precautions against aerosol exposure should be considered in accordance with your local regulatory bodies.

To minimize the potential risk of aerosol exposure from the Opentrons OT-2 Liquid Handler, it is important to ensure the following conditions are observed:

- Perform preventive maintenance as outlined by the *Opentrons OT-2 Preventive Maintenance Guide*.
- Ensure that all instrument covers, pipettes, modules, tubes, plates, and vials are installed and secured properly.
- Use proper pipetting technique to aid in the mitigation of aerosols.

Toxic Fumes

If the user is working with volatile solvents or toxic substances an efficient laboratory ventilation system should be used to remove any vapors that may be produced.

Flammable Liquids

The Opentrons robotic systems have not been evaluated for use with flammable liquids and should not be used with flammable liquids.

1.4 Product Labels

You will find the following labels on the Opentrons OT-2 Liquid Handler.

- Intellectual Property Label
- ETL Label
- Electrical Hazard Label
- General Warning Label
- WEEE Label
- Product Label
- Pinch Point Label
- High-Voltage Label
- Power and Fuse Rating Label

1.5 Compliance Standards

This Opentrons OT-2 Liquid Handler and its accessories have been tested and found to be in compliance with all applicable requirements of the following safety and electromagnetic standards:

- IEC/UL/CSA/EN 61010-1:2001 Safety requirements for electrical equipment for measurement, control, and laboratory use – Part 1: General requirements
- IEC/UL/CSA/EN 61010-2-010:2015 Ed.3 Safety Requirements For Electrical Equipment For Measurement, Control And Laboratory Use – Part 2-010: Particular Requirements For Laboratory Equipment For The Heating Of Materials
- IEC/UL/CSA/EN 61010-2-051:2015 Ed.3 Safety Requirements For Electrical Equipment For Measurement, Control And Laboratory Use – Part 2-051: Particular Requirements For Laboratory Equipment For Mixing And Stirring

Electromagnetic Compatibility (EMC)

- IEC61326-1:2013 Electrical Equipment for measurement, control, and laboratory use – EMC Requirements, Class A
- EN 61000-3-2:2014 Electromagnetic compatibility (EMC) - Part 3-2: Limits - Limits for harmonic current emissions
- EN 61000-3-3:2013
- KN 61000-6-3:2012
- EN 61000-3-2:2014
- IEC 61000-3-3:2013 Ed.3+A1
- IEC 61000-6-1*AEI
- EN 61000-3-3:2008
- FCC Part 15, Subpart B, Sections 15.107 and 15.109 as a Class A digital device
- IC ICES-003:2016Ed.6 Information Technology Equipment (Including Digital Apparatus) – Limits and Methods of Measurement

FCC Warnings and Notes

WARNING: Changes or modifications to this unit not expressly approved by Opentrons Labworks, Inc., could void the user's authority to operate the equipment.

- Note: This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.
- Note regarding FCC compliance: Although this instrument has been tested and found to comply with Part 15, Subpart B of the FCC Rules for a Class A digital device, please note that this compliance is voluntary, for the instrument qualifies as an "exempted device" under 47 CFR 15.103(c), regarding the cited FCC regulations in effect at the time of manufacture.

CISPR 11 CLASS A

WARNING: Class A equipment is intended for use in an industrial environment. In the documentation for the user, a statement shall be included drawing to the fact that there may be potential difficulties in ensuring electromagnetic compatibility in other environments, due to conducted as well as radiated disturbances.

Chapter 2 | Introduction

The Opentrons Liquid Handler Series includes the OT-1, OT-2, and OT-2R Liquid Handlers. Throughout this guide, the descriptions, illustrations, and procedures only apply to the OT-2R Liquid Handlers. Before using the Opentrons Liquid Handler system it is essential that you read this user manual carefully and pay particular attention to the safety information. The instructions and safety information in the user manual must be followed to ensure safe operation of the instrument and to maintain the instrument in a safe condition.

2.1 Opentrons Liquid Handler Specifications

Opentrons Liquid Handlers are designed to transfer 1-1000 microliter volumes of aqueous and DMSO-based solutions with accuracy and precision for a wide variety of applications.

2.2 Accuracy and Precision

To test the accuracy and precision of the Opentrons Liquid Handler prior to shipping, gravimetric testing is performed. GEN2 Pipettes are tested with the OT-2 Robot, Opentrons Tips, and Opentrons API. Opentrons does not include a pre-wet test for each individual aspirate of the pipette. A graphical breakdown of the process is seen in the figure below:

Gravimetric Calibration Test Process Diagram
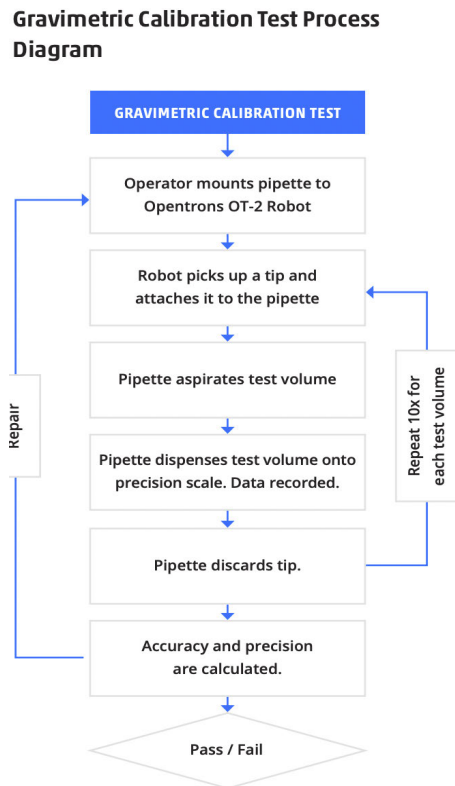
Accuracy and Precision of Opentrons Single-Channel GEN2 Pipettes

Opentrons Single-Channel GEN2 pipettes reliably conduct liquid transfers from 1 µL – 1000 µL. The transfer ranges available for each pipette are:

- P20 GEN2 (1 µL – 20 µL)
- P300 GEN2 (20 µL – 300 µL)
- P1000 GEN2 (100 µL – 1000 µL)

Table 2.2.1: Single-Channel GEN2 pipette accuracy and precision specifications.

MODEL	VOLUME	ACCURACY		PRECISION	
		% D	µL	% CV	µL
P20 GEN	1 µL	+/- 15%	0.15 µL	+/- 5 %	0.05 µL
	10 µL	+/- 2%	0.2 µL	+/- 1 %	0.1 µL
	20 µL	+/- 1.5%	0.3 µL	+/- 0.8 %	0.16 µL
P300 GEN2	20 µL	+/- 4%	0.8 µL	+/- 2.5%	0.5 µL
	150 µL	+/- 1%	1.5 µL	+/- 0.4%	0.6 µL
	300 µL	+/- 0.6%	1.8 µL	+/- 0.3%	0.9 µL
P1000 GEN2	100 µL	+/- 2%	2.0 µL	+/- 1%	1 µL
	500 µL	+/- 1%	5.0 µL	+/- 0.2%	1 µL
	1000 µL	+/- 0.7%	7.0 µL	+/- 0.15%	1.5 µL

Accuracy and Precision of Opentrons 8-Channel GEN2 Pipettes

Opentrons 8-Channel GEN2 pipettes reliably conduct liquid transfers from 1 µL – 300 µL. The transfer ranges available for each pipette are:

- P20 GEN2 (1 µL – 20 µL)
- P300 GEN2 (20 µL – 300 µL)

Table 2.2.2: 8-Channel GEN2 pipette accuracy and precision specifications

MODEL	VOLUME	ACCURACY		PRECISION	
		% D	µL	% CV	µL
	1 µL	+/- 20%	0.2 µL	+/- 10%	0.1 µL
P20 GEN2	10 µL	+/- 3%	0.3 µL	+/- 2%	0.2 µL
	20 µL	+/- 2.2%	0.44 µL	+/- 1.5%	0.3 µL
	20 µL	+/- 10%	2.0 µL	+/- 4%	0.8 µL
P300 GEN2	150 µL	+/- 2.5%	3.75 µL	+/- 0.8%	1.2 µL
	300 µL	+/- 1.5%	4.5 µL	+/- 0.5%	1.5 µL

Chapter 3 | Installation and Relocation

The Opentrons Liquid Handler and Opentrons Client Application are built to be installed by the end user. The instrument is shipped with all parts and components required for instrument installation and setup.

Questions about installation should be directed to *support@opentrons.com*, and send any questions about onsite installation to *sales@opentrons.com*.

This chapter covers pre-installation requirements, software installation, and OT-2 relocation.

3.1 Pre-Installation Requirements

Site Requirements

The following site requirements must be met before the OT-2 Liquid Handler can be installed:

Operating space and weight requirements	Sturdy, level, water-resistant benchtop capable of supporting 42 kg (93 lbs) with sufficient bench space to accommodate the Opentrons Liquid Handler in a minimum operating space of 63 cm x 57 cm x 66 cm / 25 in x 22.5 in x 26 in (width, depth, height) The OT-2 should be placed on a tabletop supported by four legs with cross-members. The table should be stationary, and we do not recommend using a table with wheels.
AC Power	Power Supply: 36V - 10A Input Power Supply: 100-240VAC, 50/60Hz
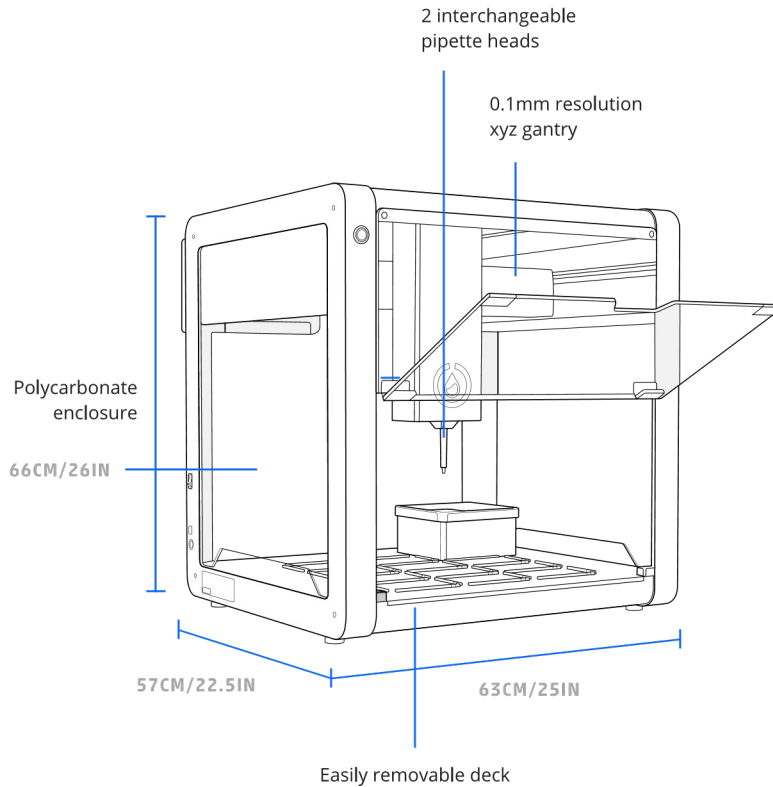
Figure 3.1.1: OT-2 minimum operating space

Operating Environment

Temperature and humidity recommendations for the OT-2

The OT-2 should be used in the recommended environment:
- Indoor use only
- Temperature: 20 °C ± 4° C (68 °F ± 7.2 °F)
- Relative humidity: Recommended 40-60% RH. Up to 80%
- Altitude: Up to 2,000 meters above sea level

Note: The OT-2 has been lifetime tested for up to three years of average use in these conditions. Opentrons defines the average use of the OT-2 to be four hours per day for five days per week.

Operating Outside the Recommended Range
Operating the instrument outside of the recommended operating environment is not recommended due to the following adverse possible effects:
- Pipettes may not perform optimally within their precision and accuracy ranges
- Low humidity can cause a buildup of static electricity causing tips to cling to pipettes when ejected.
- Operating in temperature and/or humidity extremes above or below manufacturer's recommendations can reduce the equipment's lifetime.

User Supplied Equipment Requirements

The user must provide the following items:

Client PC System Requirements	The Opentrons App for the OT-2 fully supports the following operating systems: Windows 10 or later, macOS 10.10 or later, Ubuntu 12.04 or later **Minimum Requirements:** 64-bit processor, 512 MB of RAM, 300 MB of free hard drive space for installation

Computer Configuration Recommendation
Opentrons recommends one computer per OT-2.

Operating with one computer per OT-2 is recommended for the following reasons:
- Operating the OT-2 with more than one computer per robot could slow down the overall process of running and testing samples on the system.
- Introduce confusion about which laptop contains the required protocol, labware, and/or other experimentation data files.
- It would limit Opentrons' ability to troubleshoot more advanced issues. The OT-2 requires a generated key to SSH into and keeping track of the keys would become very confusing.
- Having one dedicated computer per OT-2 simplifies the connection process and mitigates the risk of potential connection problems.
- If using a wired connection, it would be necessary to move the computer around the room, switch cables, and connect to each OT-2 separately.

The Opentrons App is optimized to work with one OT-2 at a time. The following scenarios can occur is switching between Opentrons Apps on different client PCs:
- The OT-2 could lose its place in the calibration flow when switching between the different computers.
- Even if the computer has multiple USB ports, there can be weird routing issues that can lead to the inability to connect to the different OT-2s.

If it is necessary to operate multiple OT-2s with one computer:
1. Connect the computer to the first OT-2.
2. Upload your protocol.
3. Calibrate required pipette and labware.
4. Run the protocol.
5. Disconnect from the first OT-2.
6. Connect the computer to the second OT-2. Once connected to the second OT-2, flick the deck lights to verify you are connected to the correct OT-2. Repeat steps 2-4.
7. Repeat steps 5 and 6 for remaining OT-2s.

The Opentrons OT-2 Liquid Handler Components
The following components are included with the Opentrons Liquid Handler:

Number of Components	Name of Component	Description
1	OT-2	Opentrons Liquid Handling robot
1	Top Window Cover	Polycarbonate window. This component is shipped in the same box as the OT-2 and must be placed on the OT-2 during initial robot setup.
4	Front, Side, & Rear Window Cover	Polycarbonate window. These window covers are pre-assembled and installed on the OT-2 prior to shipping.
1	Power Supply	Power Supply (36V/6A)
1	Regional IEC Power Cable	Connects the power supply bar to the type of wall socket used in your region. Currently supported regions are the United States, United Kingdom, European Union (outside Italy), and Italy.
1	Ethernet Cable	Connects the OT-2 to a client PC
1	Ethernet-to-USB dongle	USB adapter for computers without an Ethernet port
1	Hand Hex Screwdriver	2.5 mm hand screwdriver for attaching and removing pipettes
1	Wrench	14 mm wrench for adjustable OT-2R feet
1	Torx Key	Torx Key (T10)
1	Allen Key Set	Allen key set (1.5 mm, 2 mm, 2.5 mm, 3 mm)
1	Super Lube	Silicone-based lubricant
2	Hex Nut	M3 Hex Nut
2	Square Nut	M4 square nut
4	Pipette Screws	4 pieces of M3x6 mm socket head screws for pipette attachment

1	Trash Bin	Black polycarbonate plastic bin.
1	Calibration Block	The Calibration Block is a tool used to calibrate the length of pipette tips and to level 8-Channel pipettes. It provides a flat surface parallel to the OT-2's deck that can be reached by Opentrons pipettes.

3.2 Installation Overview

The OT-2 Liquid Handler is installed by the user. The following is an overview of the hardware and software installations that will need to be performed to set up the OT-2 Liquid Handler.

Unbox the OT-2 Liquid Handler

1. Unpack the OT-2 Liquid Handler by cutting open the shipping box that the OT-2 Liquid Handler comes in.
2. Remove the box lid.
3. Remove the top layer of foam. This will reveal the top window and the rear window cover; both will be taped to the top foam. Place both to the side (the window and rear window cover will be installed in a later step).
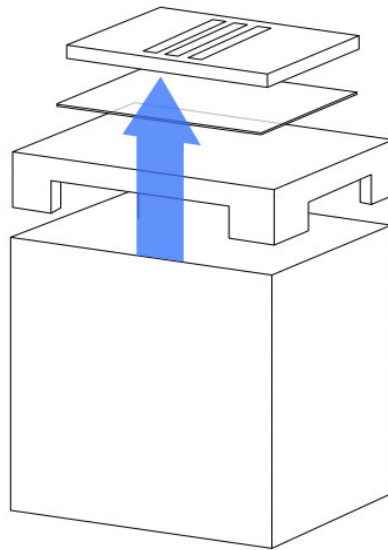
Remove the top layer of foam from the box as indicated above and place the window covers carefully to the side.

4. Remove the larger white foam enclosure.
5. Unwrap the protective plastic enclosure.
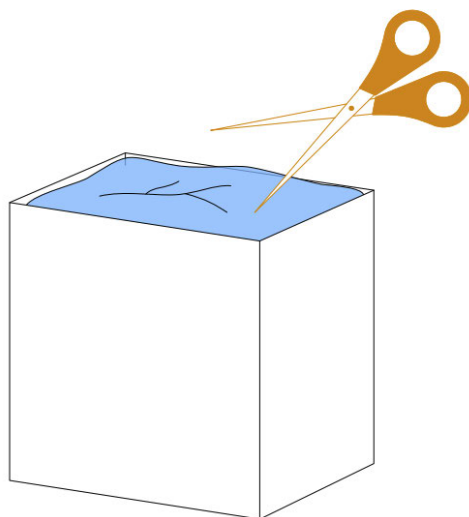
Using scissors, carefully cut open the VCI bag encasing the OT-2. If using a cutting tool, be careful not to accidentally damage the OT-2.

6. Once the foam and plastic layer is removed, with the help of two people carefully lift the OT-2 by its frame and remove the OT-2 from the box. Place the OT-2 in an area that meets the minimum operating space requirement and place the machine on a secure and stable surface.
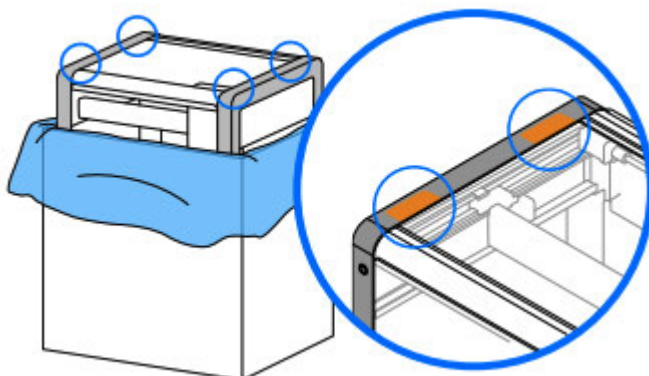
To remove the OT-2 from the box that the OT-2 was shipped in, lift the OT-2 by its frame in the locations indicated in orange.

7. Open the robot's door to cut away and remove the bubble wrap.

     Note: Be careful not to scratch the doors or panels with your scissors.

8. The bubble wrap will also contain the window covers for the side windows of the OT-2. Unwrap the window covers and place the covers to the side, for use in a later step.

9. Remove the protective film from all the windows.

Prepare the OT-2 Liquid Handler

Remove the shipping ties from the OT-2's deck by following these instructions:

1. During shipment, zip ties hold the accessory kit to the OT-2 deck. Cut the zip ties fastening the accessory kit to the deck of the OT-2. Remove and open the accessory kit.
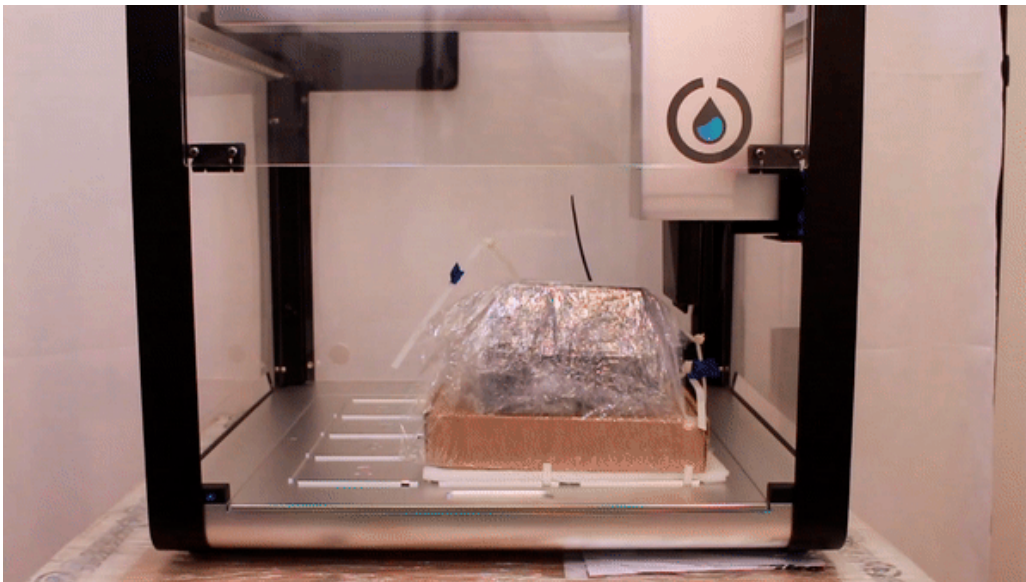
Image of Accessory Kit zip tied to OT-2 deck.

2. Remove the zip ties from the deck by using the 2.5 mm hex screwdriver to remove the two screws fastening the deck. **Don't discard these screws.** Lift the deck out to remove the zip ties. Then replace and refasten the deck with the two screws.
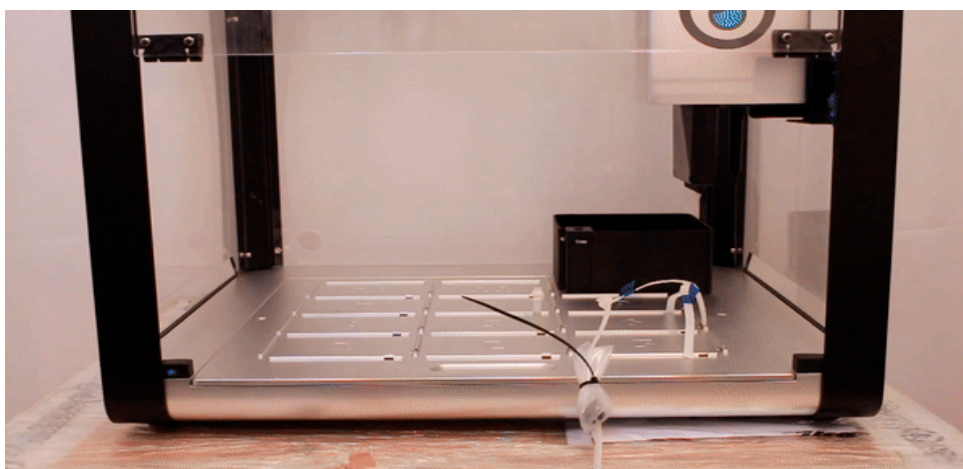
Image of zip ties securing the OT-2 deck.

Unlock the OT-2 Liquid Handler

To prevent damage during shipping, the moving parts of the OT-2 are locked in place with three *locking brackets*. Before turning on the OT-2 for the first time, it is necessary to

remove the brackets. The specific hardware is easily identifiable: each screw and bracket is flagged with blue tape. The brackets themselves may be black or red.

⚠️ **Warning:** Do not plug in or power on your robot. You will be prompted to do so at a later step in the manual.

1. Turn the robot around so the red flags on the locking hardware are accessible.

2. The first two screws to remove can be found behind the pipette carriage. The screws can be accessed by opening the front door. Use the 3 mm hex key to remove these screws.
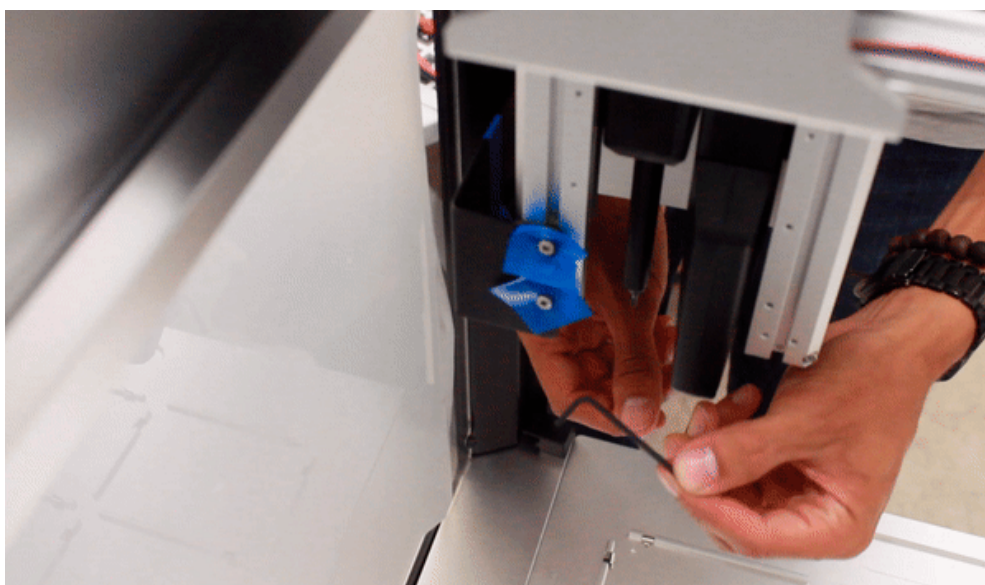
Image of first two screws removed using the 3mm hex key.

3. Collect the two square nuts that fell out of the pipette carriage.

📄 **Note:** If no nuts fell out, the nuts are stuck inside the channel. It is recommended that the user remove the nuts now to eliminate the risk of the nuts being shaken loose and falling out during a protocol run. To remove the stuck nuts:

   a. Use the 2 mm hex key to loosen the small screw inside the pipette enclosure so the nuts have more room to slide past it.
   b. If the nuts still haven't fallen out, stick your hex key up the channel and dislodge the stuck nuts.
   c. The nuts will now become dislodged.
   d. When the nuts have fallen out, retighten the small screw.

If trouble arises when removing the two square nuts, contact Opentrons Support for help.
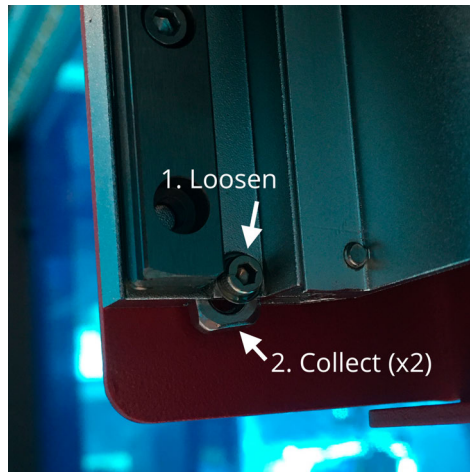
Two nuts to be collected from the pipette enclosure.

4. The next set of screws are located slightly above the pipette carriage. There are three screws in total. To access the third screw, the two screws shown below must first be removed. Use the 3 mm hex key to remove these screws.
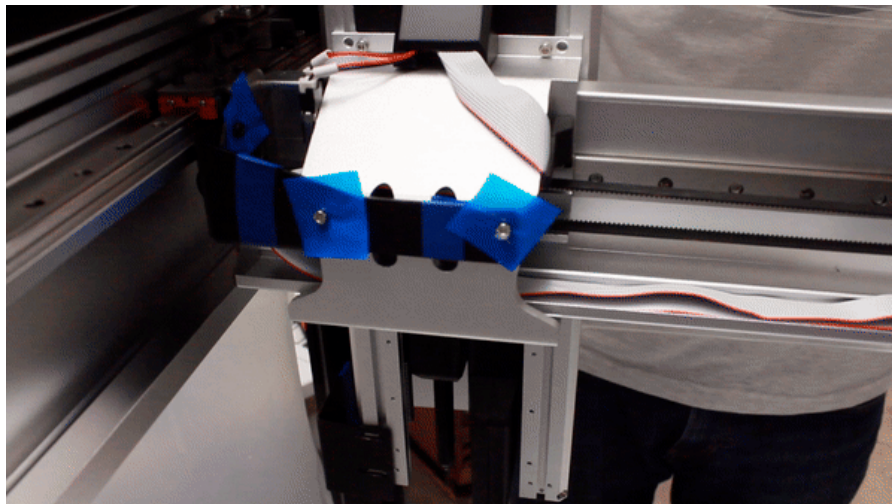
Two screws located above the pipette carriage that must be removed to access the third screw.

5. Move the Z carriage over to allow better access to the third locking screw.
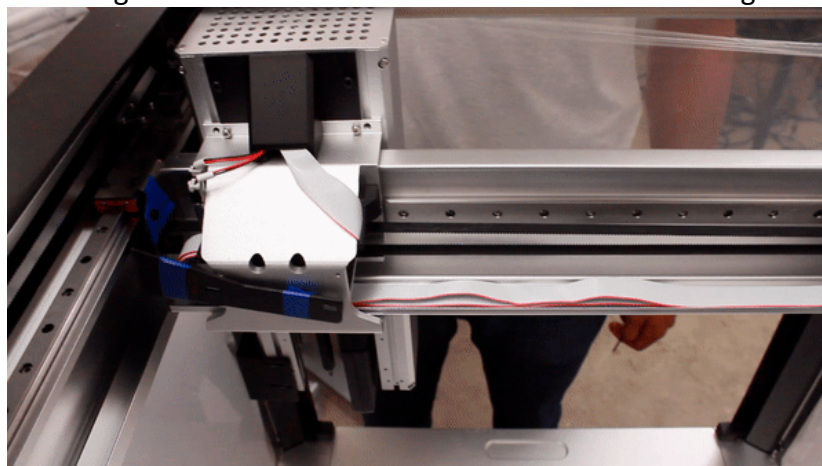
6. Unscrew the final locking screw to remove the bracket.
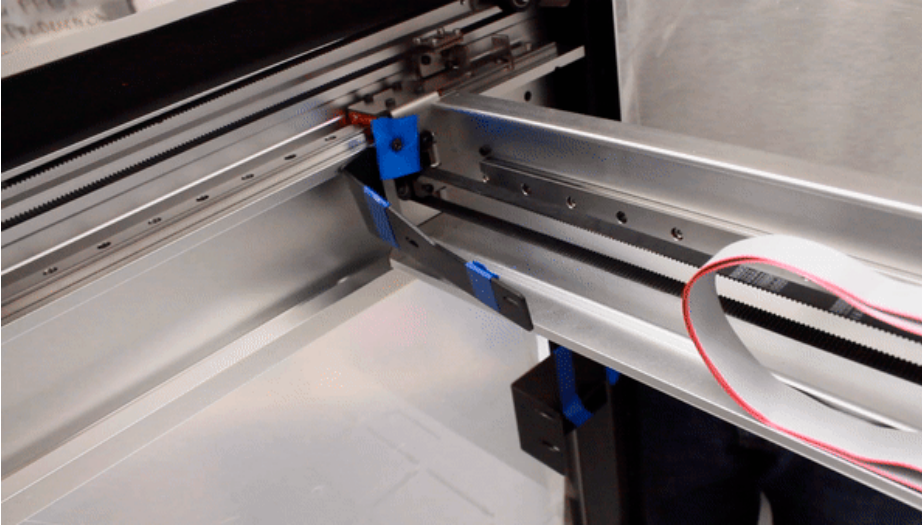
7. Now that the carriage is out of the way, the two screws for the bottom locking bracket are accessible. Using the 3 mm hex key to remove the screws.
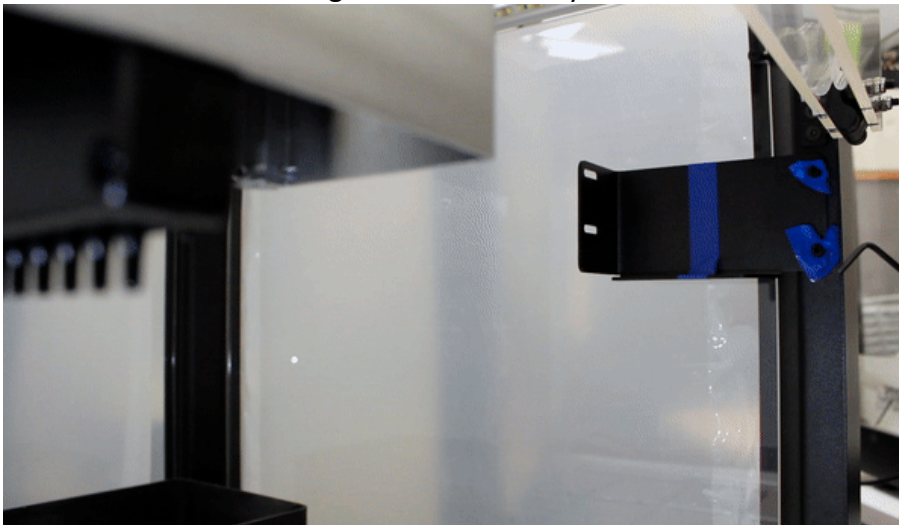
8. The last locking bracket is the smallest and easiest to remove. Using the 3 mm hex key, remove the screw and small L-shaped locking bracket.
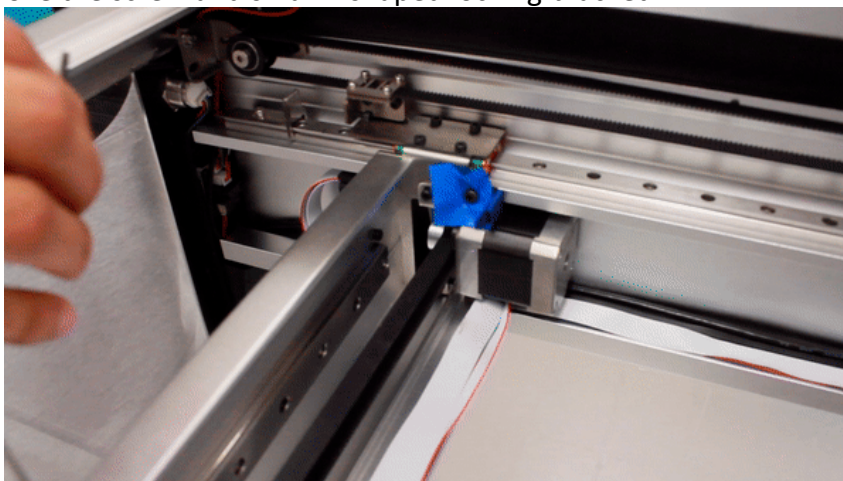
Install the top window and the window covers

The OT-2 Liquid Handler has:
- A removable top window
- Removable window covers for the holes in the left, right, and rear windows

Installing these is optional, but the window covers can help stop cross-contamination from entering and leaving the OT-2 Liquid Handler.
1. Slide the window onto the top of the OT-2 Liquid Handler.
2. Open the plastic bag for the top window screws.
3. If the screws don't already have washers and O-rings, insert them.
4. Screw the four screws into the corners of the top window.
5. Install the window covers. The left and right-side windows of the OT-2 have holes at the bottom.
6. Use the included *side window covers* and *rear window cover* to cover them up. These were set aside during the initial unboxing of the OT-2.
7. The window covers snap on magnetically. Attach them to the inside of the OT-2.

Note: Save the box and foam to reuse for OT-2 relocations.

Power on the OT-2 Liquid Handler
Connect the power supply to the wall. When the power supply is successfully receiving power, an LED light on the power brick will indicate it is receiving power correctly by lighting up blue.
When connecting the power supply to the robot, make sure it is in the correct orientation. Connect the power supply brick to the OT-2 using the correct orientation for the power supply connector shown below: the flat part of the power cord should be aligned with the flat part of the sheet metal on the power port.

To plug in, match the flat side of the power cord with the flat side of the power port.
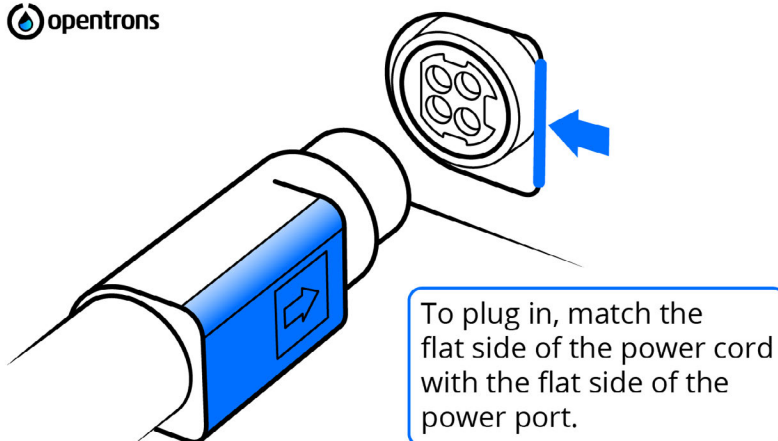
Connect the power supply brick to the OT-2 in the orientation shown above.

Locate the power supply connector on the rear left-hand side and power on using the power switch.
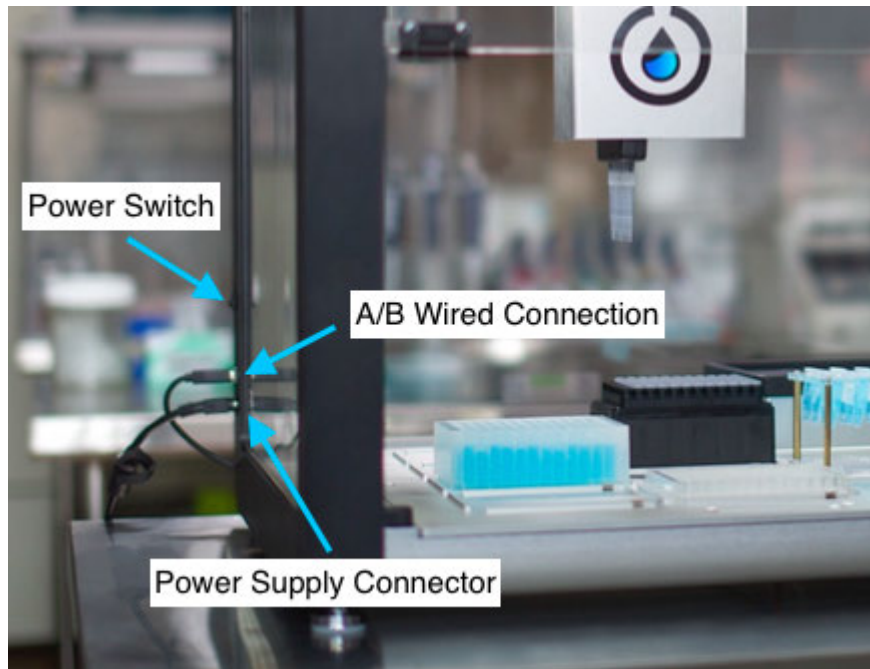
Image displays the location of the power supply port, power switch, and A/B wire connection port on the OT-2.

What to expect when powering on the OT-2

It takes several minutes for the robot to completely power on before you can connect to it. You'll know the robot has powered on successfully after:
1. The blue light on the front of the robot will blink.
2. Then the blue light will turn solid.
3. The robot should then emit audible mechanical noises as the pipette mounts home.

Note: If the blue light on the front of the robot does not turn solid after the robot has been powered on for more than five minutes, contact Opentrons Support for help.

Connect to the OT-2 Liquid Handler by Ethernet
The OT-2 directly connects to the user's preferred computer system via Ethernet cable. The Ethernet port on the outside of the robot connects directly to the Raspberry Pi's Ethernet port internally.
1. Locate the Ethernet port. There are four panels that enclose the OT-2: front, left, back, and right. If you are facing the front panel of the OT-2 (with the swinging door), the Ethernet port is located on the left side panel, in the lower left, below the power switch. In the image below, note the orientation of the OT-2. Ensure you are checking the left panel of your robot:
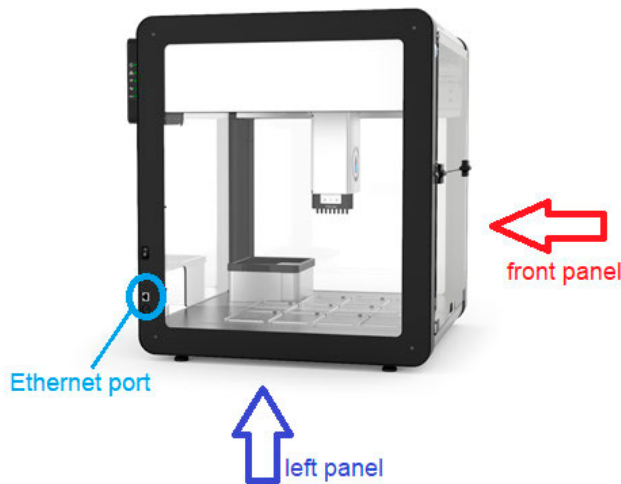
Use the image above to locate the Ethernet port on the OT-2.

2. Plug in the Ethernet cable. Use that to connect the OT-2 to the client PC (client workstation) or router.

Connect to the OT-2 using the Supplied Ethernet-to-USB Dongle

The OT-2 has an Ethernet port on the side. The provided Ethernet cable is used to connect the OT-2 to the client PC. Use the provided Ethernet-to-USB dongle if the computer that will be used with the OT-2 does not have an Ethernet port.

To use the Ethernet-to-USB dongle, connect the Ethernet cord into the robot. Connect the Ethernet cord into the dongle's Ethernet jack. Then attach the USB plug into the computer's USB port.

Connect to the OT-2 Liquid Handler over Wi-Fi

By connecting the OT-2 to Wi-Fi a user can:
- Run protocols wirelessly, from anywhere – as long as the client PC is on the same network as the OT-2.
- Enable Opentrons Support to diagnose and fix problems more efficiently remotely.

Note: Before connecting, read *Network Requirements for the OT-2* to ensure that the networking environment is properly configured. Ensure you are running the latest version of the Opentrons App and that your robot is fully up to date. Only change the Wi-Fi network of your robot while connected via USB or Ethernet. Otherwise, you'll lose connection to your robot when it leaves to join the new network.

Connect to a WPA2 Personal network

WPA2 Personal is a standard Wi-Fi network that requires a password. To connect to a WPA2 Personal network:

1. Go to **Robot Settings > Networking > Wi-Fi.**
2. Select your network from the dropdown list.
3. Select WPA2 Personal from the Authentication dropdown.
4. Enter the network's password.
5. Click **Connect.** The OT-2 will then proceed to join the chosen network.

Connect to a WPA2 Enterprise network
A WPA2 Enterprise network (also known as 802.1x; "eduroam" is a common example) is a Wi-Fi network with extra authentication mechanisms for extra security. The credentials the user will need to provide depends on how the network has been set up. If unsure, consult your IT team.

To connect to a WPA2 Enterprise network:
1. Go to **Robot Settings > Networking > Wi-Fi.**
2. Select your network from the dropdown list.
3. Select the network's authentication method, provided by your IT team.
4. Enter the correct user credentials, provided by your IT team.
5. Click **Join.** Your OT-2 will attempt to join the network.

Connect to a captive portal network

A *captive portal network* is the kind of network deployed at airports, hotels, and other public access points. A captive portal will present the user with the opportunity to connect without requiring a password, but when connected to it, the user will be shown a web page prompting the user to log in. The OT-2 can't use captive portal networks. Ask your local network administrator if there's a different network the OT-2 can connect to, or just use USB or Ethernet connection to connect to the OT-2.

Connect to an open Wi-Fi network

The OT-2 can be connected to a Wi-Fi network without any security, but this is not recommended by Opentrons.
1. Go to **Robot Settings > Networking > Wi-Fi.**
2. Select your network from the dropdown list.
3. Select None from the Authentication dropdown.
4. Your OT-2 will attempt to join the selected network right away.

3.3 Software Installation

Minimum System Requirements
To ensure the best experience while using the Opentrons Client Application, we recommend you use a system with at least the following resources available:
- A 64-bit processor
- 512 MB of RAM
- 300 MB of free hard drive space for installation

OS	Processor	RAM	Hard Drive Storage
Windows	64-bit	2 GB	17 GB
macOS	64-bit	2 GB	9 GB
Linux	64-bit	512 MB	6 GB

Supported operating systems for the Opentrons Client Application
The Opentrons Client Application for the OT-2 fully supports the following operating systems:
- Windows 10 or later
- macOS 10.10 or later
- Ubuntu 12.04 or later

Set up the Opentrons Client Application (Opentrons App)
Install the Opentrons Client Application (Opentrons App) onto the customer-supplied PC (client workstation).

To install, go to the Opentrons Client Application download page and follow the OS-specific instructions below.

macOS

The Opentrons Client Application for macOS is distributed as a DMG (disk image) file. To install the app:
1. Click the **Mac** download button on the download page.
2. Open the downloaded file named **Opentrons-...dmg** in your Downloads folder. The disk image will mount, and a new window will open.
3. In the new window, drag the **Opentrons.app** file into the **Applications** folder.
4. Open the **Applications** folder.
5. Double-click the **Opentrons** app to run it. Click **Open** or **Allow** on any security or firewall prompts to ensure the Opentrons App can run and communicate with the OT-2.
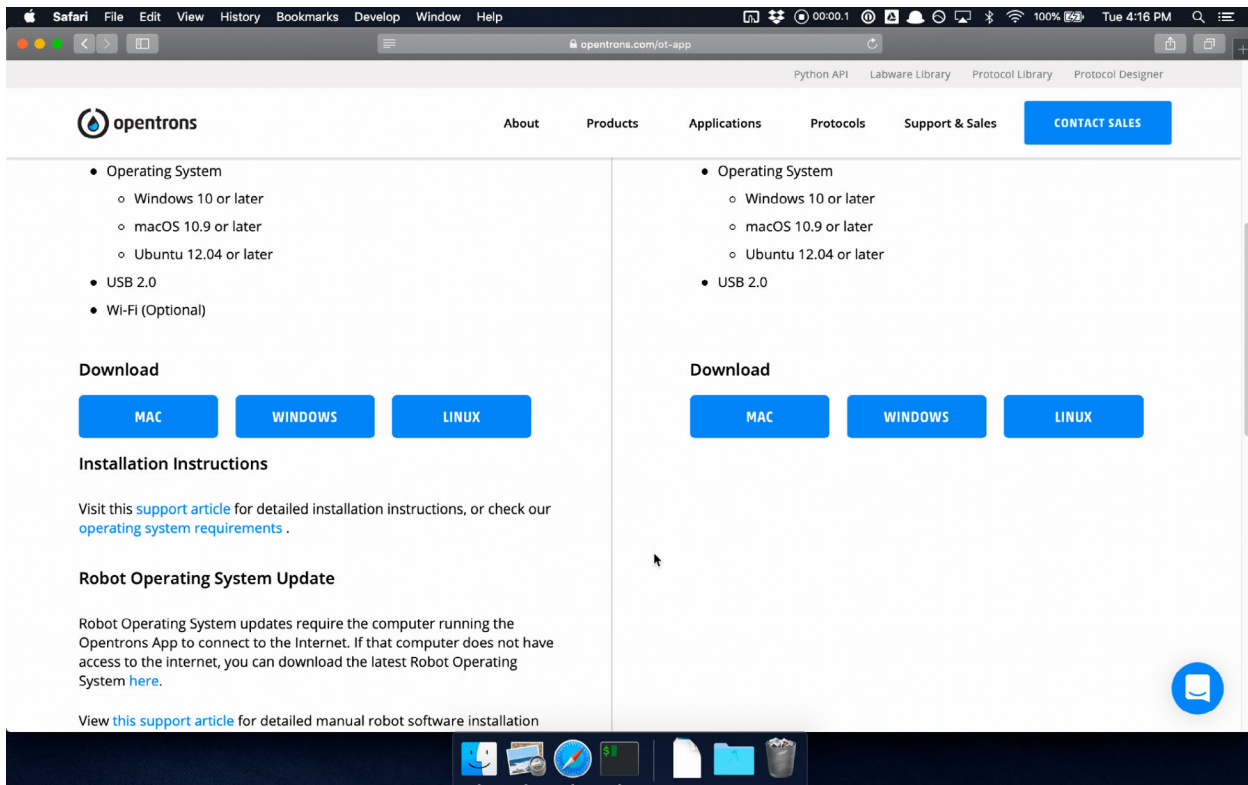
Visit the Opentrons website to download the Opentrons Client Application for macOS.

Windows 10

The Opentrons Client Application for Windows is distributed as an installer. To install the app:

1. Click the **Windows** download button on the download page.
2. Click **Run** in the browser, or click **Save** and then double-click the downloaded file named **Opentrons-...exe** in your Downloads folder.
3. Follow the prompts in the install wizard to install the app for a single user or for all users of the computer (administrator permission required).
4. The app will launch automatically once the install wizard is finished.
5. Click **Open** or **Allow** on any security or firewall prompts to ensure the Opentrons App is able to run and communicate with your OT-2.
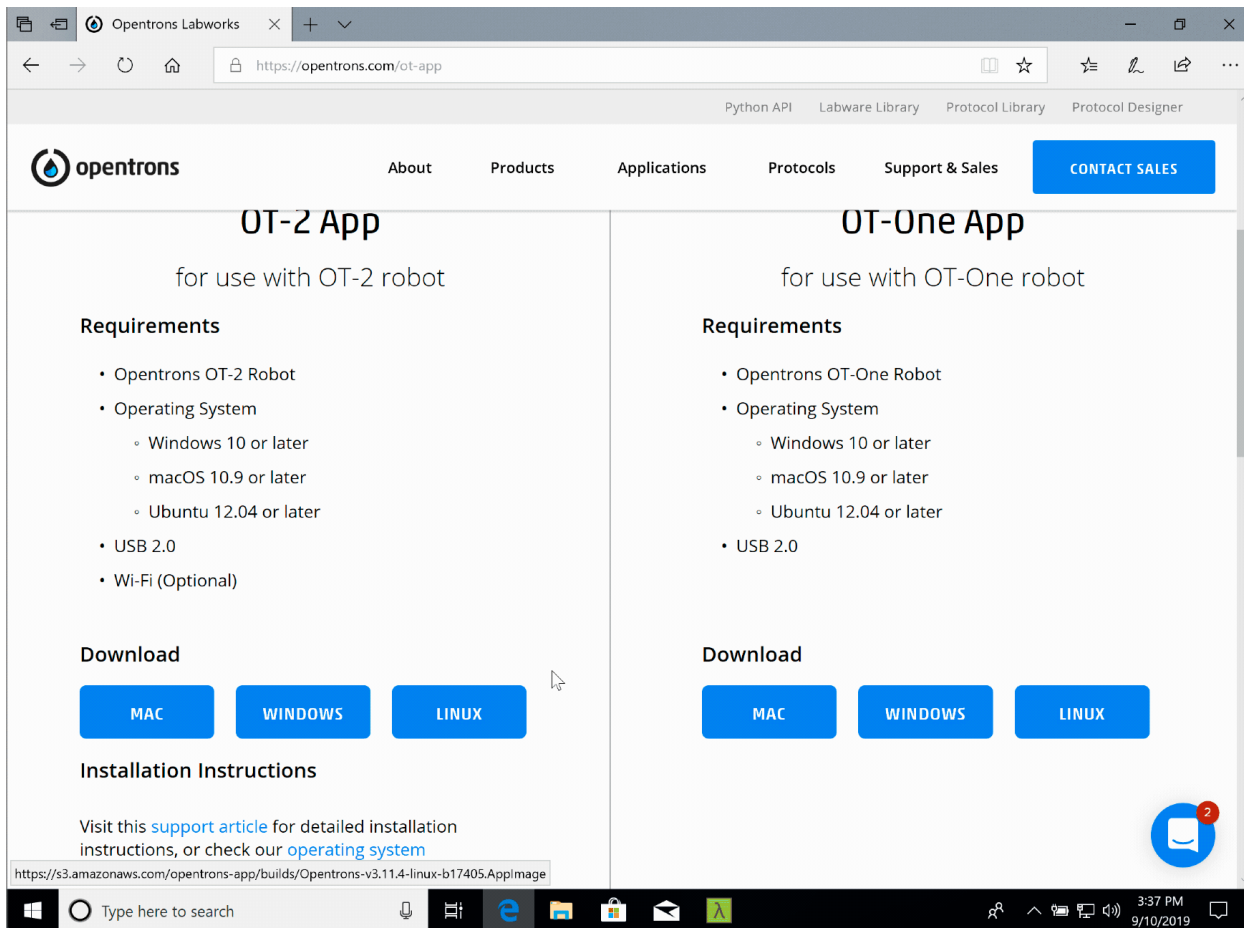
Linux

The Opentrons Client Application for Linux is distributed as an [AppImage](#) package. Users can download the Opentrons Client Application and run it without `apt-get` or other package managers.

The following instructions are specifically for Ubuntu 18.04 but should be broadly similar for all Linux distributions.

1. Click the **Linux** button on the download page and then click **Save.**
2. Move the downloaded AppImage file from the Downloads folder to a place easily accessible on the desktop or in the Applications folder.
3. Right click the downloaded AppImage file and select **Properties.** A new window will open.
4. Under the **Permissions** tab, check **Allow executing file as program**.
5. Double-click the AppImage file to run the app.
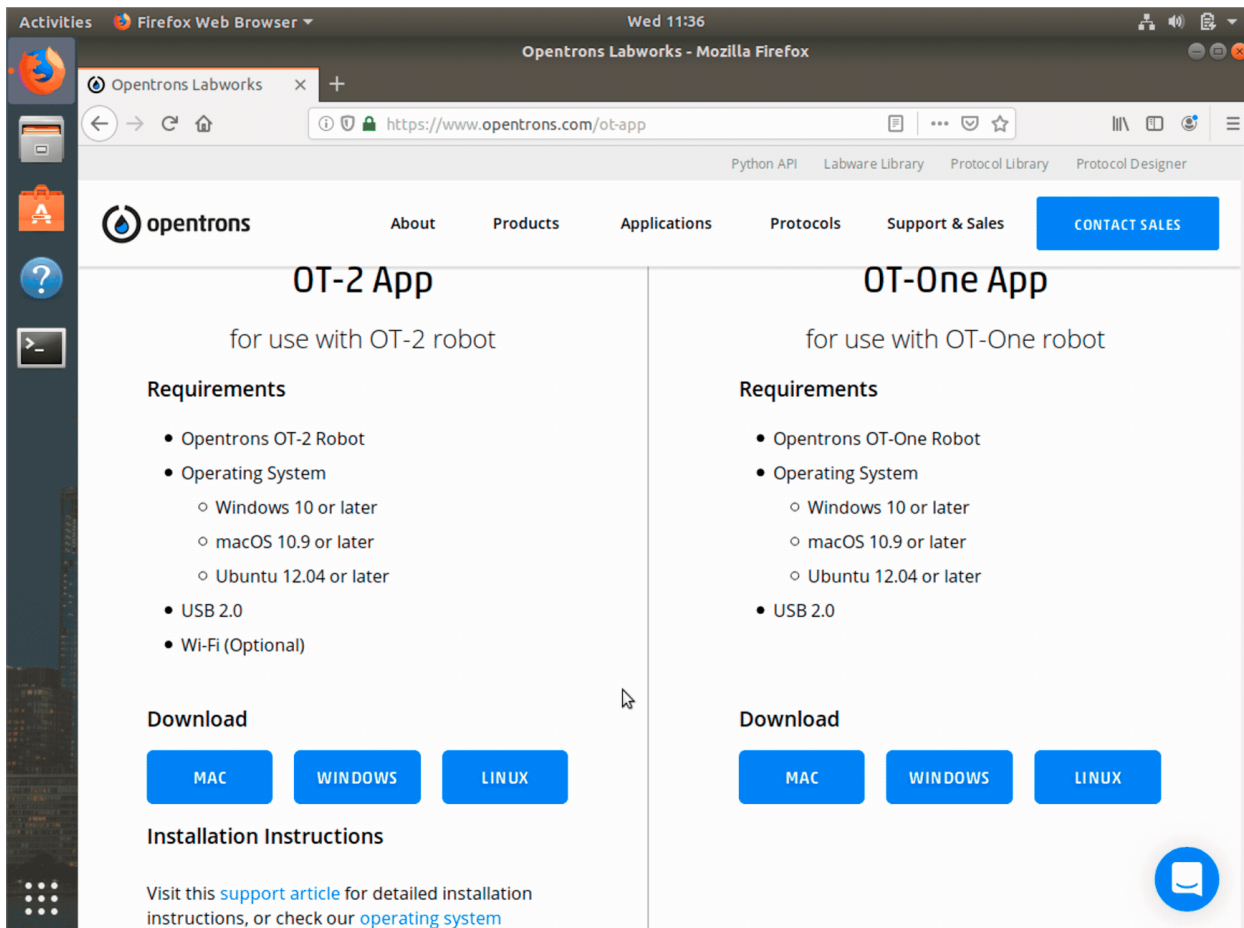
Visit the Opentrons website to download the Opentrons Client Application for Linux.

Third-party launchers are not supported on the OT-2

Third-party AppImage launchers, such as AppImageLauncher, are known to interfere with the Opentrons App's self-update functionality and thus are not supported nor do we recommend the use of third-party launchers to operate the OT-2.

Software Versions

Patch notes for previous software versions can be found in the CHANGELOG.md file in the Opentrons/opentrons GitHub repository. All notable changes to this project will be documented in this file. See *Conventional Commits* for commit guidelines.

3.4 Update the Opentrons App and the OT-2

Ensure that the OT-2 is running the latest software update to take full advantage of the latest features (and bug fixes) that have been released since the initial manufacturing of the OT-2.

Understanding the update process

When the computer is properly connected to the Opentrons Liquid Handler and the Opentrons Client Application is opened, the Opentrons Liquid Handler will have access to the internet using the computer's internet connection. An update message will appear when the robot is first connected.

It's important to understand there are *two different phases* involved in updating the robot's software:

1. Updating the **Opentrons App** on the client's computer; and
2. Updating the **OT-2** robot.

Caution: Updating the app on a user's computer does not automatically update the software on the OT-2.

Set up the Opentrons App and OT-2 without internet access

The Opentrons Client Application usually accesses software releases from the internet, but it can be updated without an internet connection. To set up the OT-2 without connecting the Opentrons Liquid Handler to a Wi-Fi network, download the version of the Opentrons App appropriate for the computer from *this page*.

📄 **Note:** If not connecting the OT-2 to the internet, remember to periodically check the Opentrons website for updates because the Opentrons Client Application cannot automatically detect newly available software versions without an internet connection.

Update your OT-2

When the OT-2 is first connected to the Opentrons App, if a new version of the robot software is available, the following message will appear:

⚠ Robot Update Available

Your robot software version and app version are out of sync.
For an optimal experience, we recommend you upgrade your robot software to 6.0.1 match your app.

1. Your current app version: 6.0.1
2. Your current robot server version: 6.0.0
3. Latest available version: 6.0.1

| NOT NOW | VIEW ROBOT UPDATE |

Notification that appears in the Opentrons App when a robot software update is available.

The version numbers displayed will reflect the currently installed and latest available versions of the Opentrons App.

To update the robot, click **View Robot Update** and then select **Update Robot** after reviewing the changes for the new software version on the next screen.

There are three phases to updating the robot:

1. Uploading to the OT-2 robot: The Opentrons App first sends the file to the robot. This usually takes less than a minute.
2. Validating the update: The robot then validates the file to make sure no errors occurred in transit.
3. Applying the update: The robot will apply the update and this step can take 3 minutes. During this time, the light on the front will turn off and then begin to flash blue. The pipette motors will begin to move slightly, and an audible noise will be heard.

Update complete message: When this message displays in the Opentrons App, the update has been completed successfully and the user can move on to the next step in setting up the Opentrons robot.

Future updates

Opentrons software is updated frequently to bring new features to users. When updating, make sure to update both the Opentrons App and the OT-2 for the best experience.

If you dismiss an update notification or want to check for updates manually, go to **App Settings** by clicking the gear icon at the bottom left of the window. App Software Version shows the version of the app you are running currently. To the right, you will either see the text "Up to date" or a button to **View software update**.

If an update is available for the OT-2 robot, a banner will appear on its device card and device detail page, saying "A robot software update is required to run protocols with this version of the Opentrons App." Click **View update** in the banner to apply the update to the robot.

Disable Opentrons App Update Notifications

The user can turn off Opentrons App update notification pop-ups. The controls to do so are in the modal itself as well as a toggle in **App Settings > General > Software Update Alerts**. This setting only disables the pop-up. It does not stop the app from continuing to (gently) notify the user that an app update is available.
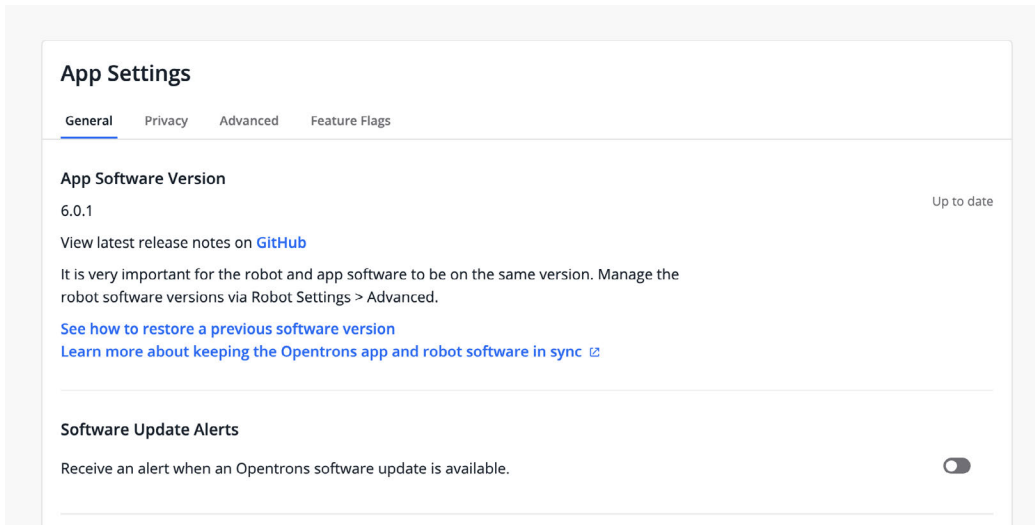
Troubleshooting an OT-2 Software Upgrade

If the message "successfully updated" does not display after waiting for 10 minutes, check to see if the blue light on the front of the robot is on and solid (not flashing). If the light is solid, this means the update was applied successfully and the user's computer missed the "Done" signal from the robot.

Note: Larger software updates will require more time to complete installation.

3.5 Uninstall the Opentrons Client Application

If experiencing problems with the Opentrons Client Application or if it is necessary to remove it from a client's computer system, uninstall the Opentrons App according to the following procedure.

Windows 10
1. From the **Start** menu, open **Add or remove programs.**
2. In the list of programs, select **Opentrons.** There may be a version number in the name, for example the name will display as "*Opentrons v3.12.0*."
3. Click **Uninstall.**

macOS
1. Locate the Opentrons.app application file. It will be located in the Applications directory. If you cannot find it, use Spotlight to search for "Opentrons" and press **Command+Return** to open Finder where the file is located.
2. Move Opentrons.app to the trash.

Linux
1. Locate the Opentrons AppImage file. The filename will start with "Opentrons" and end with ".AppImage."
2. Delete the AppImage file.

Remove configuration Files

If experiencing problems with the Opentrons App, clearing out the configuration files before reinstalling can help resolve issues.

⚠️ **Warning:** if you clear out the configuration files and reinstall the app, all your previous preferences, including a user's support chat identity will be lost.

To remove configuration files, delete the following directory:

- Windows 10: `C:\Users\your_user_name\AppData\Roaming\Opentrons`
- Linux: `/home/your_user_name/.config/Opentrons`
- macOS: `/Users/your_user_name/Library/Application Support/Opentrons`

Remember to replace your_user_name with your actual username.

3.6 Attaching Pipettes

The OT-2 has two *pipette mounts.* Each pipette mount can hold one *Opentrons pipette*. The pipette screws into the mount, and the OT-2 controls the pipette electrically through a cable. Follow the steps laid out in this section to connect the Opentrons pipette to the OT-2.
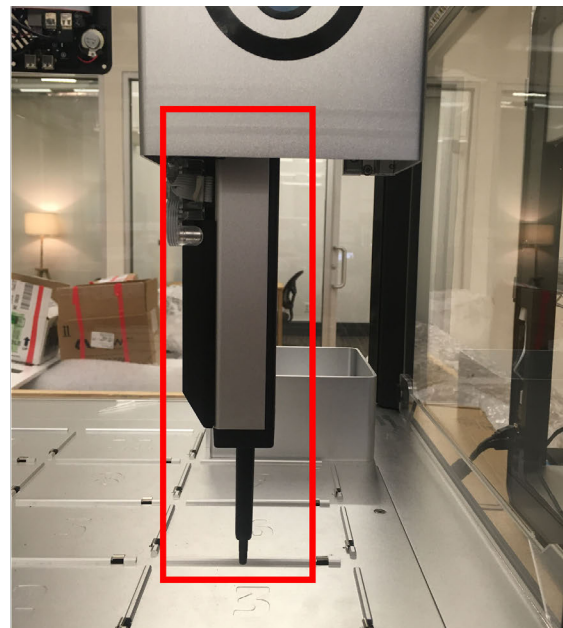
Left picture is an image of the left mount with no pipette attached. In the picture, the right mount is raised and hidden from view. Right picture is an image of the left mount with a P300 Single-Channel GEN2 attached. In the picture, the right mount is raised and hidden from view.

GEN1 or GEN2 pipette

Determine if you have a GEN1 or GEN2 pipette. Some of these steps change depending on whether the pipette is GEN1 or GEN2. If unsure, see *Determine the difference between a GEN1 and GEN 2 Pipette*

Attaching GEN2 pipette

Required materials for mounting a pipette
- One **pipette screw**, included in a small plastic bag with the OT-2.
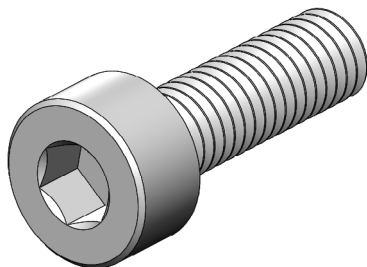
3-D rendering of Opentrons pipette screw.

- **2.5 mm hex screwdriver**
- **If attaching an 8-Channel GEN2 pipette**, the **Calibration Block will also be necessary for assembly**. The Calibration Block came in its own box, shipped along with your 8-Channel GEN2 pipette.

 Note: Individually purchased 8-channel pipettes don't come with a calibration block.

Step 1: Lower the pipette mount for access

In the Opentrons App, go to the **Pipettes & Modules** page for the OT-2. Select **Attach** for the mount that you want your pipette to go on. The gantry will move towards the front window, and the mount will lower.
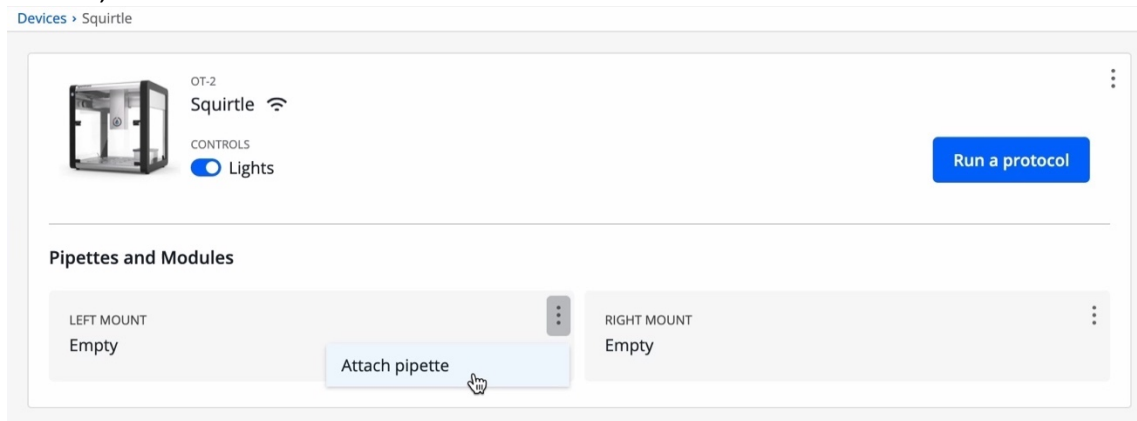
Selecting "Attach pipette" from the three-dot menu

Step 2: Select the pipette to attach

From the dropdown menu in the Opentrons App, select the model of pipette to attach.

Tip: If you have both a GEN1 pipette and a GEN2 pipette to attach, attach the GEN1 pipette first. Otherwise, space limitations might make it difficult to squeeze into place.

Step 3: Align the pipette to the mount

Pass the pipette into the OT-2 through the front window. Hold it vertically somewhere behind the lowered mount.

Slide the pipette towards you, keeping it vertical. The connector on the pipette needs to match up with the connector on the mount. The front edge of the pipette should be even with the front edge of the mount.

NOTE: If having trouble, approaching the OT-2 from a lower angle will allow for a clearer view of the pipette mount.
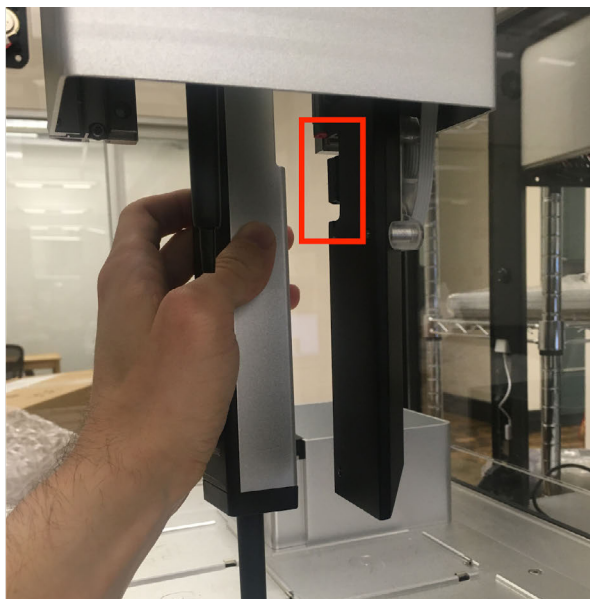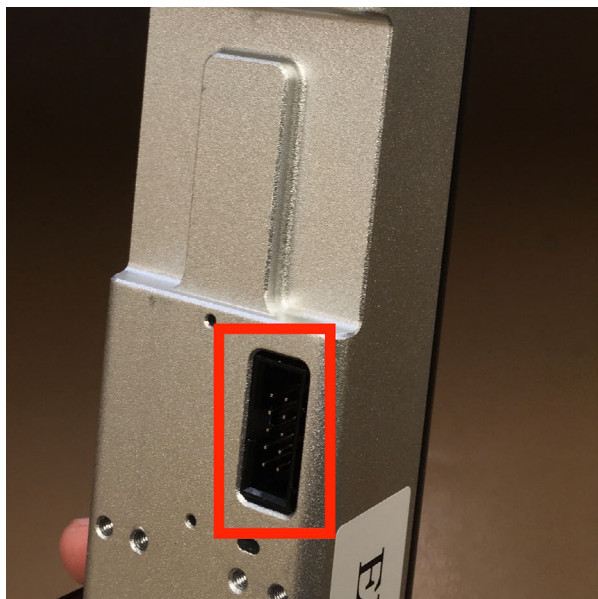
The picture on the left displays the connector port on the pipette. The picture on the right shows how the pipette should connect to the connector on the pipette mount.

Hold the pipette as indicated in the picture to align it to the mount.

Step 4: Secure pipette to mount
A GEN2 pipette when properly attached will feel like it's hanging off a hidden ledge. Hold the pipette in place.

Step 5: Screw the pipette into the mount

A total of three 2.5 mm hex screws will secure your pipette to its mount. There are:
- Two top screws, which are permanently embedded in the transparent plastic at the top of the mount.
- One bottom screw, which you need to insert yourself. Several bottom screws came with your pipette or OT-2, packed in a small bag.

Follow the Opentrons App's instructions to screw in all three screws. If attaching an 8-Channel pipette, leave them slightly loose; the screws will be tightened in a later step.

Note: If using an 8-Channel pipette on an *OT-2R*, ensure that you are using the correct bottom pipette attachment screw which is shorter than the one used for the OT-2. Otherwise, the *8-Channel pipette will not be able to eject tips*.
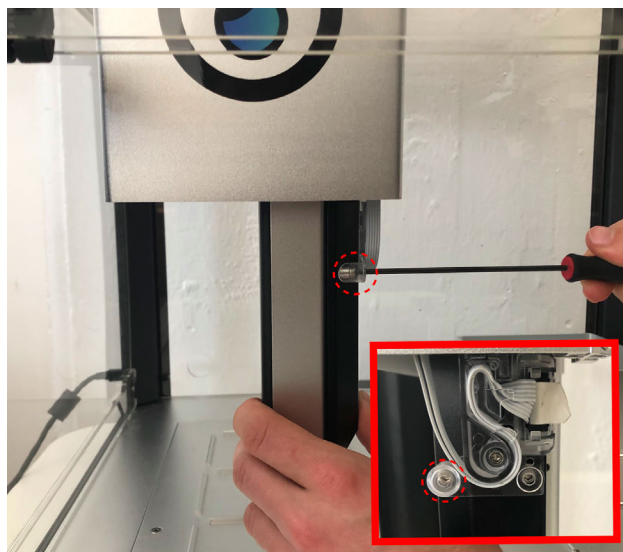
Two top screws
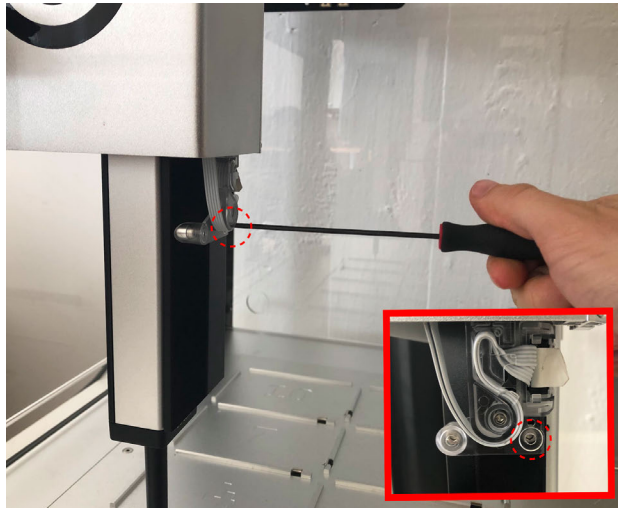
One bottom screw

Step 6: Connect the pipette cable

After screwing in the pipette, plug it into the pipette mount. The OT-2 will control the pipette through a white ribbon cable.

Find the white connector tab and press it into the pipette.
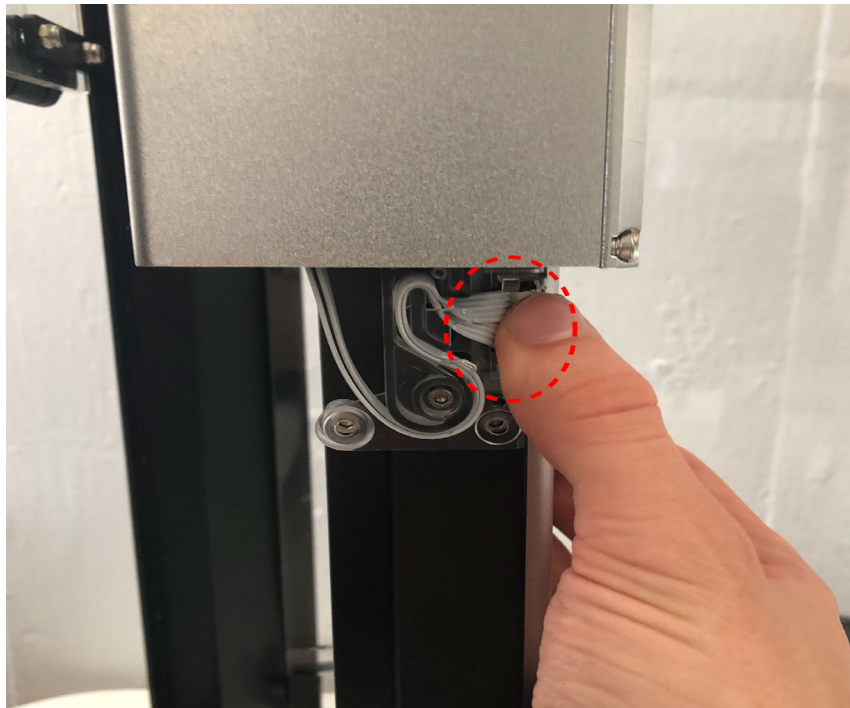
Press the white connector tab into the pipette as indicated above.

Step 7: Check the pipette connection

Next, you'll make sure that the OT-2 can communicate with the pipette over the cable that you just plugged in.

In the Opentrons App, select **Have robot check connection.** The OT-2 will test the connection automatically.

Attach P20 Single-Channel GEN2 Pipette

STEP ONE	STEP TWO
Attach pipette to mount, **starting with screw 1.**	Connect the pipette to robot by pushing in the white connector tab.
2.5mm screwdriver	
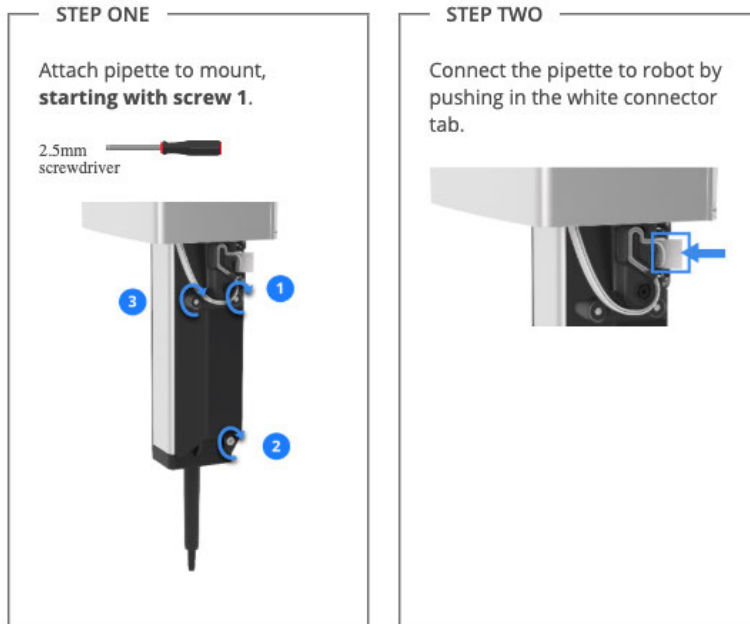
HAVE ROBOT CHECK CONNECTION

When the robot is checking the pipette connection the Opentrons App will walk through the above checks.

If all goes well, a green checkmark will appear in the Opentrons Client Application.

✅ P20 Single-Channel GEN2 successfully attached.

CONTINUE TO PIPETTE OFFSET CALIBRATION

EXIT WITHOUT CALIBRATING

When the pipette is connected, the Opentrons App will confirm the connection with a green checkmark and prompt the user to proceed to pipette offset calibration

📄 **Note:** If a green checkmark doesn't appear, an error message of "Unable to detect pipette" will display. If this message appears, see _Troubleshooting pipette connection problems_.

Step 8: Attach and level an 8-Channel pipette

8-Channel pipettes are sensitive to tilt. If attaching one, take the steps needed to make sure it's attached perfectly straight. To learn more, see *Leveling 8-Channel pipettes*.
If attaching a Single-Channel pipette, skip ahead to the next step.

Leveling 8-Channel pipettes

If an 8-Channel pipette is attached slightly askew, it can affect:
- Reliability of tip pickup and ejection
- Evenness of aspiration and dispensation
- Accuracy of calibration

To avoid this, ensure that 8-Channel pipettes are perfectly level. 8-Channel pipettes are leveled during the *pipette attachment process*.

Attaching an 8-Channel GEN1 pipette

1. By hand *gently and slowly* pull the pipette all the way down to the OT-2's deck.
2. Wiggle the pipette so that all 8 of its nozzles touch the deck at the same time.
3. While holding the pipette down, tighten the 3 screws.

Attaching an 8-Channel GEN2 pipette

Use the Calibration Block shipped with your pipette.

Calibration Block

Follow the Opentrons App's video instructions to use the Calibration Block to level your pipette. Remember to tighten the 3 screws when you're done.

Setting up OT-2 for the first time

If setting up the OT-2 for the first time, select **Exit without calibrating** to return to the device detail page. A banner will appear in the Pipettes and Modules section with the message, "Pipette Offset calibration needed. Calibrate now." Ignore the warning for now.

When the pipette is attached successfully the above message will display.
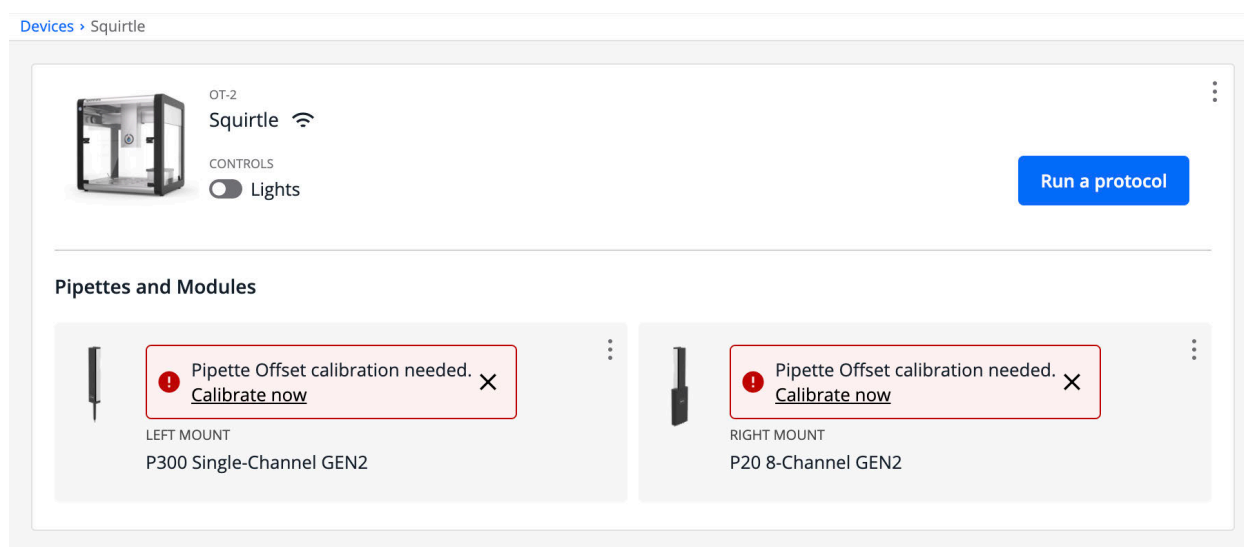
When Pipette Offset calibration is needed, the Opentrons App will display a banner that includes a link to start the calibration.

Attaching a new pipette to an OT-2

If attaching a new pipette to an OT-2 and the OT-2 already has a good *deck calibration*, select **Continue to pipette offset calibration**.

3.7 Calibration

The OT-2 moves pipettes around in 3D space. To do that accurately and precisely, you need to calibrate it. Positional calibration occurs in four parts: Deck Calibration, Tip Length Calibration, Pipette Offset Calibration, and Labware Position Check.

Deck Calibration

Deck Calibration makes sure your OT-2 moves its pipettes in the correct directions, and by the correct distances. It gives your OT-2 its sense of direction and its sense of scale.

Deck calibration compensates for a few different kinds of slight physical variation:

- Variation in how the deck is attached. The OT-2's deck is removable, and it has some wiggle room when it's attached to the OT-2's frame. It might be rotated a little bit.
- Variation in the rails that the pipettes ride around on. Those rails might not be perfectly perpendicular to each other, or to the rest of the OT-2.
- Variation in the motors and pulleys. A system of motors and pulleys moves the pipettes forward, back, left, and right. The amount that those motors need to turn to move the pipettes a given distance might not perfectly match the theoretical amount.

Those variations can be caused by:

- Normal variation from how the factory assembles the OT-2
- Jostling from shipping or moving
- Normal variation from when you unbox and assemble the OT-2
- Mechanical shifts from using the OT-2 a lot, or over a long time

An inaccurate Deck Calibration can cause problems like the pipette being off-position by a direction or amount that changes depending on where it is on the deck.

Deck Calibration doesn't compensate for the deck being translated (shifted straight forward, back, left, right, up, or down). So, on its own, it isn't enough to make sure your pipettes can accurately move to targeted locations. That gap is filled by Pipette Offset Calibration, Tip Length Calibration, and Labware Position Check.

The OT-2 also can't currently account for the deck being tilted or bent. It assumes the whole deck sits perfectly level.

How Deck Calibration works

Deck Calibration works by measuring the relative alignment between:

- the OT-2's deck, the surface that holds all your labware, and
- the OT-2's gantry, the structure and machinery that carries the pipettes around.

When you calibrate the deck, the Opentrons App will indicate reference points that are precision-engraved into the deck surface. You'll manually jog the pipette to each of those points. This shows your OT-2 how it needs to move its motors in order to travel between those points. By extension, it gives your OT-2 its sense of direction and scale, relative to the deck. It will scale, rotate, and shear (but not translate) its internal coordinate system to match.

When to perform Deck Calibration

You need to calibrate the deck when first setting up the OT-2. The Opentrons App will guide you through it.

You can recalibrate the deck whenever you want, but you only need to do it once in a while. You should only need to do it:

- After moving the OT-2 more than a short distance.
- Once or twice per year, for maintenance.
- When troubleshooting certain movement, positioning, and [calibration problems](#).

The OT-2 will use the new Deck Calibration until you do Deck Calibration again.

Tip Length Calibration

When a pipette picks up a tip, the distance that the tip presses onto the pipette nozzle will vary from pipette to pipette—even between pipettes of the same exact model. (This is because of normal manufacturing variation in the pipette nozzles.) Tip Length Calibration accounts for this by informing your OT-2 how that model of tip fits onto the nozzle of that specific, physical pipette. Then the OT-2 knows how far up or down to move that pipette when it's running your protocol so that the tip is positioned at the correct height.

To run a protocol, you need a Tip Length Calibration for every combination of physical pipette and model of tip that's used.

Tip Length Calibration doesn't account for differences between individual tips. The OT-2 assumes that each tip is consistent with other tips of the same model. For example, if your protocol has a P300 Single-Channel pipette using two racks of 96 Opentrons 300 μL tips, all of those 192 tips will share the same Tip Length Calibration. The OT-2 assumes that each of those 192 tips presses onto that pipette in the same way, forming the same total length.

Tip Length Calibration can also compensate for small differences between a tip's expected length (according to its [labware definition](#)) and its true, physical length. But it won't let you substitute entirely different tips. For example, if you want to use third-party Fisherbrand 300 μL tips, don't try to trick the OT-2 by having your protocol load Opentrons 300 μL tips, physically providing the Fisherbrand 300 μL tips instead, and expecting Tip Length Calibration to cover up the difference. Either physically provide the Opentrons 300 μL tips that the protocol expects, or rewrite your protocol to load Fisherbrand 300 μL tips with a [custom labware definition](#). Otherwise, you might cause movement, positioning, and calibration problems.

How Tip Length Calibration works

Tip Length Calibration measures how far the bottom of the tip extends below the bottom of the pipette nozzle.

During Tip Length Calibration:
1. You'll manually jog the bare pipette—without any tip attached—so that its nozzle just barely touches a flat surface. The OT-2 will note the pipette's height.

2. The pipette will pick up a tip.
3. You'll manually jog the pipette—with the tip attached, this time—so that it just barely touches that same flat surface. The OT-2 will note the pipette's height again.
4. The OT-2 will take the difference between the two heights as the tip length and store it for any time in the future when it uses that specific pipette with that model of tip.

The OT-2 gives each physical pipette its own set of Tip Length Calibrations. For example, if your OT-2 has two P300 Single-Channel GEN2 pipettes, each one could have one Tip Length Calibration for Opentrons 300 µL tips and one Tip Length Calibration for Opentrons 200 µL filter tips. And all four of those Tip Length Calibrations would be independent from each other.

When to perform Tip Length Calibration
The Opentrons App will automatically prompt you to calibrate tip length when you need to. Specifically, it can prompt you:
- Before running a protocol, if it uses a pipette and tip pairing for which you haven't yet done a Tip Length Calibration.
- Before Pipette Offset Calibration. (See the "Pipette Offset Calibration" section, below.) Tip Length Calibration is a prerequisite for Pipette Offset Calibration.

You can also manually choose to recalibrate, if you suspect something is wrong with your current Tip Length Calibration. To do so, go to the Calibration section of the Robot Settings page for your robot in the Opentrons App.

Pipette offset calibration
When the OT-2 moves a pipette to different points, there might be a systematic error. For example, your particular P300 Single-Channel might always go 1 mm too far to the right when it's attached to your OT-2's left mount.
Pipette Offset Calibration compensates for this. It nudges the pipette—wherever it goes—by some constant amount to make the true tip position match what the protocol targeted. The offset can be forward, back, left, right, up, or down.
This accounts for:
- Manufacturing variation in the pipette
- Manufacturing variation in the mount
- Variation in how that pipette attaches to that mount
- How the whole deck is shifted side-to-side. (On the other hand, how the deck is rotated is one of the things that Deck Calibration accounts for.)

How Pipette Offset Calibration works

When you calibrate a pipette offset:
1. The OT-2 will attempt to move the pipette to a predefined reference point.
2. You'll manually adjust the pipette so that it's exactly centered over it.
3. The OT-2 will save that adjustment and use it until you overwrite it.

Your OT-2 has one Pipette Offset Calibration per physical pipette, per mount. For example, if you have a P300 Single-Channel GEN2 pipette, it will have separate Pipette Offset

Calibrations for when you attach it to the left mount and when you attach it to the right mount. And if you get a second P300 Single-Channel GEN2, it will have its own set of Pipette Offset Calibrations.

When to perform Pipette Offset Calibration

The Opentrons App will automatically prompt you to calibrate pipette offsets when you need to.

Specifically, it can prompt you:
- After attaching a pipette, if you've never calibrated an offset for that pipette on that mount before.
- Before running a protocol, if one of the pipettes that it uses doesn't already have a Pipette Offset Calibration for that mount.
- After recalibrating tip length, if that model of tip was the one used when you last did Pipette Offset Calibration for that pipette on that mount. See the "Relationships between calibrations" section, below.

If you suspect there's something wrong with your existing Pipette Offset Calibration, you can also start a Pipette Offset Calibration yourself. To do so, go to the Calibration section of the Robot Settings page for your robot in the Opentrons App.

Labware Position Check and Labware Offset Data

Labware Position Check fine-tunes the expected position of a particular combination of labware definition, deck slot, and OT-2.

Theoretically, since the OT-2 knows [the labware's geometry](#) and where each slot on the deck is, it should be able to perfectly predict the location of each well. However, in reality, there can be slight variations in physical labware and how those labware fit into slots. Those slight variations do add up, and can cause the pipette tip to miss wells or crash. Labware Position Check compensates for this.

How Labware Position Check works

Once the previous calibration steps have been completed, you can perform Labware Position Check after you upload a protocol and before you run it.

For each labware in your protocol:
1. The OT-2 will move the pipette tip to where it thinks the top of well A1 is.
2. You'll manually jog the pipette to center it perfectly at the top of well A1, as needed.
3. The OT-2 will save that adjustment for when it runs your protocol.

Then, during your run, any time the pipette moves to any well in that labware, the OT-2 will apply the Labware Offset data that you saved.

When to perform Labware Position Check

Opentrons recommends using Labware Position Check before every protocol run, but it is not mandatory. If you have saved Labware Offset data and apply it before running Labware Position Check, the pipette will start at those saved positions — you only have to double-check their accuracy.

For details on running Labware Position Check and using Labware Offset data, see [How Labware Offsets work on the OT-2](#).

There is no need to — and in fact, no way to — check the position of any well other than A1. The OT-2 already accounts for labware with non-identical wells by making reference to the labware definition. For example, if you are having issues when using differently sized tubes in the same tube rack, check that your labware definitions match the physical labware you are using in your protocol.

Relationships between calibrations

The different parts of your OT-2's positional calibration—Deck Calibration, Tip Length Calibration, Pipette Offset Calibration, and Labware Position Check—can all be calibrated, reset, and recalibrated individually.

However, some parts depend on others. There's an order of operations to them: you might need to do one part before you can do another. And recalibrating one part can sometimes mean you also have to recalibrate other parts.

Normally, you don't need to worry about the details. The Opentrons App will automatically ensure that you calibrate things in the right order.

General requirements

Each part of positional calibration on the OT-2 is mandatory except for Labware Position Check, which is highly recommended.

Before you can run a protocol, you need:
- Deck Calibration, for the whole OT-2.
- A Pipette Offset Calibration for each pipette that your protocol uses, on its respective mount.
- A Tip Length Calibration for every combination of tip and pipette that your protocol uses.

After you upload a protocol to the OT-2, the Opentrons App will prompt you to complete any missing calibration steps before letting you run the protocol. It will also prompt you to optionally run Labware Position Check.

Pipette Offset Calibration requires the OT-2 to have a Deck Calibration

When you calibrate a pipette offset, you center the pipette over one of the deck's engraved reference points. Therefore, Pipette Offset Calibration depends on the OT-2 already having a good Deck Calibration.

The Opentrons App won't let you start a Pipette Offset Calibration unless the OT-2 already has a valid Deck Calibration. If you recalibrate the deck later on, that will clear all your Pipette Offset Calibrations, since they wouldn't make sense with the new Deck Calibration, anyway.

Pipette Offset Calibration requires that pipette to have one Tip Length Calibration

The process of Pipette Offset Calibration involves picking up an Opentrons tip and using that as a pointer when centering the pipette over one of the deck's engraved reference points, instead of using the bare pipette nozzle itself. This is necessary because the pipette can't always lower far enough for its nozzle to reach the deck's surface, and, even if it could, it would be hard to tell when it's centered.

Therefore, Pipette Offset Calibration depends on having a good Tip Length Calibration for that model of Opentrons tip. The OT-2 needs to know how much length that tip adds to that pipette in order to set that pipette's z-offset (height adjustment). For other kinds of tips that your protocol uses—whether they're different models of Opentrons tips, or third-party tips that you're using with a custom labware definition—the Tip Length Calibrations are totally independent from Pipette Offset Calibration.

If you try to calibrate a pipette's offset without it already having a Tip Length Calibration for the specified model of tip, the Opentrons App will automatically guide you through completing that Tip Length Calibration first. And if you redo that Tip Length Calibration, your Pipette Offset Calibration will be cleared, and you'll immediately be prompted to redo it. After all, the accuracy of the original pipette offset relied on the accuracy of the original tip length.

Perform Calibration Procedure

Part 1: Initiate robot calibration
Connect to your OT-2 and find it in the **Devices** list. Click on the three-dot menu on its device card and select **Robot Settings** to go to the Calibration tab of the Robot Settings screen.

You might see warnings on this screen indicating that some calibrations are missing. You can ignore these warnings for now, since they will disappear when calibration is complete.

Robot Settings

Calibration   Networking   Advanced

About Calibration

For the robot to move accurately and precisely, you need to calibrate it. Positional calibration happens in three parts: deck calibration, pipette offset calibration and tip length calibration.

[Download calibration data]

See how robot calibration works

⚠ Deck calibration missing   Calibrate now

Deck Calibration

Deck calibration measures the deck position relative to the gantry. This calibration is the foundation for tip length and pipette offset calibrations. Calibrate your deck during new robot setup. Redo deck calibration if you relocate your robot.

[Calibrate deck]

Not calibrated yet

⚠ Pipette Offset calibration missing

Pipette Offset Calibrations

Pipette offset calibration measures a pipette's position relative to the pipette mount and the deck. You can recalibrate a pipette's offset if its currently attached to this robot.

Pipette Model and Serial	Mount	Tip Rack	Last Calibrated	
P300 Single-Channel GEN2 P3HSV212021070904	Left		⚠ Missing calibration	⋮
P20 8-Channel GEN2 P20MV212021081906	Right		⚠ Missing calibration	⋮

Calibration warning messages will appear before initial calibration. After completing calibration, the messages will no longer appear.

Part 2: Perform the deck calibration
Closely follow the Opentrons App's instructions to perform the deck calibration.

Part 3: Finish up
When calibration is complete, the OT-2 will return the tip to its rack and save the deck calibration.

A checkmark will then appear along with the message "Deck Calibration complete".

Deck Calibration complete!

Exit
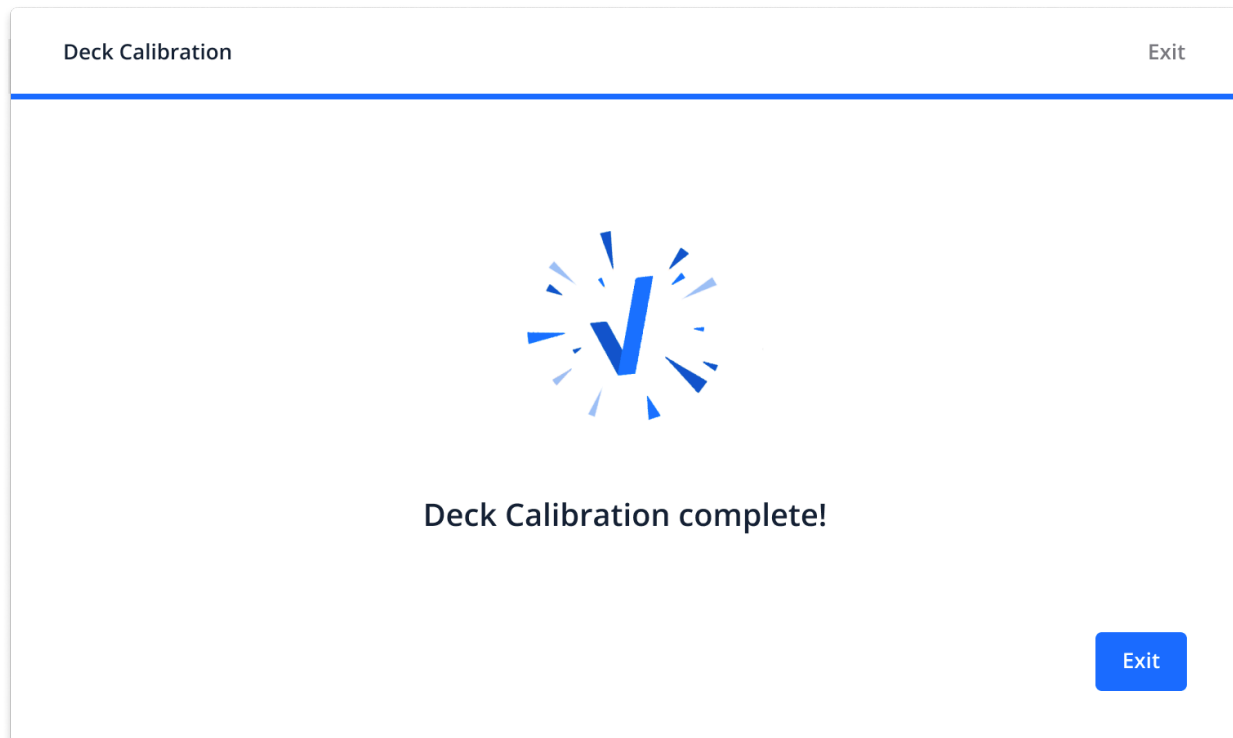
A checkmark will appear when Deck Calibration is complete.

Perform Tip Length and Pipette Offset Calibration
After Deck Calibration, it is necessary to conduct the rest of positional calibration.

Tools to perform Tip Length Calibration:
- Opentrons pipettes. The pipettes should be attached to the OT-2.
- *Opentrons* tips for those pipettes, in *Opentrons* tip racks. When running your protocols you can use other tips, but if calibration occurs with other tips, the calibration data will not be accurate.
- Calibration Block

Step 1: Begin
On the Calibration tab of the Robot Settings screen, scroll down to Pipette Offset Calibrations. Click on the three-dot menu next to a pipette and select **Calibrate Pipette Offset**. If using two pipettes, calibrate one pipette at a time.

> ⛔ Pipette Offset calibration missing

Pipette Offset Calibrations

Pipette offset calibration measures a pipette's position relative to the pipette mount and the deck. You can recalibrate a pipette's offset if its currently attached to this robot.

Pipette Model and Serial	Mount	Tip Rack	Last Calibrated	
P10 Single-Channel GEN1 123	Left		⛔ Missing calibration	⋮
P300 Single-Channel GEN1 321	Right	Opentrons 96 Tip Rack 300 µL	9/26/202...	

Calibrate Pipette Offset 🖱️
Download calibration data

Pipette Offset Calibration can be started from the Robot Settings screen (above) or the pipette card on the robot details screen.

The Opentrons App will ask for a Calibration Block. If you have one, select **Use calibration block**. If you don't have one, calibration can be performed by using the trash bin.

Tip: If "Remember my selection for next time and don't ask again" is selected, it is possible to change this preference later under **App Settings > Advanced > Tip Length Calibration Method**.

Tip Length Calibration   Exit

Do you have a calibration block?

This block is a specially made tool that fits perfectly on your deck and helps with calibration. If you do not have a Calibration Block, please contact us so we can send you one.

While you wait for the block to arrive, you can use the flat surface on the trash bin of your robot instead.

☑ Remember my selection and don't ask again   Use trash bin   Use calibration block

Calibration step asking for the Calibration Block.

Step 2: Calibrate tip length

During initial Pipette Offset Calibration, the Opentrons App will prompt the user to calibrate a *tip length* first. This is because the process of Pipette Offset Calibration involves attaching a tip, and the OT-2 will need to know the length of that tip beforehand.

Preparation for Tip Length Calibration, including required labware.

A tip rack is necessary for this step of calibration and the Opentrons App will show what kind of tip rack needs to be used. This Tip Length Calibration will inform the OT-2 how that model of tip fits onto the nozzle of that unique pipette.

If you don't have the kind of tip rack that the Opentrons App asks for, contact Opentrons Support.

Carefully follow the Opentrons App's instructions to calibrate tip length. It will guide the user through:
1. Lowering the pipette nozzle, without a tip attached, to a flat surface (either your Calibration Block or the trash bin).
2. Picking up a tip.
3. Lowering the pipette to the same flat surface, now with a tip attached, to measure the tip length.

When Tip Length Calibration is complete, a green checkmark and a "Tip Length Calibration complete" message will appear.
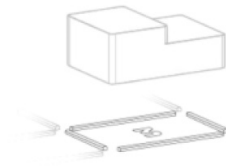
Tip Length Calibration complete!

You can remove the Calibration Block from the deck now.

Exit

When Tip Length Calibration is complete, a success message will appear and remind you to remove the Calibration Block, if you used it.

Step 3: Calibrate pipette offset
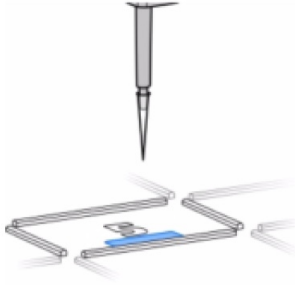
Calibrate z-axis in slot 5

Jog the pipette until the tip is **barely touching (less than 0.1 mm)** the deck in **slot 5.**

If the pipette is over the embossed 5, on the ridge of the slot, or hard to see, switch to the x- and y-axis controls to move the pipette across the deck.

JUMP SIZE

∩↓ **Jump Size**
– / +

Tiny	Small
0.1 mm	1 mm

DIRECTION CONTROLS

X- and Y-axis
Arrow Keys

Z-Axis
Shift + Arrow Keys

Jog too far or bend a tip? **Start over**

? Need help?

Confirm placement

Jog controls for Pipette Offset Calibration.

Carefully follow the Opentrons App's instructions to calibrate pipette offset. It will guide you through:

1. Lowering the pipette tip to the deck, to calibrate the pipette's *z-offset* (height adjustment).
2. Centering the pipette tip over an engraved reference point, to calibrate the pipette's *x-offset* (left/right adjustment) and *y-offset* (forward/back adjustment).

A checkmark and a "Pipette Offset Calibration complete" message will appear when pipette calibration is complete. Select **Exit** to return the tip to the rack and finish up.

Pipette Offset Calibration complete!

Exit
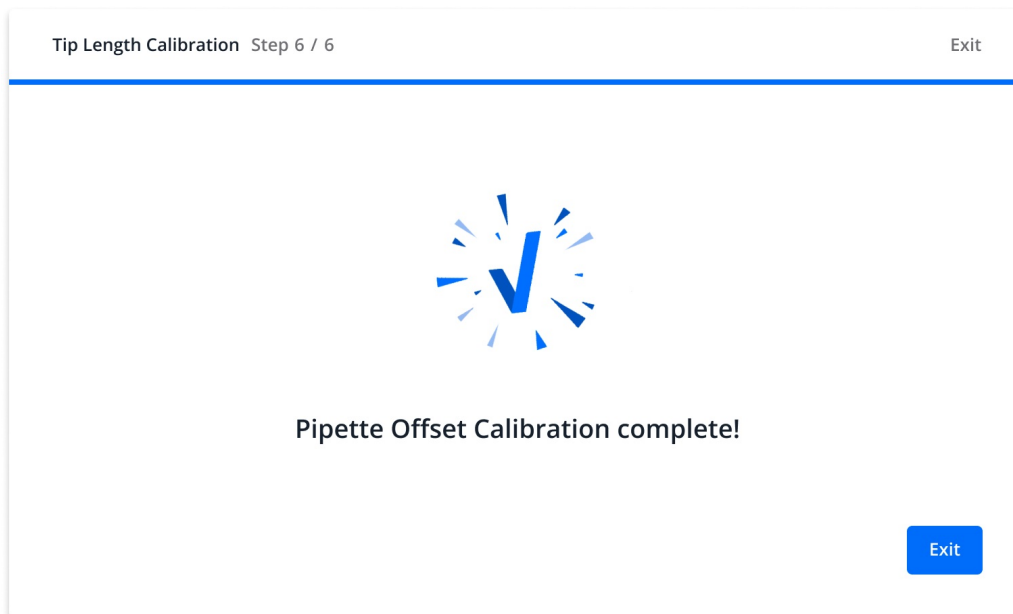
When Pipette Offset Calibration is complete, a checkmark will appear.

Next steps
If you've attached a second pipette, you'll need to do Tip Length Calibration and Pipette Offset Calibration for that one, too.

After completing a Pipette Offset or Tip Length Calibration, you shouldn't need to redo it unless you suspect something is wrong with it.

If you ever want to manually redo Pipette Offset Calibration or Tip Length Calibration, you can come back to the Calibration tab of the Robot Settings screen at any time.

3.8 Instrument Relocation and Transport Procedures

To prevent damage during shipping and transportation, the user should secure the axes of the OT-2 with 3 locking brackets. These are the same brackets and screws removed when the robot was initially unboxed. These screws and brackets were flagged with blue tape.

Relocation Procedure

Stage 1
1. Manually move the gantry up to the front of the robot, as indicated in the figure below.

Move the gantry up to the front of the robot.

2. Take the long screw for stage 1.

OT-2 bracket.

Move the bracket behind the right side of the gantry.

OT-2 bracket.

Make sure the hole in the bracket is aligned with the hole on the robot (pictured below).
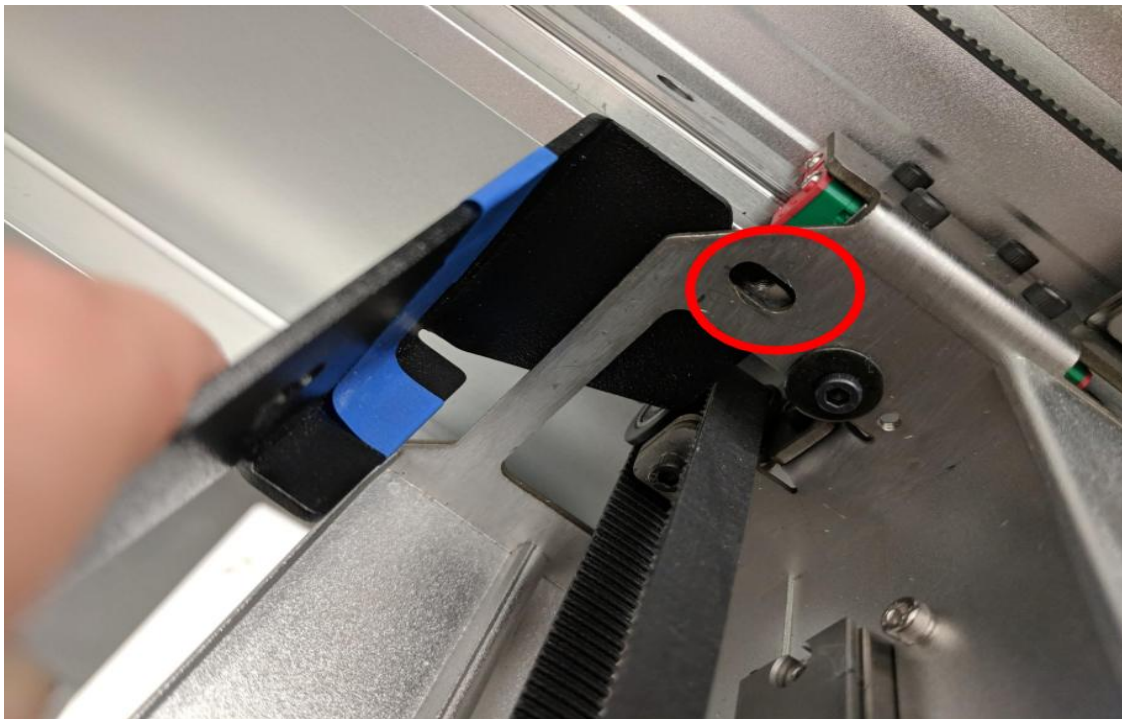
Hole in OT-2 bracket, highlighted by the red circle.

3. Loosely secure the long screw from stage 1 through the bracket and into the threaded hole on the robot.
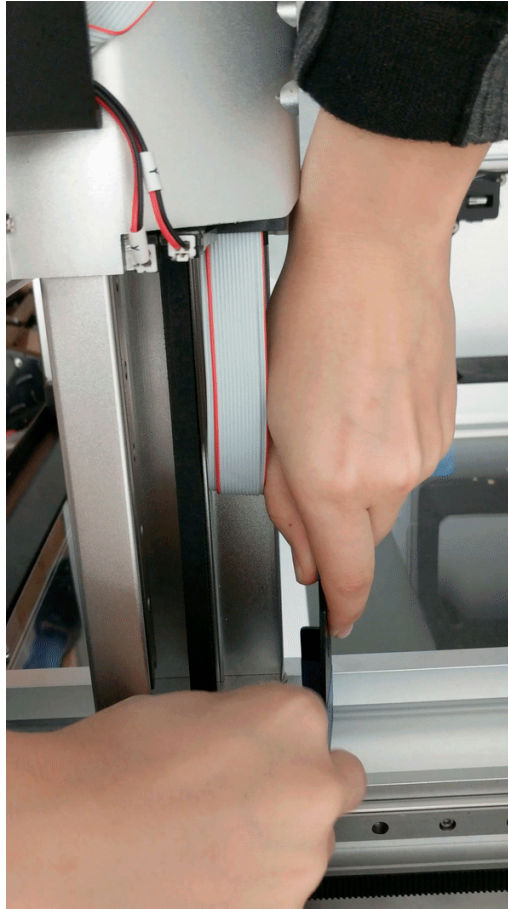
Secure the bracket.

4. Slide the gantry to the right side of the robot while holding the bracket parallel to the table.
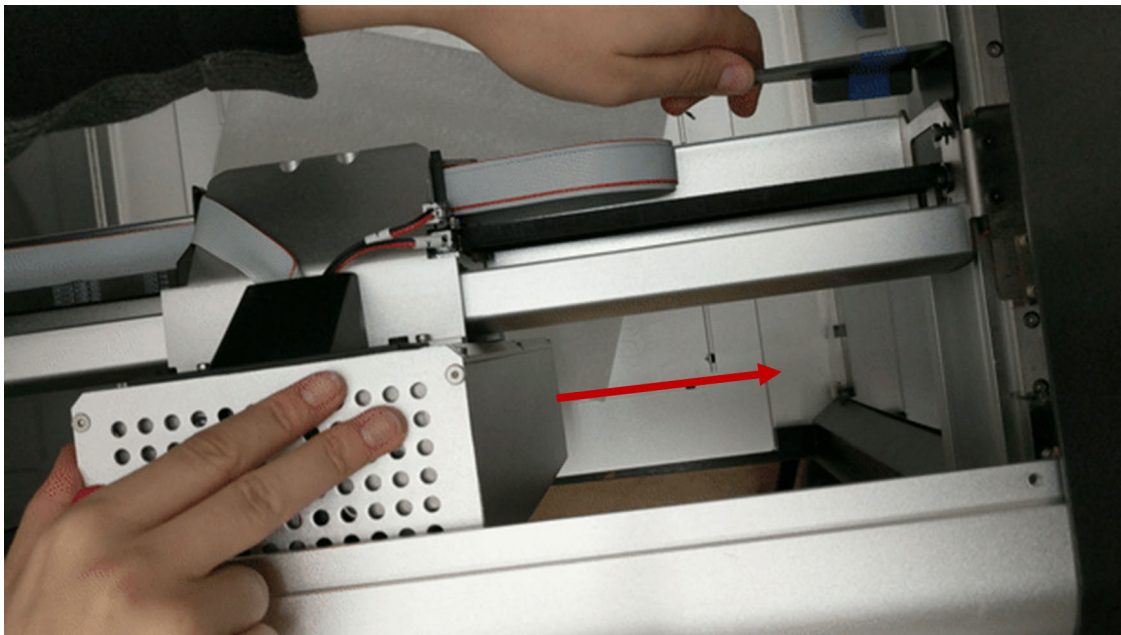
Slide the gantry to the right side.

Make sure the screw holes in the robot align with the holes in the bracket (pictured below).
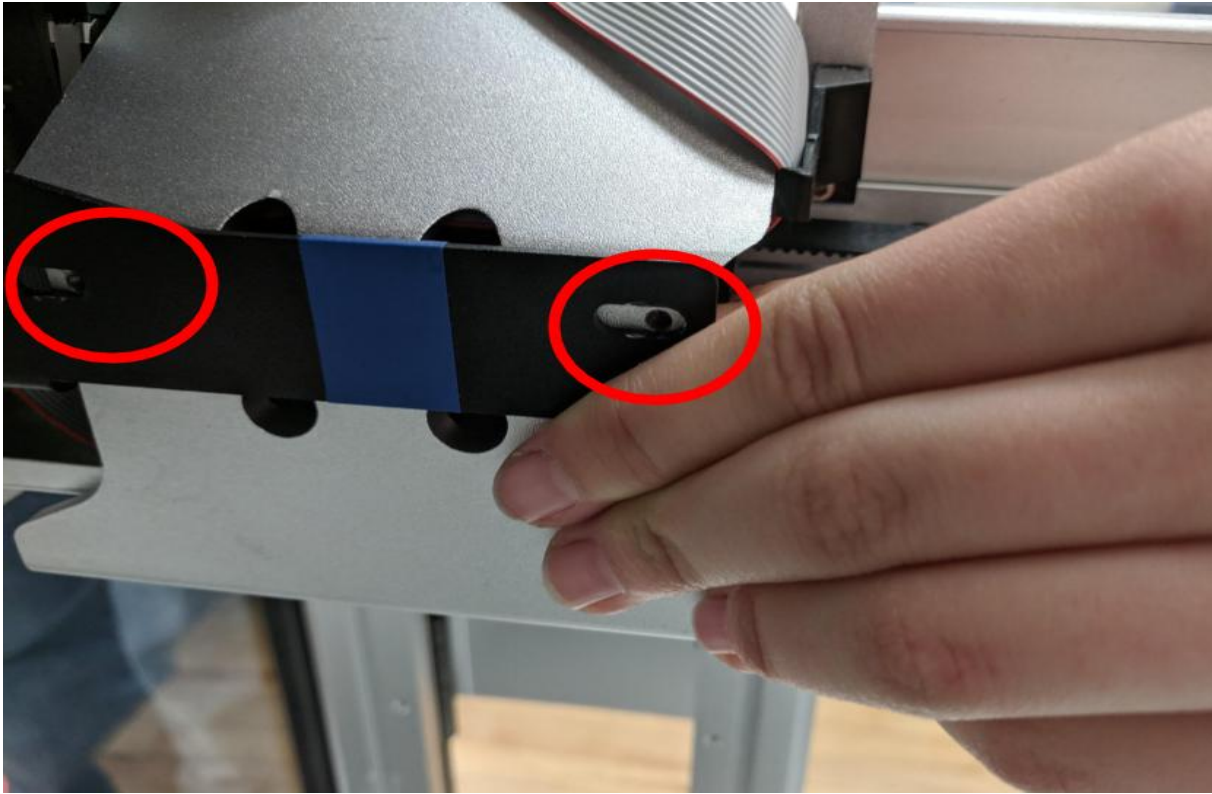
Bracket holes highlighted by red circles.

5. Lightly secure the remaining two screws into the holes.
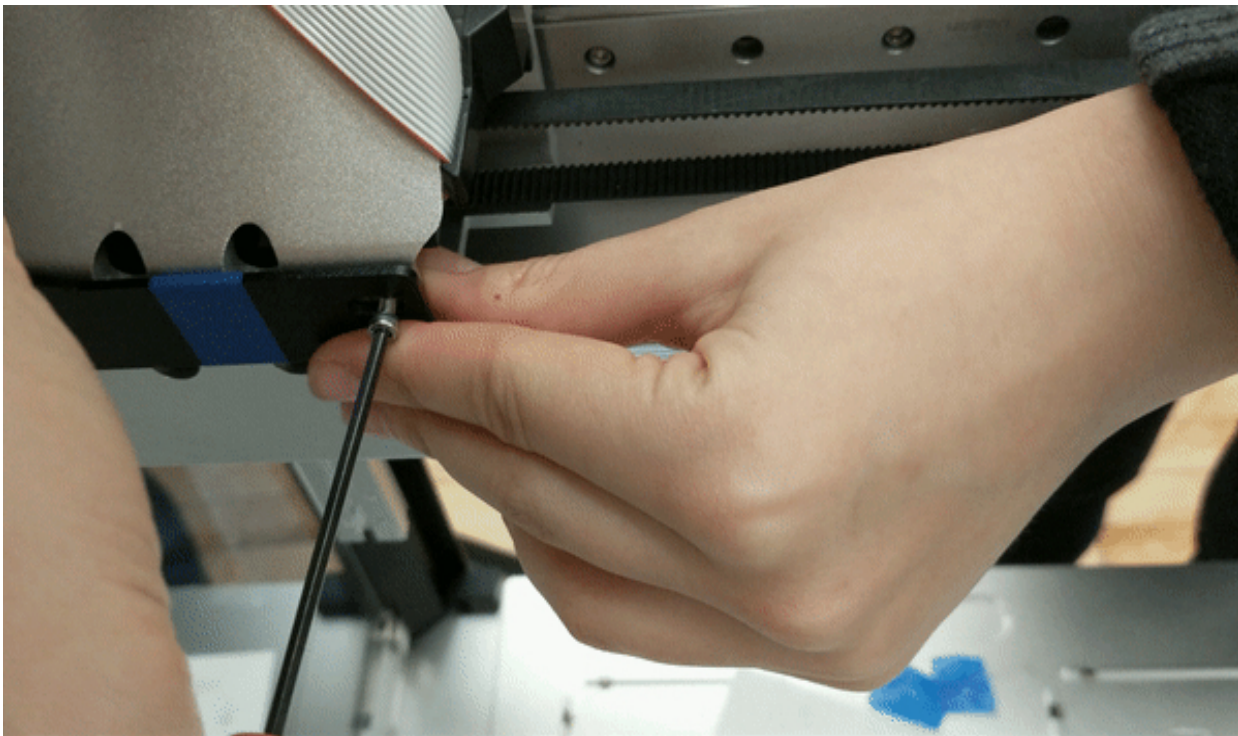
Lightly secure the screws.

6. Tighten all three screws.
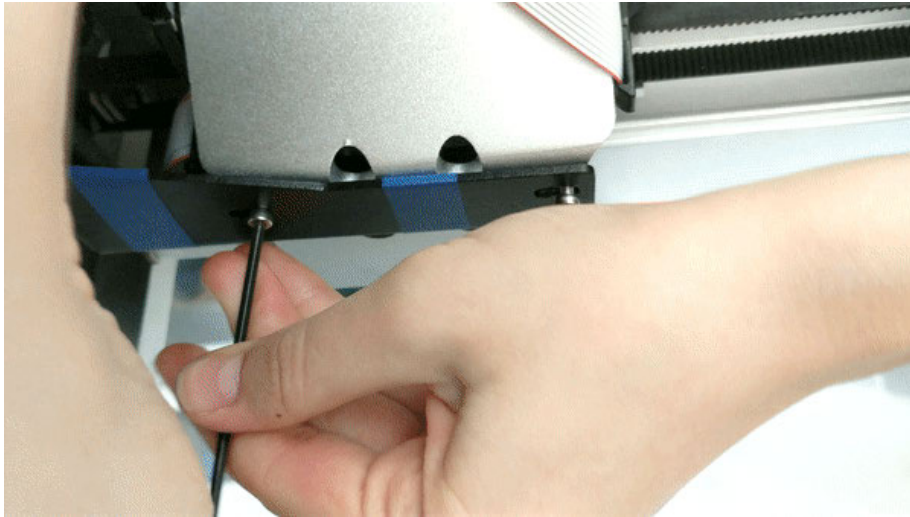
Tighten screw # 1.
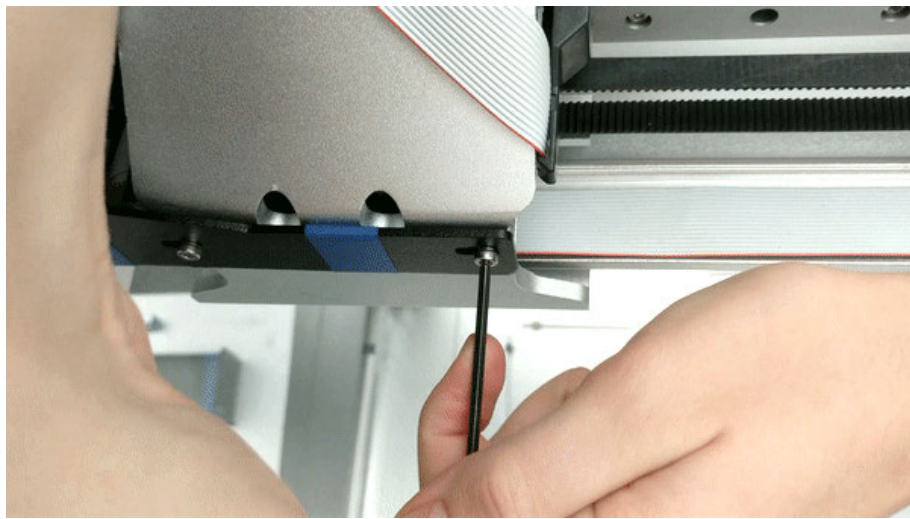
Tighten screw # 2.
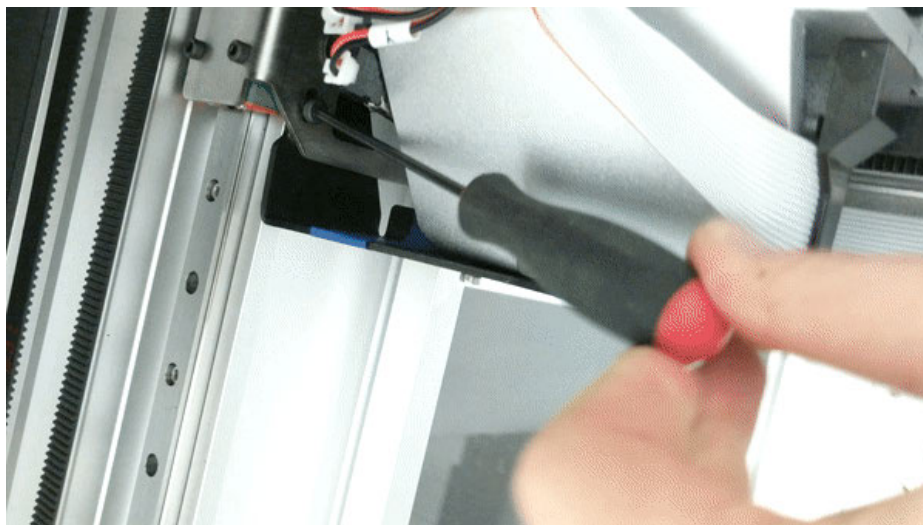
Tighten screw # 3.

Stage 2

1. Gather two of the screws, the two nuts, and the bracket for stage 2.
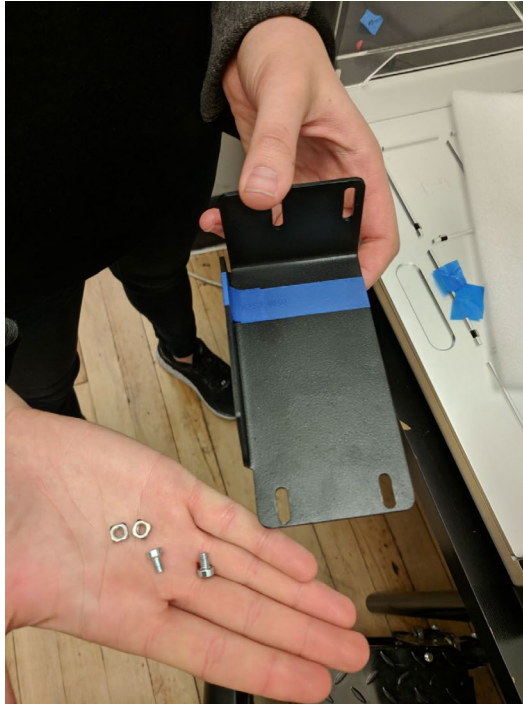
Parts used in Stage 2.

Thread the screws through the holes on the short side of the bracket. Very loosely secure the screws with the nuts. The nut should be facing the inside of the bracket, and the heads of the screws should be facing outwards (pictured below).
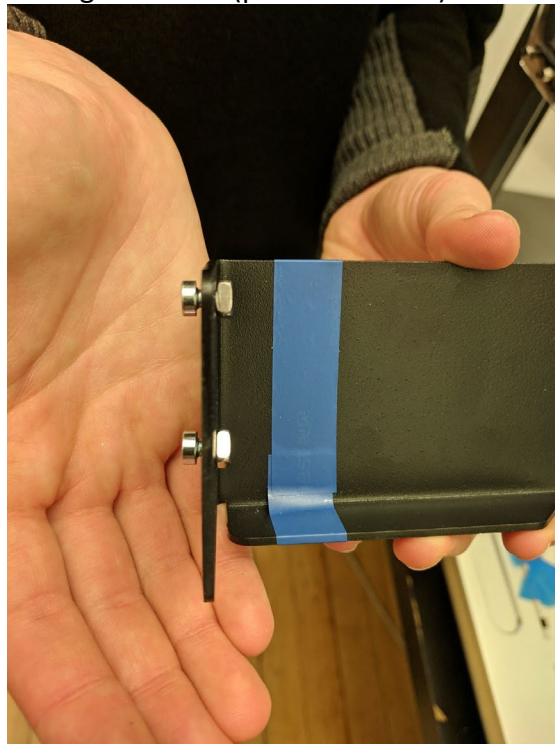
Bracket with screws.

2. ⬜ **Note**: Thread both nuts through the notch behind the right side of the gantry (marked by an arrow in the figure below).
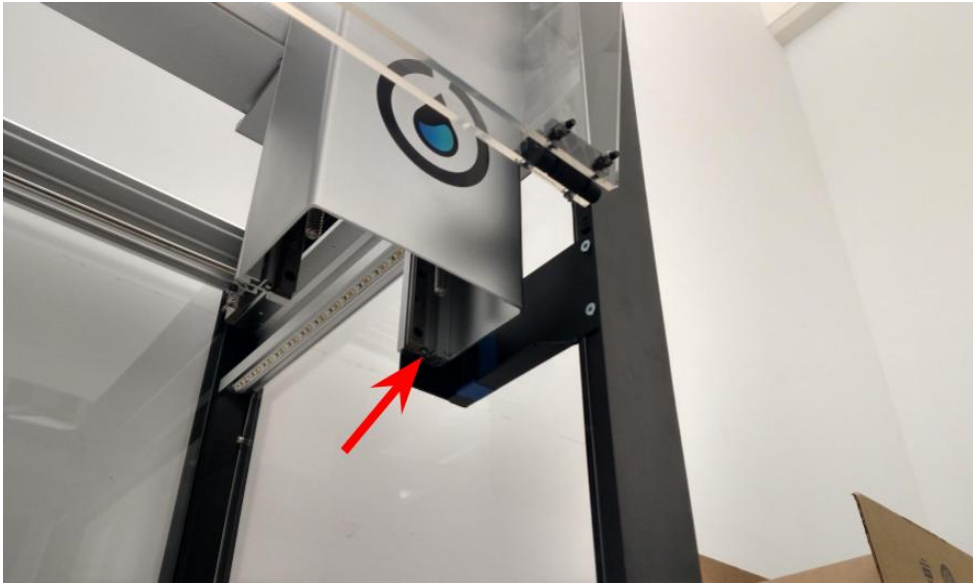
Image of OT-2 bracket.

Close up image of OT-2 bracket.

3. Raise the bracket until the holes in the bracket align with the holes in the side of the robot.

Image of OT-2 bracket.

Hold the bracket in place. Lightly secure the two remaining stage 2 screws through the bracket into the side of the right side of the robot.

4. Tighten all four Stage 2 screws.
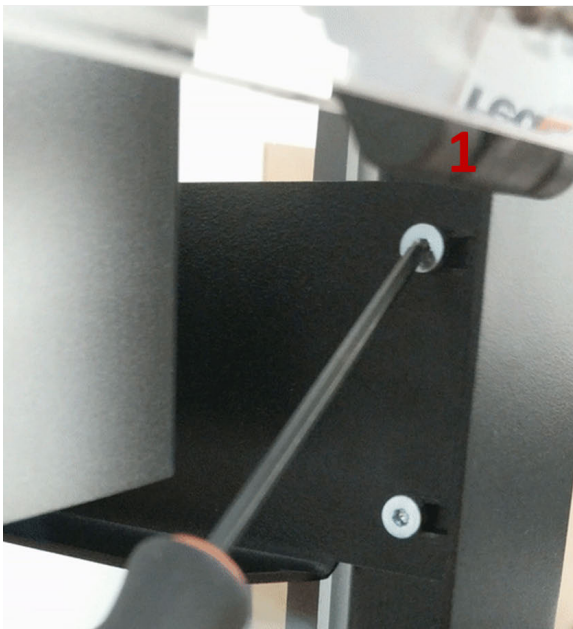
Screws 1 & 2.
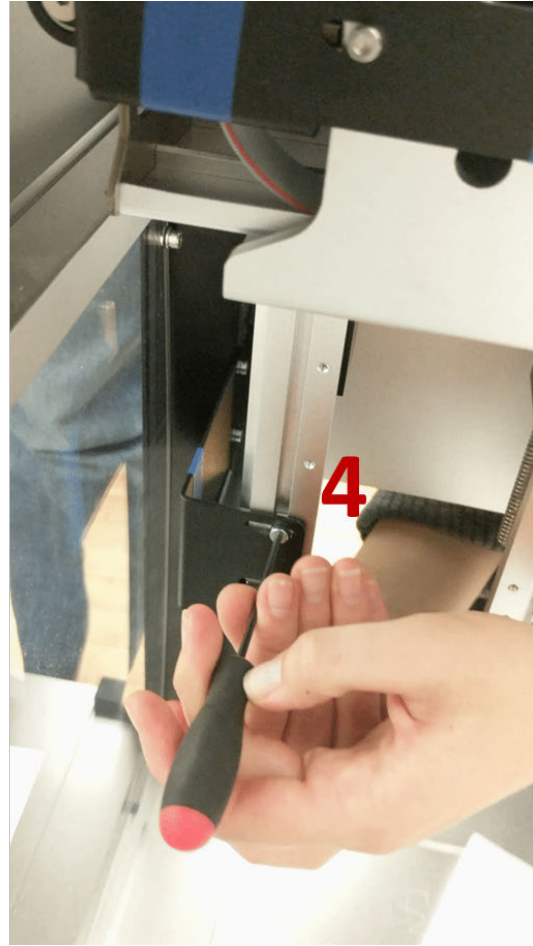
Screws 3 & 4.

Stage 3

1. Take the small bracket and screw for stage 3.

Parts used in Stage 3.

2. Insert the bracket behind the thin sheet metal piece on the left side of the robot. Make sure the blue tab is parallel to the back wall of the robot and not rotating downwards.
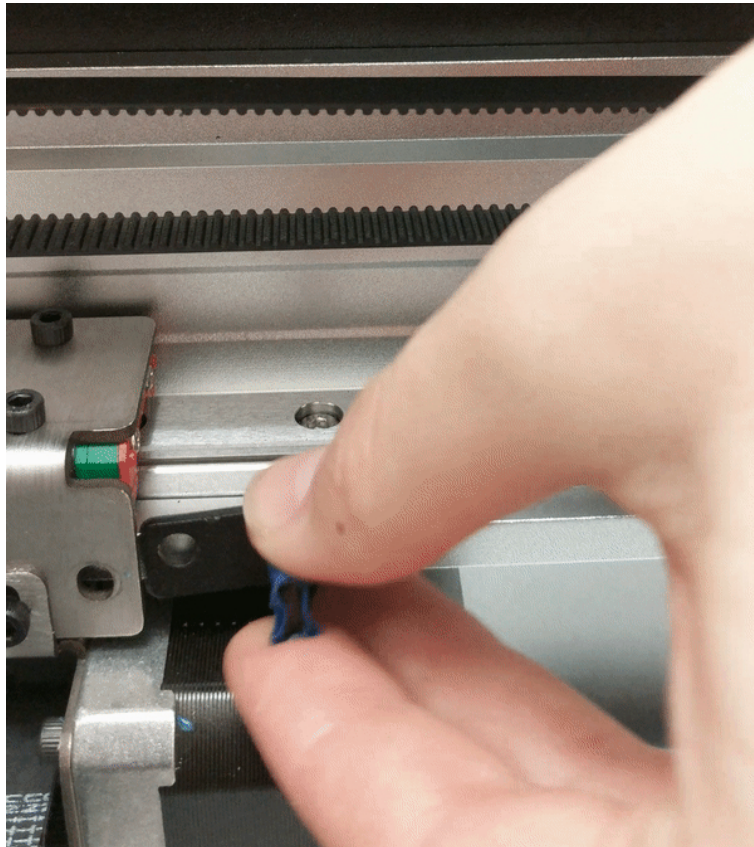
3. Tighten the screw through the sheet metal piece into the small bracket.
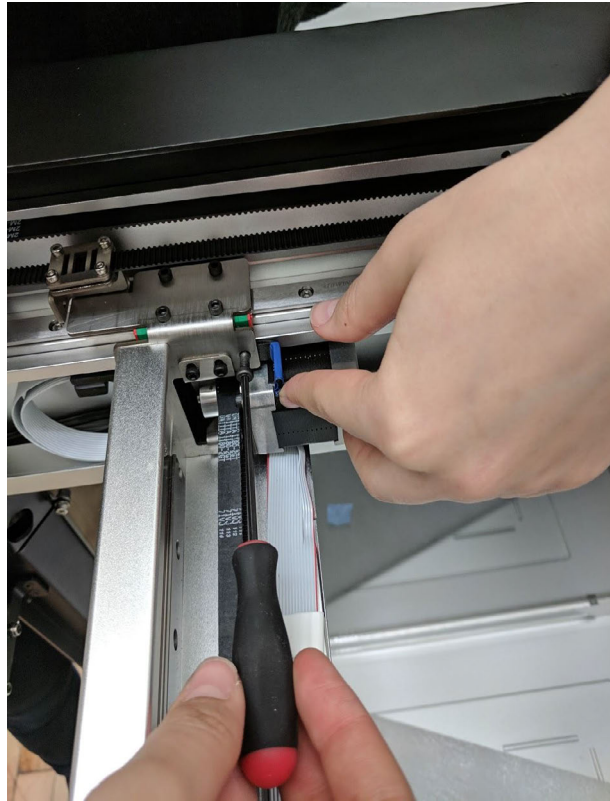
Tighten the screws as indicated in the picture above to screw the sheet metal piece into the small bracket.

The robot is now locked and ready for transportation. Here are some zoomed out views of a locked OT-2 for reference.

Rear view of stage 1 and 2.

View of stage 2.
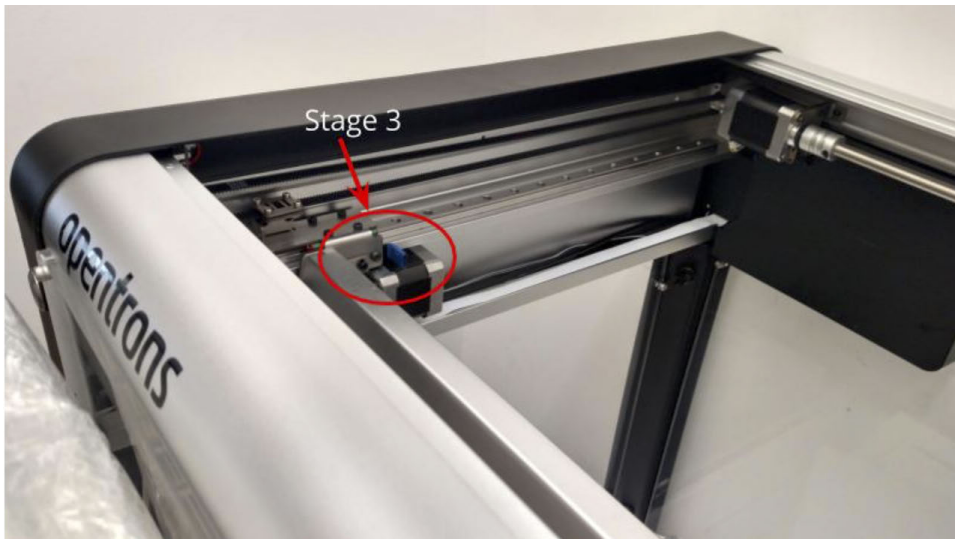
View of stage 3.

Chapter 4 | Opentrons System Description

4.1 System Overview

The topics below describe the front and back panel of the OT-2.

Front Panel
The figure and table below describe the callouts for the front panel of the Opentrons Liquid Handler.
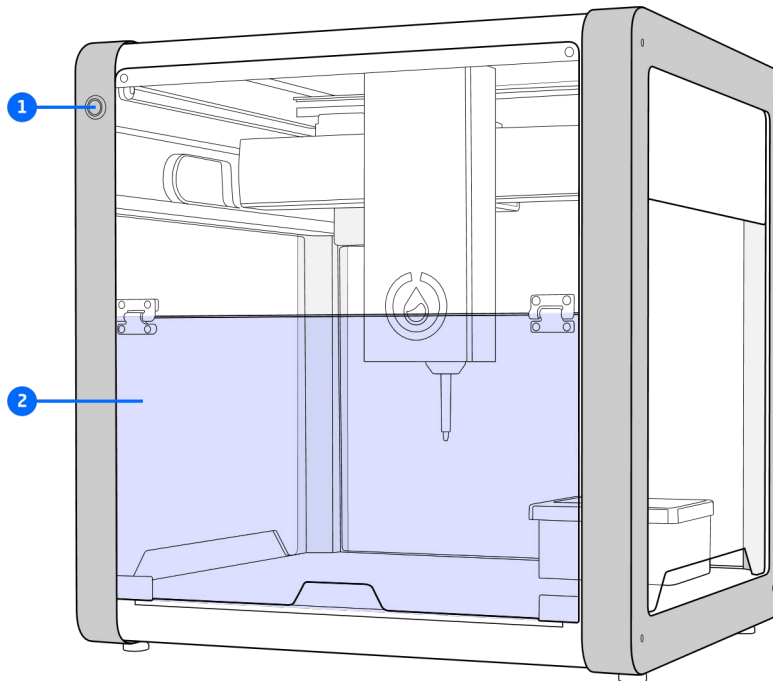
Figure 4.1.1: OT-2 front panel

Callout table for status light and door.

Callout Number	Name	Description
1	Status Light	Indicates the power status of the OT-2. When powered on and initialized, the status light will turn a solid blue color.
2	Door	Polycarbonate door

Back Panel
The figure and table below describe the callouts for the back panel.
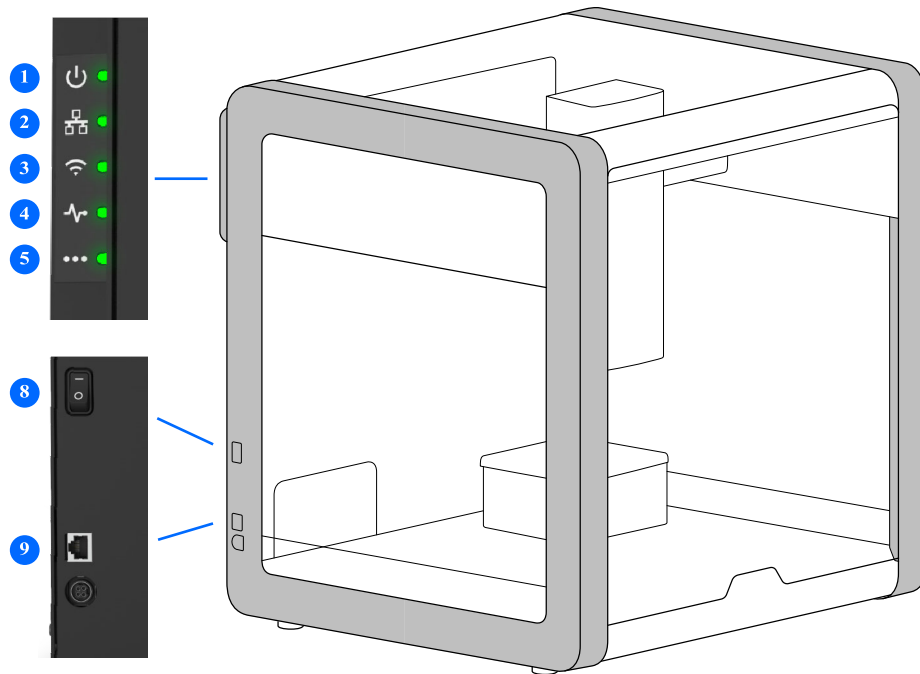
Figure 4.1.2: OT-2 back panel

Table 4.1.1: Callout table for OT-2 back panel

Callout Number	Name	Description
1	Power	Light should stay on and remain solid if the OT-2 is turned on and initialized.
2	Ethernet	Light should be solid when the robot is connectable over Ethernet.

3	Wi-Fi	Light should be solid when the OT-2 has connected to a Wi-Fi network.
4	Heartbeat	Light will flash once every few seconds if the robot's operating system has successfully booted.
5	Extra	Light should remain off for now (reserved for future use).
6	Backpack	Backpack holds the: Central routing board, Raspberry Pi, and USB hub. The backpack is covered by a metal panel, removable with 4 screws.
7	4 USB Ports	4 USB ports available for connecting modules.
8	Ethernet Port	1 Ethernet port for connecting the OT-2 to a computer.
9	Power Switch	Single-pole on-off rocker switch with curved black actuator.

4.2 Component Description
Each of the components that the user may interact with are described in detail in the topics below.

Status Indicator Lights
The OT-2 has five (5) status indicator lights, known as LEDs, located on the side of its external electronics enclosure. This enclosure is located on the upper left side of the back panel of the robot. These status lights allow a user to quickly assess if the Opentrons Liquid Handler is operating normally or if it needs operator attention.

Front Status Lights
The blue light on the robot is a status indicator. When the robot is initialized, the light will turn blue.
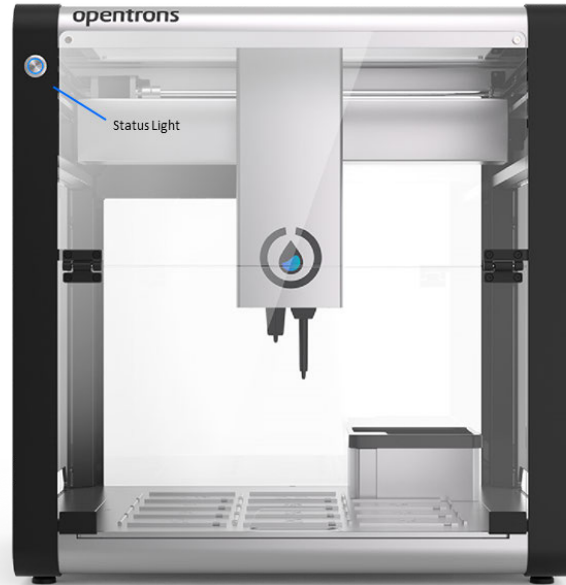
Status Light

Back Status Lights
The table below lists the callouts for back light status.

Symbol	Name	Description
	Power	The power light should stay on and remain solid if the OT-2 is turned on. 1. Light on: This light **should always be on and solid if the robot is turned on**. **Note**: This light is hardwired to the 5V power supply of the Raspberry Pi. 2. Light off: If the OT-2 is on and this light is not illuminated, either the power cable is loose/unplugged, or a serious electrical issue has occurred, and hardware likely needs to be replaced.

	Ethernet	The Ethernet light should be solid when the robot is connectable over Ethernet.
		1. Light on: This light will be illuminated if the Raspberry Pi has successfully assigned an IP address to its Ethernet (wired) interface. This indicates:
		  a. The Raspberry Pi recognizes that the cable is plugged in and that the other end is connected to something that's turned on.
		  b. The Raspberry Pi has assigned an IP address to its Ethernet interface through the usual means (either

		self-assigned a link-local address, self-assigned a static IP address, or received a DHCP address from the network).
		2. Light off: The OT-2 is not currently able to establish a connection over Ethernet.
		a. This is normal for a short period (~1 min) after booting up or after a cable has just been plugged in.
	Wi-Fi	The Wi-Fi light should be solid when the OT-2 has connected to a Wi-Fi network. Light on: This light will be on if the Raspberry Pi's Wi-Fi interface is connected to a network and has been assigned an IP address. Light off: If this light is off, it indicates that the OT-2 is not currently on a Wi-Fi network, which means the Opentrons App is not currently able to establish a connection over Wi-Fi.
	Heartbeat	The heartbeat light will flash once every few seconds if the robot's operating system has successfully booted. 1. Light flashing (alternating between on and off): This light will flash intermittently which indicates the operating system of the OT-2 is running normally. a. **Note**: Individual components of the robot may still fail and this light will continue to blink normally. 2. Light solid off or solid on: If this light is not flashing, the OT-2's operating system has suffered a failure and the system is likely not connectable.
	Extra	This light should remain off for now (reserved for future use).

User-Accessible Components of the OT-2

Status Light

Y Axis

X Axis

Pipette Carriage

Left Pipette Mount

Right Pipette Mount

Electronic Single-Channel Pipette

Electronic 8-Channel Pipette

Trash bin
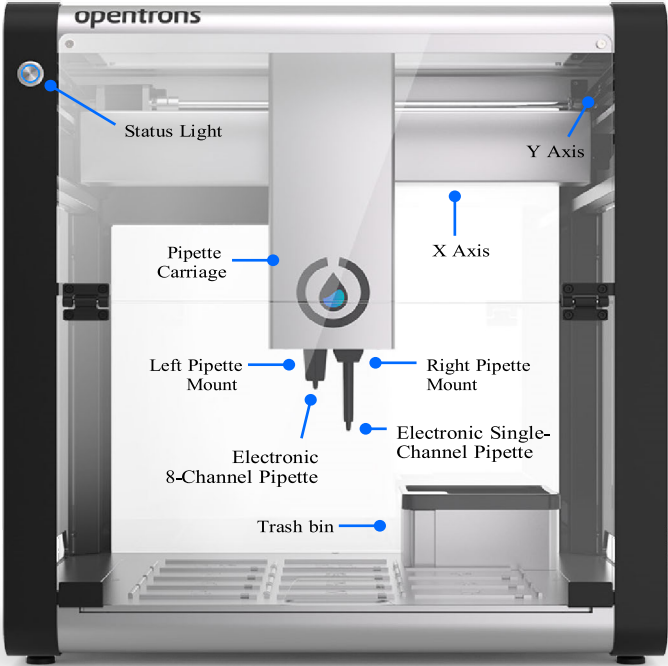
Figure 4.2.1: First set of user accessible components of the OT-2

Trash cover ❻

Stationary deck ❶

Door switch ❷

❺ Tip probe cover

❹ Trash base

❸

Removable deck
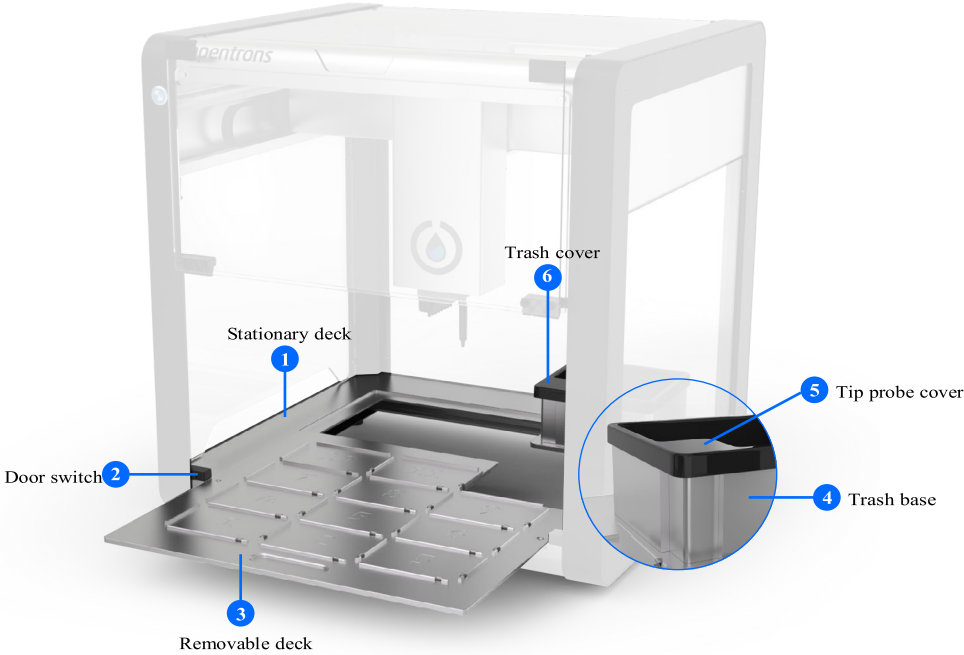
Figure 4.2.2 Deck components

Table 4.2.1: Callout table for accessible components of the OT-2.

Callout Number	Name	Description
1	Stationary Deck	Clear anodized aluminum
2	Door Switch	An auto-stop feature to reduce the risk of injury when either the top window or front door are open. The auto-stop feature is disabled by default.
3	Removable Deck	11 SLAS/ANSI deck positions
4	Trash Base	Clear anodized aluminum
5	Tip Probe Cover	Holds limit switches used for homing the pipettes.
6	Trash Cover	Removable polycarbonate black trash bin.
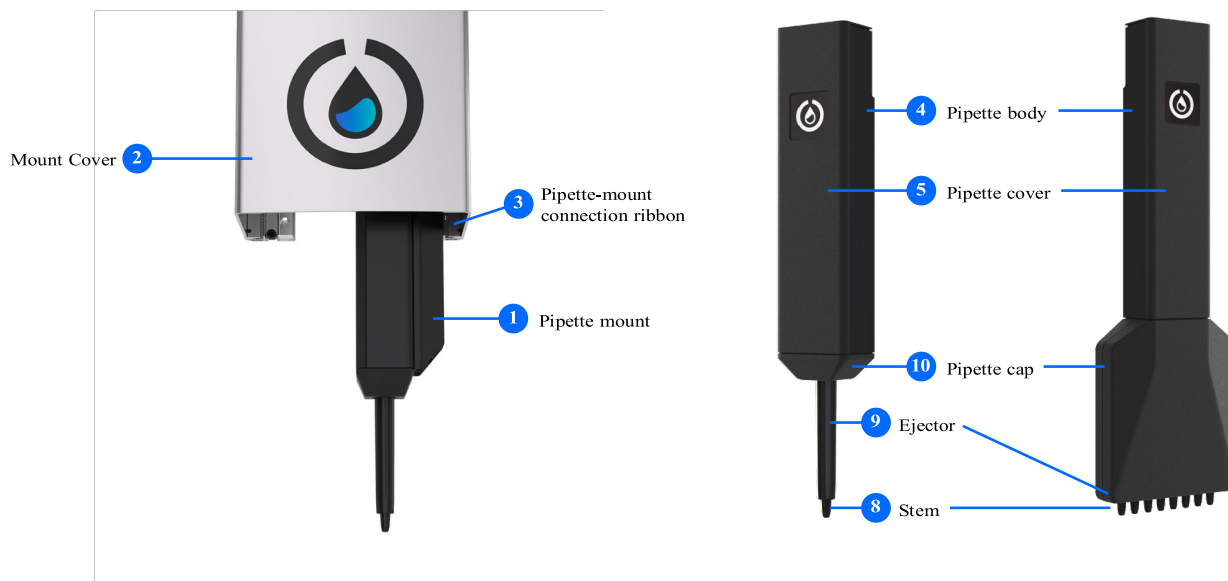
Figure 4.2.3 Single and 8-channel pipette components

Callout Number	Name	Description
1	Pipette Mount	Two pipette mounts. Each pipette mount can hold one Opentrons pipette. The pipette screws into the mount, and the OT-2 controls the pipette electrically through a cable.
2	Mount Cover	Composed of black PC-ABS
3	Pipette-Mount connection ribbon	The OT-2 controls the pipette through a white ribbon cable.
4	Pipette Body	The body of the Opentrons pipette is composed of ABS plastic. For pipette body dimensions, see the Single-channel GEN2 or 8-Channel GEN2 brochure.
5	Pipette Cover	Composed of black PC-ABS
6	8-Channel GEN2 Pipette Tip Ejector	When the driver shaft moves sufficiently far down it will trigger the tip ejector and eject the pipette tip.
7	Single-Channel GEN2 Pipette Stem	Composed of black PC-ABS
8	8-Channel GEN2 Pipette Stem	Composed of black PC-ABS
9	Single-Channel GEN2 Pipette Tip Ejector	When the driver shaft moves sufficiently far down it will trigger the tip ejector and eject the pipette tip.
10	Pipette Cap	Composed of black PC-ABS

4.3 Working Area Diagrams

Below are drawings that lay out the working area of the OT-2 as well as the max height distance a piece of labware can be depending on the pipette.

Opentrons Pipettes Lowest Bounds
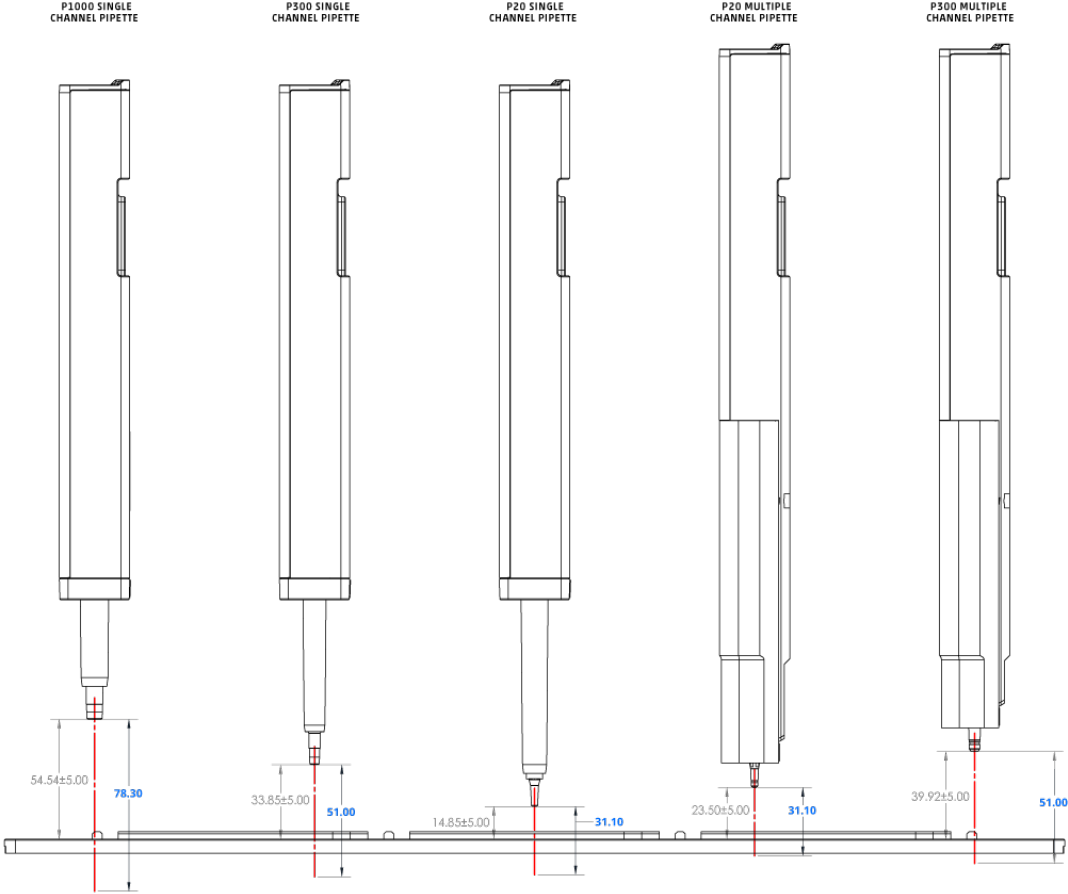
Figure 4.3.1: OT-2 pipettes mounted at minimum Z position relative to the robot deck

Opentrons Pipettes Highest Bounds
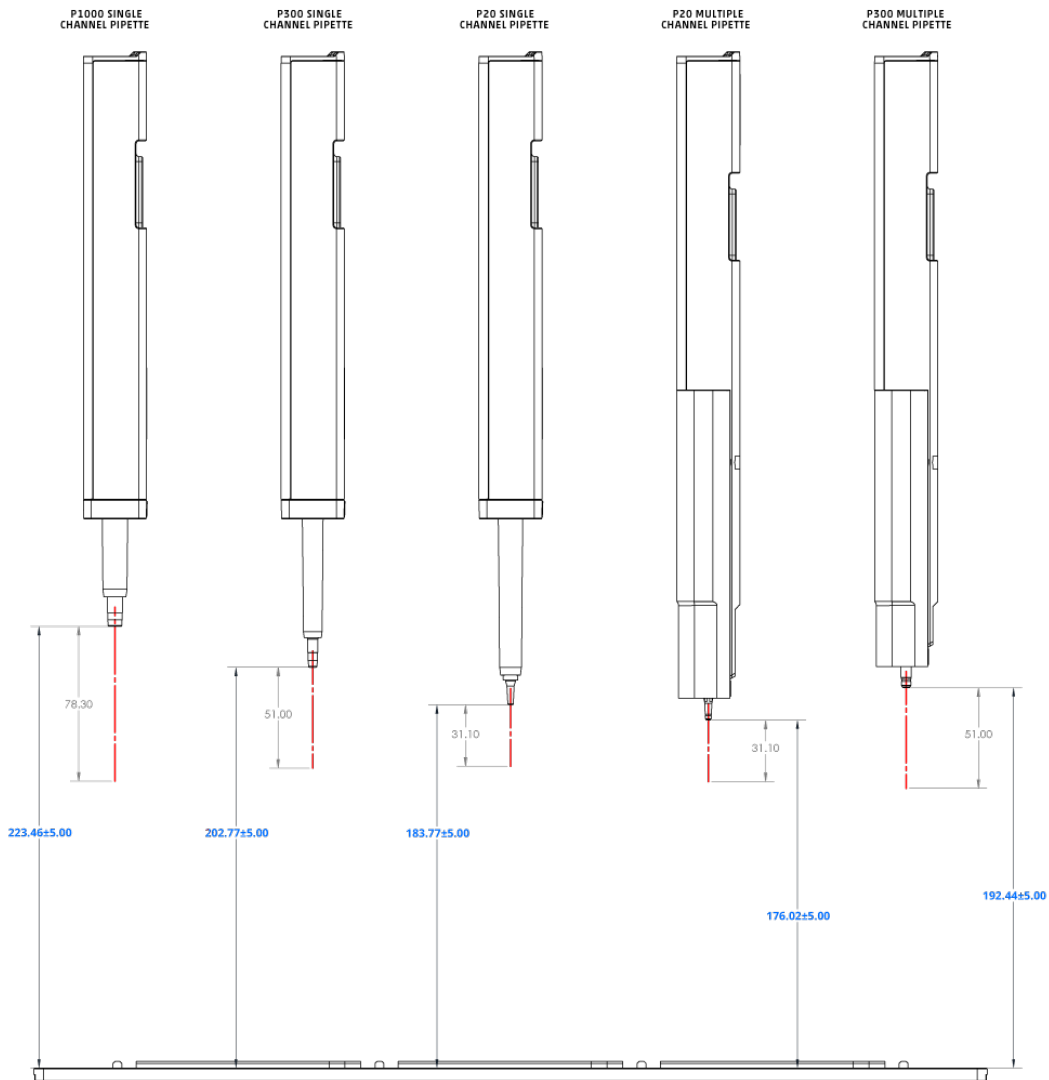
Figure 4.3.2: OT-2 pipettes mounted at maximum Z position relative to the robot deck

4.4 Ethernet Connection

The Opentrons Liquid Handler includes several communication connections to the lower left below the power switch. The Ethernet cable connects the Liquid Handler directly to the client PC or to a network. This connection allows you to monitor and control the Opentrons Liquid Handler. The OT-2 eliminates the internal USB-to-Ethernet converter, which means the Ethernet port on the outside of the robot connects directly to the Raspberry Pi's Ethernet port internally.

Ethernet Port Location

There are four panels that enclose the OT-2: front, left, back, and right. If facing the front panel of the OT-2 (with the swinging door), the Ethernet port is located on the left side panel, in the lower left, below the power switch.
In the image below, note the orientation of the OT-2. Check the left panel of the robot:

Ethernet port

Front panel

Left panel
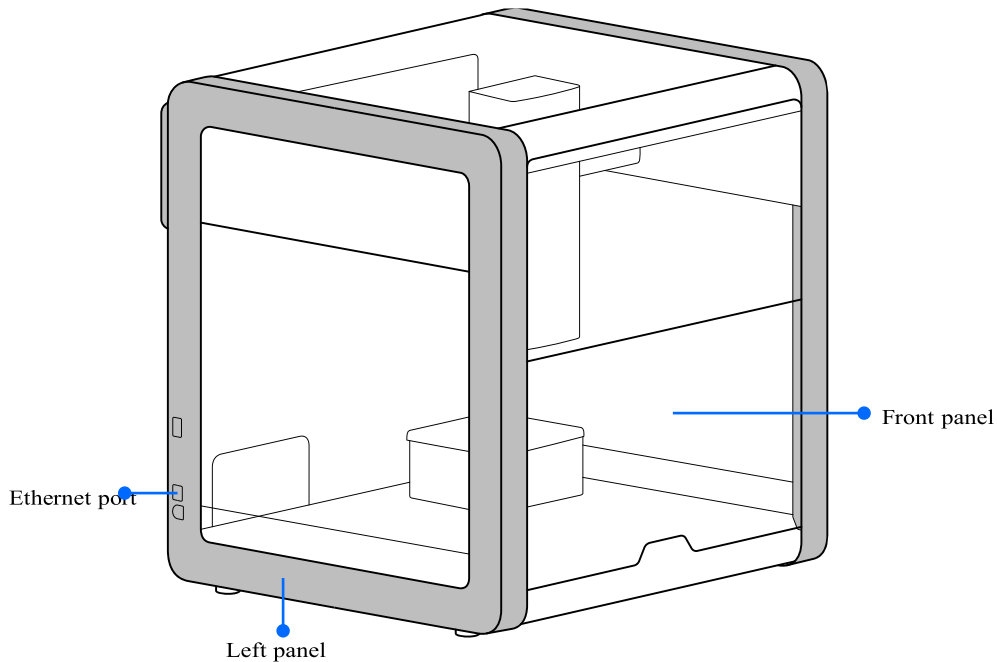
Figure 4.4.1: Ethernet location on the OT-2

4.5 Power Consumption Considerations for the OT-2

The total power consumption of the OT-2 depends on the specifics of the protocols run on the robot. Power consumption ranges from 100 W to 180 W when running a protocol. The OT-2 consumes 90-120 W when idle.

4.6 OT-2 Motor Specifications

The X, Y, Z axis are currently powered by Nema 17 two-phase stepper motors.

```
Default_max_speeds = {'X': 600, 'Y': 400, 'Z': 100, 'A': 100}

Serial_speed = 115200

Default_current = {'X': 1.5, 'Y': 1.75, 'Z': 1.0, 'A': 100}

Low_current = {'X': 0.3, 'Y': 0.3, 'Z': 0.1, 'A': 0.1}

High_current = {'X': 1.5, 'Y': 1.75, 'Z': 1.0, 'A': 1.0}
```

Figure dimensions

unit: mm
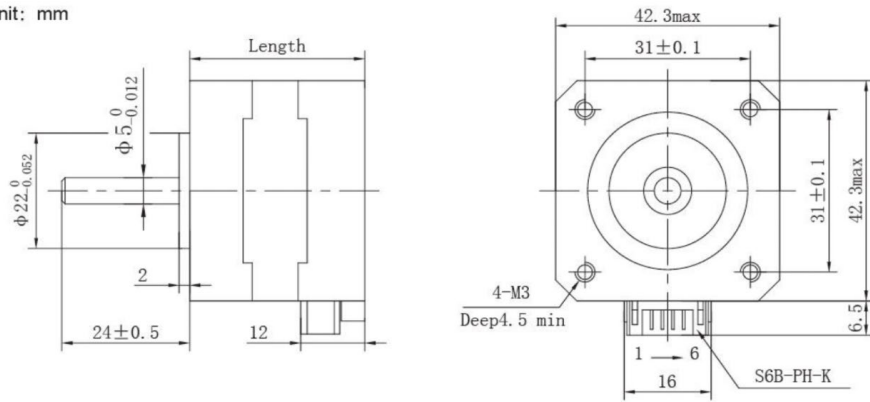
Wiring diagram & Stepping sequence

Bi-polar
Wiring diagram
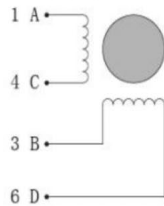
Stepping sequence
Bi-polar full step

PHASE / STEP	A	B	C	D
1	+	+	−	−
2	−	+	+	−
3	−	−	+	+
4	+	−	−	+

CW   CCW

电机旋转方向以出轴安装面为参考面

Uni-polar
Wiring diagram
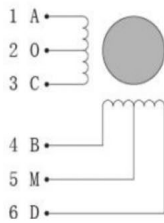
Stepping sequence
Uni-polar full step

PHASE / STEP	A	B	C	D	0	M
1	−	−			+	+
2		−	−		+	+
3			−	−	+	+
4	−			−	+	+

CW   CCW

VM: 24VDC; Driver: 2H422; Full step

42SHD0512 ——
42SHD0002 ——
42SHD0216 ——
42SHD0441 ——
42SHD0001 ——
42SHD0202 ——
42SHD0404 ——
42SHD0801 ——

VM: 24VDC; Driver: 2H6422; Full step

42SHD0009 ——
42SHD0219 ——
42SHD0029 ——
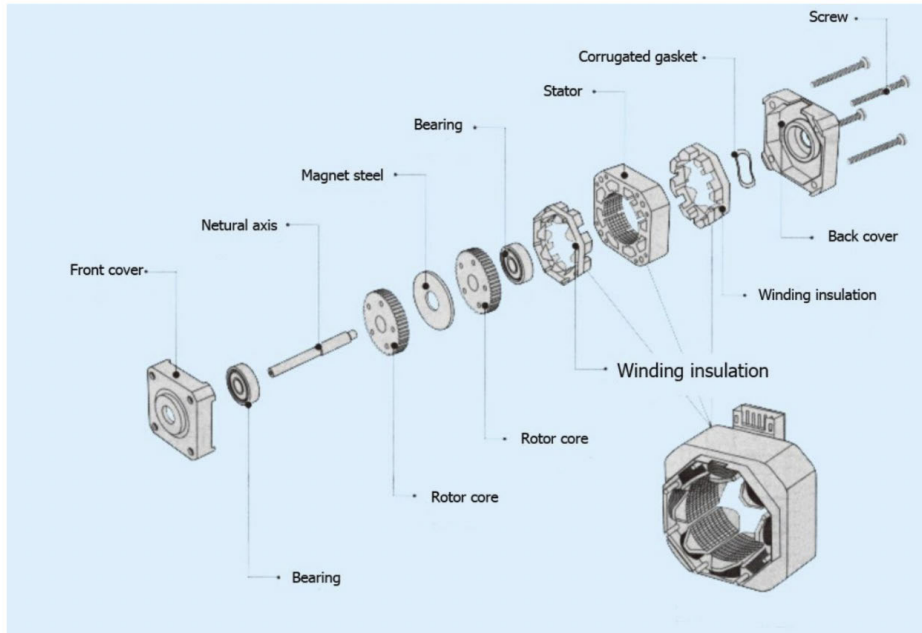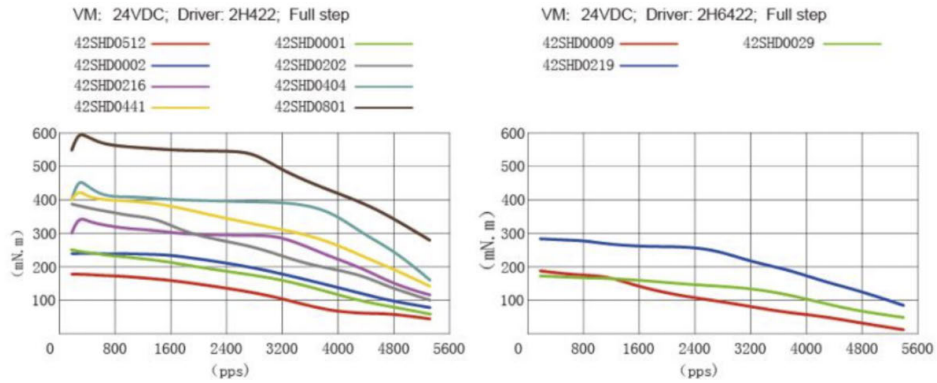
The electronic pipettes are powered using captive Can Stack 25s, which are smaller linear actuators.
The specifications and defaults are listed below:

Default_max_speed = {'B': 50, 'C': 50} mm/s
Default_current = {'B': 0.1, 'C': 0.1}
Low_current = { 'B': 0.1, 'C': 0.1},
High_current = {'B': 0.5, 'C': 0.5}

CAPTIVE

15° STEP ANGLE

Ø25mm Can Stack Stepper Motor (5V/15°) Force vs Linear Speed, 24VDC Chopper Drive

J (mm/s)	10	20.2	30.4	40.8	50.8	60.8
B (mm/s)	5	10.2	15.2	20.4	25.4	30.4
AA (mm/s)	2.5	5.1	7.6	10.2	12.7	15.2
RPM	250	500	750	1000	1250	1500
PPS	200	400	600	800	1000	1200

Ø25mm Can Stack Stepper Motor (12V/15°) Force vs Linear Speed, 24VDC Chopper Drive

J (mm/s)	10	20.2	30.4	40.8	50.8	60.8
B (mm/s)	5	10.2	15.2	20.4	25.4	30.4
AA (mm/s)	2.5	5.1	7.6	10.2	12.7	15.2
RPM	250	500	750	1000	1250	1500
PPS	200	400	600	800	1000	1200

4.7 OT-2 Serial Number

Each OT-2 has a unique serial number. An example of the serial number format is illustrated below:
1OT2CEP20190924A01
The product serial number is located on the back of the OT-2:

If the serial number cannot be viewed by the user, the serial number can be viewed in the Opentrons App. The serial number will be displayed when a user connects to the OT-2.

Use the OT-2 serial number to determine manufacture date

The OT-2 serial number includes the date your OT-2 was manufactured, in YYYY-MM-DD format.
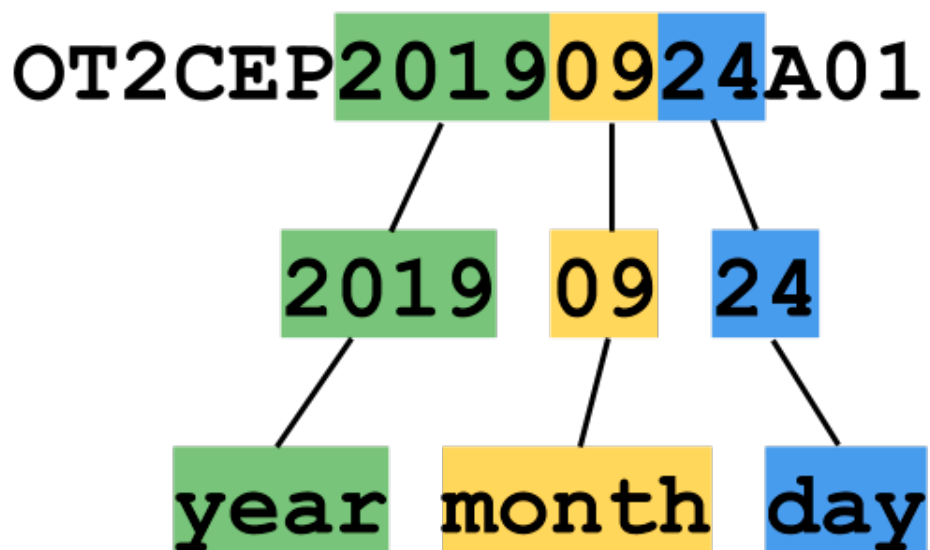
4.8 Opentrons Modules

The following section covers the Opentrons Liquid Handler modules that can be integrated into protocols run on the robot.

Opentrons HEPA Module

The Opentrons HEPA (High-Efficiency Particulate Air) Module is a positive pressure clean air system for use with the Opentrons OT-2 liquid handling robot. The HEPA Module introduces a steady stream of filtered air into the OT-2 enclosure, displacing contaminants and providing a positive pressure air boundary to prevent the introduction of airborne contaminants.

The HEPA Module is an attachment that is placed on top of an OT-2. When installed, the HEPA Module removes 99.99% of 0.3 µm DNA-containing particulates and biological contaminants, creating a clean work environment inside the OT-2.

The following section gives a system overview of the module and guides the user through HEPA module integration with the OT-2. For a detailed overview of the HEPA module view the *HEPA Module Manual* or *HEPA Module White Paper*.

Note: The HEPA module does not display in the Opentrons App, and it does not have any in-app functionality.

Note: The HEPA Module is not a biosafety cabinet and should never be used for pathogens. It should not be used for situations with volatiles or any potentially hazardous aerosolized materials.

OT-2 with a HEPA Module attached on top.

Opentrons HEPA Module Specifications

Specification	Detail
Airflow	Vertical downflow; 0.5 m/s - 1.0 m/s
HEPA Filter Grade/Efficiency	99.99% efficiency for 0.3 μm particles
Controls	On/off; Fan speed
Fan Lifetime	3 years of moderate use
Filter Lifetime	1 year of moderate use. Filter performance in the following settings: ● Laboratory setting: 4160 hours (2 years, 8 hrs/day, 5 days/week) ● Clean room lab setting: 6240 hours (3 years, 8 hrs/day, 5 days/week)
Dimensions	625 x 599 x 303 mm (l/w/h)
Weight	60 lbs / 27 kg
Voltage	110V - 220V
Noise level	65 decibels
Input Power	100-130VAC 50/60Hz (145W) or 220-240VAC 50/60Hz (145W)
AC Mains Fluctuations	+/- 10% of Input Voltage

Opentrons HEPA Equipment Ratings

Usage Aspect	Conditions for safe use
Over-voltage category	Class II
IEC Cable Rating	250V 10A
Environment	Indoor use
Ambient Temperature	Recommended 20–24 °C
Wet Location	Not intended to be used in wet locations
Noise Level	65 dB average
Relative Humidity	Up to 80%
Altitude	Up to 2,000 meters above sea level
Ingress Protection	IP50
Pollution degree	2

Opentrons HEPA Module OT-2 Integration

The intended use of the HEPA Module is as optional equipment with the Opentrons OT-2. It is designed to clean the air of the OT-2 workspace using a HEPA filter to reduce the amount of particles in the workspace. The filter fan continuously will move air through the workspace and vent through the sides of the OT-2 to maintain a clean workspace.

Warning: The HEPA Module doesn't provide operator protection like a fume hood or biosafety cabinet, only sample protection. Therefore, the module shouldn't be used with pathogens. It also shouldn't be used with any hazardous aerosolized materials. To use the OT-2 for these applications, the system must be operated in a biosafety cabinet.

When removing the HEPA module from the package for the first time, the user will encounter a layer of foam containing the HEPA accessories and the HEPA Module underneath.
The HEPA accessories include:
- IEC cable (region specific)
- 8x installation screws (M4x8mm low-profile screws)
- 1x 2.5mm hex key (part of the hex key set that was included with your robot)
- 4x M4 washers

Note: Before installing the HEPA Module, make sure there is 20 cm of clearance around the OT-2 to allow optimal function.
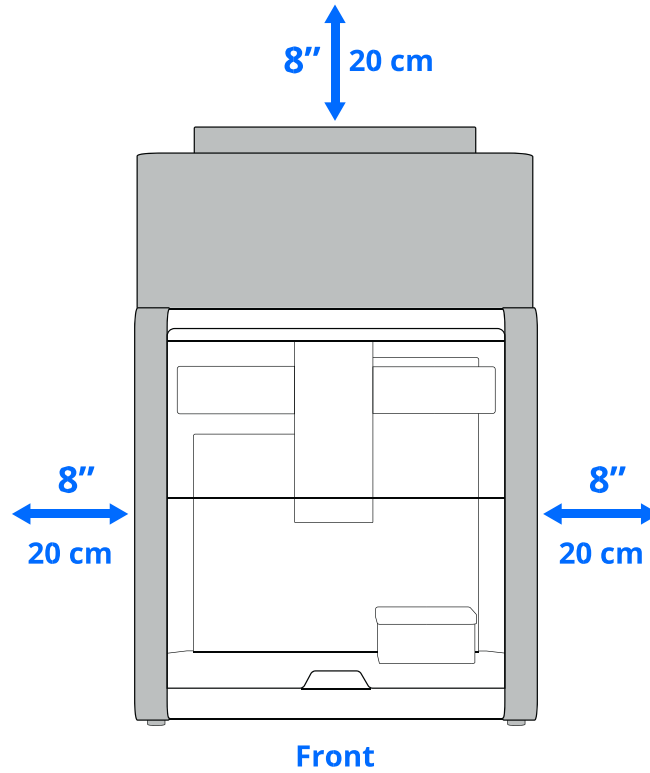
Front
Figure 4.8.1: Working area for the HEPA module install on the OT-2.

1. Remove the top window of the OT-2 and store it somewhere safe.
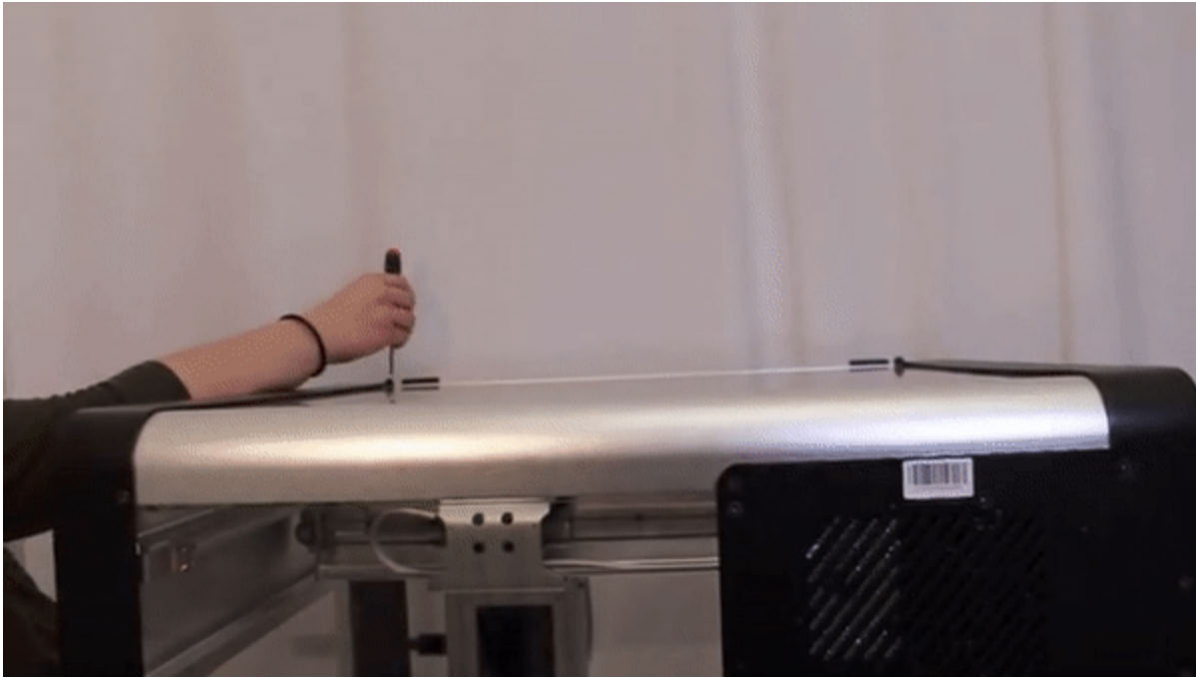
2. With another person, use the four handles on the HEPA Module to lift it out of the box and place it on a clear surface to prevent damage to the module.

 Caution: Risk of injury. The HEPA module weighs approximately 60 lbs. (27 kg). Have two or more people lift and move the unit. Use proper lifting technique to prevent injury.
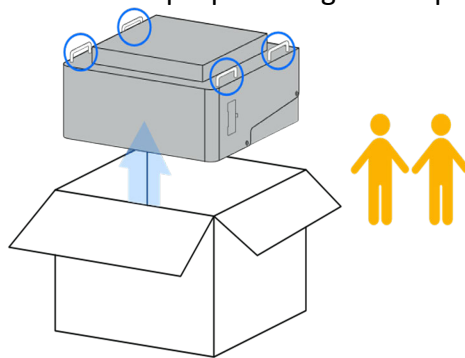
3. There are two parts to the HEPA Module: The adaptor bracket and the main HEPA unit. Using the 2.5mm hex key, loosen the screws indicated on the image below that hold the bracket and main unit together.

REAR
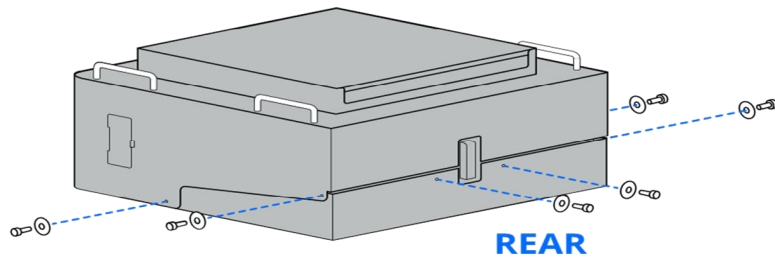
4. With the assistance of a second person, lift the main HEPA unit, removing it from the adapter, and place it on a clean and accessible workspace.
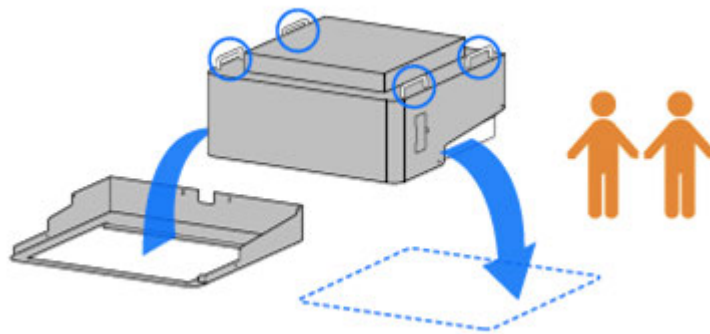
5. Next, place the adapter on the top of the OT-2. The key corners will first be slipped under the sheet metal on the left side of the robot and then the rest can be snapped into place. This can be done with one person.
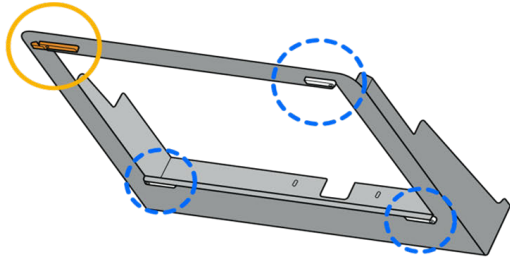
The bottom of the adaptor bracket has 4 key corners.
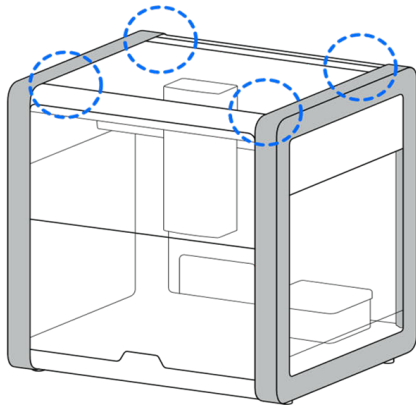
These fit into the surface that support the top window on the OT-2
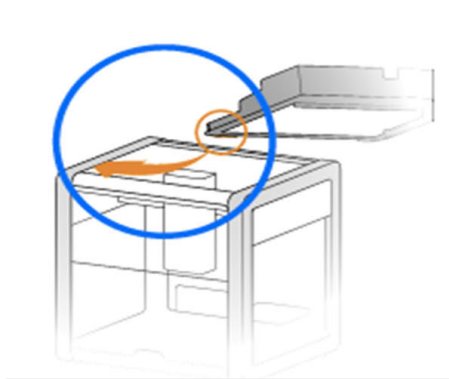
Slip the key corners on the left-hand side of the adaptor bracket under the sheet metal of the robot as shown below.
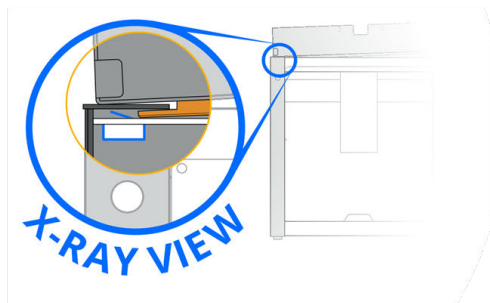
Front view of key corner

X-RAY VIEW

6. Then, take the screws from the installation screw bag and lock the HEPA adapter into the top of the OT-2 using the 2.5 mm hex screwdriver or key. The screws should go where the top window screws went.

⚠️ **Caution:** Only use the screws that go with the HEPA Module! The screws used with the top window are too long to use with the adapter bracket.

7. Using two people, slide the main HEPA unit onto the adapter bracket of the HEPA Module and push until you hit the rear end of the bracket, making sure that the screw holes line up.

⚠️ **Caution:** Be careful, when placing the module in place, fingers can become pinched between the adapter and HEPA unit.
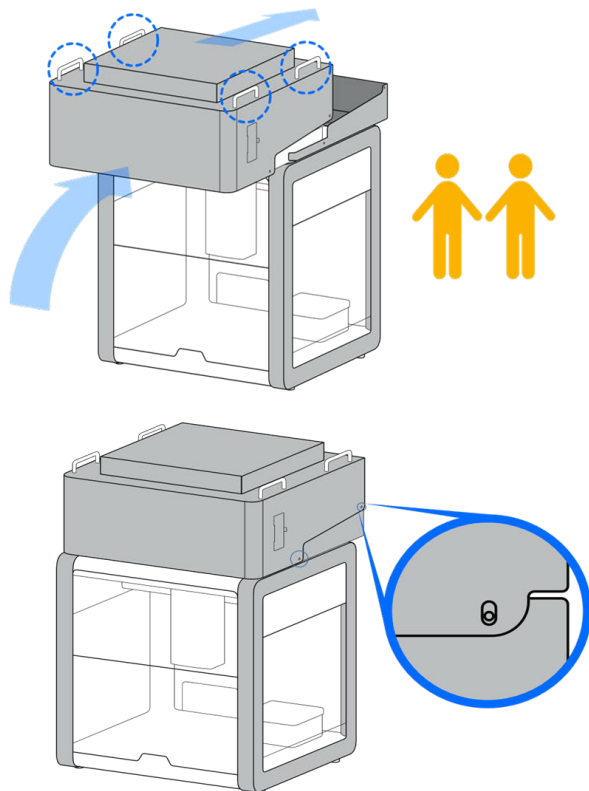
Slide the main HEPA unit onto the adapter bracket of the HEPA Module until you hit the rear end of the bracket.

Screw holes should now line up.

8. Reinstall the screws used to connect the main HEPA unit and adapter together.
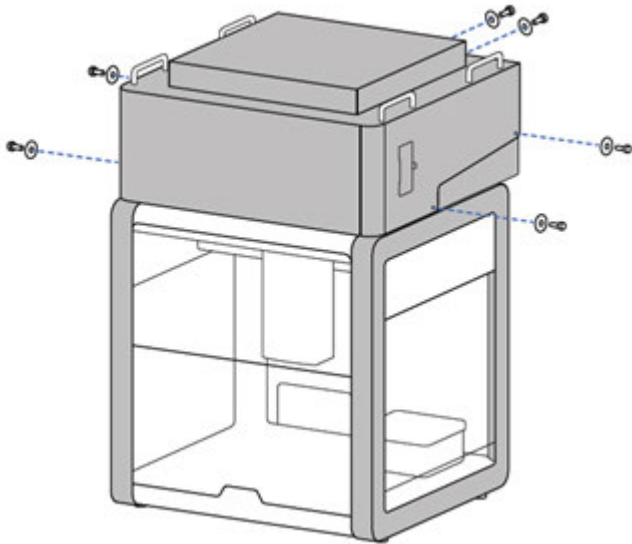
9. Plug the IEC cable into the socket located next to the power knob and the other side into a wall socket. Ensure that the HEPA Module is turned off before plugging in the cable and that the IEC plug and control knob are easily accessible and not blocked.
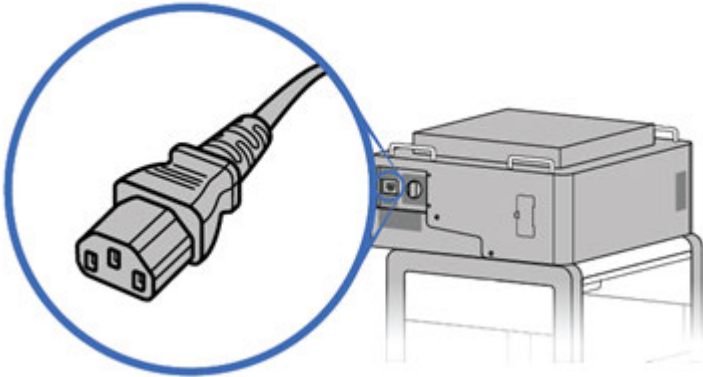
10. Write down the HEPA module installation date to ensure the filter is replaced at the right time.
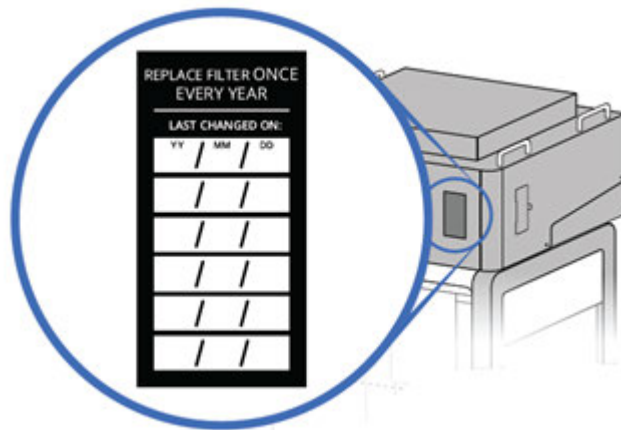
11. Make sure to disinfect any contaminated workspaces in the OT-2 before turning on the HEPA Module. To power on the module, turn the knob clockwise. This will commence airflow through the OT-2. As you turn clockwise, the airflow rate decreases.
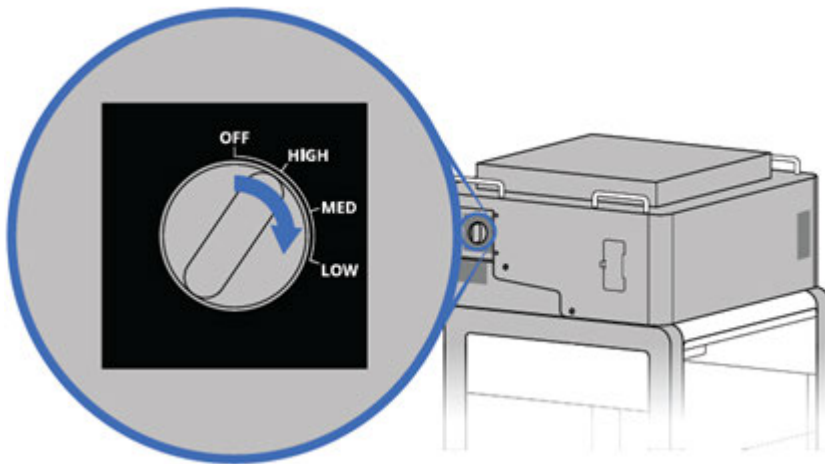
How to use the HEPA Module:

After setting up your HEPA Module, these are the steps to take to make sure it's fully functional:
1. Set up the OT-2 deck for an experimental run. Add all labware and reagents, except anything that is frozen or refrigerated, which can be added to the OT-2 when needed.
2. Close the front door of the OT-2.
3. Allow the HEPA filter to run for 1 hour on high to allow the air inside the OT-2 workspace to be filtered.
4. Run the protocol.

5. After finishing the protocol run, turn off the HEPA filter.

Tip: Refrain from opening the door of the OT-2 as much as possible for best use.

Opentrons Magnetic Module

The Opentrons Magnetic Module is a device that automatically engages and disengages high-strength neodymium magnets to seated well plates within the module housing. The module can readily be used with the Opentrons App. The Magnetic Module comes with two separate adjustable plate brackets for supporting standard and deep well labware. Users can remove and/or swap brackets by unscrewing the module's thumb screw.

The magnetic plate on the module supports many different magnetic bead products for extraction and purification of nucleic acids in the labware shown below. The brackets allow for support of protocols with standard or deep well 96 plates with volumes ranging from 10 µL to 2 mL.
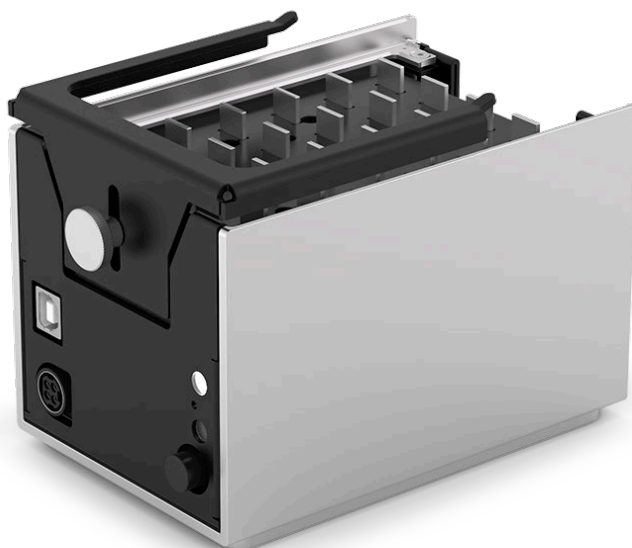
GEN2 Magnetic Module without plates.

Magnetic Module with 0.2 mL 96 well PCR plate.

Magnetic Module with 2 mL deep well plate.

Opentrons Magnetic Module Specifications

Dimensions (w/o bracket)	133 mm x 90 mm x 98 mm (l/w/h)
Weight	3.3 lbs / 1.5 kg
Module Power Input	36VDC, 6.1 A
Power Adapter Input	100-240VAC, 50/60Hz, 4.0A/115VAC, 2.0A/230VAC

Opentrons Magnetic Module Naming Conventions

Application	GEN 1 Naming	GEN 2 Naming
API Load Name	`magnetic module or magdeck`	`magnetic module gen2`
Opentrons Client Application	Magnetic Module GEN1	Magnetic Module GEN2

Opentrons Magnetic Module Key Features

Feature	GEN 1	GEN 2
Image		
Unique features per generation	Units are measured without a defined unit as follows: • Max engage height: 40 • Labware bottom: 0 • Disengaged: -0.5	Alternate magnet layout from GEN1 Units are measured in millimeters as follows: • Max engage height:

		16 mm ● Labware bottom: 0 mm ● Disengaged: -4 mm
States	Engaged (a whole number value from -0.4 to 40) Not engaged (-0.5)	Engaged (a whole number value from -3 mm to 16 mm) Not engaged (-4 mm)

Functionality available on module	● 2 RGB LEDs
	● 2 LED states:
	  ○ **Moving**
	    The top or bottom LED is on depending on whether the magnetic bed is moving up or down respectively.
	  ○ **Stationary**
	    Top LED is on when magnetic bed is positioned at >= 0.5 mm
	● Bottom LED is on when magnetic bed is positioned at < 0.5 mm.
	● The bottom LED will be **on** (and top LED off) when disengaged. But when engaged, whether the top LED will be on or the bottom one will depend on the engage height.	

Functionality available in-app	● set engage height ● disengage (if engaged)	
Python Functionality	• `MagneticModuleContext.engage()` • `MagneticModuleContext.disengage()` • `MagneticModuleContext.status`	
Locking Mechanism	Users must secure labware to the module with the supplied bracket.	

Opentrons Temperature Module

The Temperature Module is a hot and cold plate module. It sits securely on the deck of the OT-2 and holds reagents at temperatures between 4 °C and 95 °C.

GEN2 Temperature module with an aluminum block.

Opentrons Temperature Module Specifications

Dimensions (w/block)	194 x 90 x 126 mm (l/w/h)
Weight	3.3 lbs / 1.5kg
Module Power Input	36VDC, 6.1A
Power Adapter Input	100-240VAC, 50/60Hz, 4.0A/115VAC, 2.0A/230VAC

Opentrons Temperature Module Key Features

Feature	GEN 1	GEN 2
Image		
Unique features per generation	N/A	"Insulation Ring" around the top plate and on the aluminum labware for improved insulation
States	- Idle - Heating - Heated - Cooling - Cooled	
Functionality available on module	1 RGB light bar - Currently turns red when heating, blue when cooling 2-character LCD display - Can display letters and numbers like an old digital clock	
Functionality available in-app	- set temperature - deactivate (if a target temperature has been set)	
Python functionality	`TemperatureModuleContext.set_temperature()` `TemperatureModuleContext.temperature` `TemperatureModuleContext.target` `TemperatureModuleContext.status` `TemperatureModuleContext.deactivate()`	

Opentrons Thermocycler Module

The Thermocycler Module is a PCR instrument. The lid's temperature range can be set between 37 °C and 110 °C, and the block's temperature range can be set between 4 °C and 99 °C. It utilizes a disposable plate seal system. The lid is also remotely controllable for opening/closing.

Thermocycler Module with its lid open.

Thermocycler Module Specifications

Dimensions (open)	316 x 172 x 312 mm (l/w/h)
Dimensions (closed)	316 x 172 x 154 mm (l/w/h)
Uniformity	± 0.5 °C up to 95 °C
Accuracy	± 0.5 °C up to 95 °C
Settling time	within 30 sec of arrival at target temperature
Ramp Rate (block)	4 °C/s (max); 2.5 °C/s (average)
Cooling Rate	2.5 °C/s (max)
Weight	17 lbs / 7.6 kg
Module Power Input	36VDC, 17.5 A
Power Adapter Input	85-264VAC, 47-63Hz, 4.0A/115VAC, 2.0A/230VAC

Thermocycler Naming Conventions

Application	Naming
API Load Name	`thermocycler module` or `thermocycler`
Opentrons Client Application	Thermocycler

Thermocycler Key Features

Feature	GEN1
Image	
States	- Lid - Open - Closed - Lid temperature - Idle - Heating - Heated - Cooling - Cooled - Block temperature - Idle - Heating

	○ Heated ○ Cooling ○ Cooled ● Error
Functionality available on module	● 16 WRGB LED strip ○ Currently turns red when heating, blue when cooling, white LED ring around button
Functionality available in-app	● open/close lid ● set lid temperature ● set block temperature
Python functionality	● `ThermocyclerContext.open_lid()` ● `ThermocyclerContext.close_lid()` ● `ThermocyclerContext.set_lid_temperature()` ● `ThermocyclerContext.set_block_temperature()` ● `ThermocyclerContext.status` ● `ThermocyclerContext.lid_position` ● `ThermocyclerContext.lid_position` ● `ThermocyclerContext.deactivate()` ● `ThermocyclerContext.deactivate_lid()` ● `ThermocyclerContext.deactivate_block()` ● `ThermocyclerContext.execute_profile()` ● `ThermocyclerContext.load_labware()`
Locking mechanism	Users must lock their labware to the Thermocycler Module by securing the Thermocycler Latch Mechanism. This latch secures the plate onto the module to ensure level and accurate plate placement for optimal results.
Auto open/close lid during a Labware Position Check	The OT-2 has limited clearance to pass above the Thermocycler Module when its lid is closed. Because of this, the OT-2 will automatically move in a way that avoids the deck slots it occupies. If a Thermocycler Module is present during a Labware Position Check and the lid is not open, the robot will automatically open the lid to check it.

4.9 Operate Modules of the Same Type on the OT-2 Robot

Modules of the same type can be used in a protocol run on the OT-2 with software version 4.3. This means a user can use multiple Temperature Modules or multiple Magnetic Modules that can be controlled separately in the same Python API protocol.

Note: While it is entirely possible to use more than two modules of the same type at once, Opentrons encourages you to be cognizant of:
- Available deck space on the OT-2
- The locations of modules on the deck relative to USB ports on the OT-2

Prerequisites for Protocols with Multiples of the Same Module

To use this feature, all of the following need to be true:
1. You must be connected to the OT-2 via the Opentrons App and be running software version 4.3 or higher.
2. You must be using a Python API protocol. You can't currently use multiple modules of the same type with Protocol Designer protocols.

Note: You can't currently use multiple Thermocycler Modules in the same protocol.

How to connect multiple modules of the same type to your OT-2

The OT-2 maps out each module sequentially: in order of appearance to the corresponding physical robot USB ports, from lowest port number to highest port number. Typically, USB ports will be numerically ordered from left to right.

This is important to note because the order in which the modules are plugged into the OT-2's USB ports will impact how a module is called via the load_module() method in your Python API protocol.

- For example, this means Temperature Module 1 plugged into the leftmost USB port will correlate to the first temperature module that is called in your protocol via the load_module() method.
- Using the same logic, the second temperature module in your protocol will correlate to the USB port to the right of Temperature Module 1 and **should be called in your protocol via the load_module() method after Temperature Module**

Please see below for a diagram example that shows the basic mapping of two Temperature Modules on the OT-2:
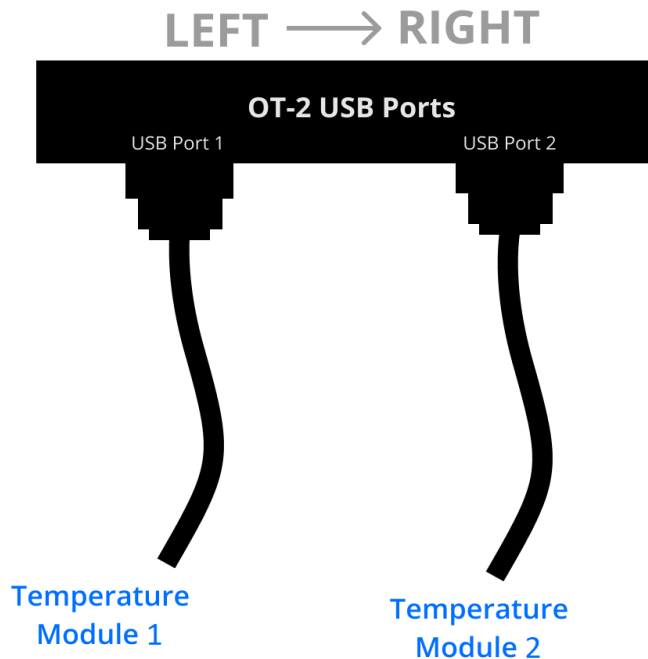
Using a Hub with Multiple Modules

If you need to increase the number of USB ports available on the OT-2 and are using a hub with this feature, ensure your multiple modules are:

1. Called in the correct order in your Python API protocol via the load_module() method
2. Correctly mapped to the physical robot's USB ports and/or hubs

Additionally, please note that a hub is **not** necessary to use multiple modules unless you require additional USB ports.

For example, in a protocol, the diagram shown below would map to your modules as follows:

● Referencing the diagram below with a hub, you should make sure that **Temperature Module 1** in your protocol is plugged into **USB Port 1 on the OT-2** and **Temperature Module 2** in your protocol is plugged into **Hub Port 1, which is then plugged into USB Port 2 on the OT-2**.
    ○ **Important note**: If for any reason you want to plug Temperature Module 2 into a USB port to the left of Temperature Module 1, you should ensure that the physical modules are placed in the corresponding slot as per your protocol.

LEFT ⟶ RIGHT

OT-2 USB Ports

USB Port 1   USB Port 2

LEFT ⟶ RIGHT

Hub

Hub Port 1   Hub Port 2

Temperature
Module 1

Temperature
Module 2
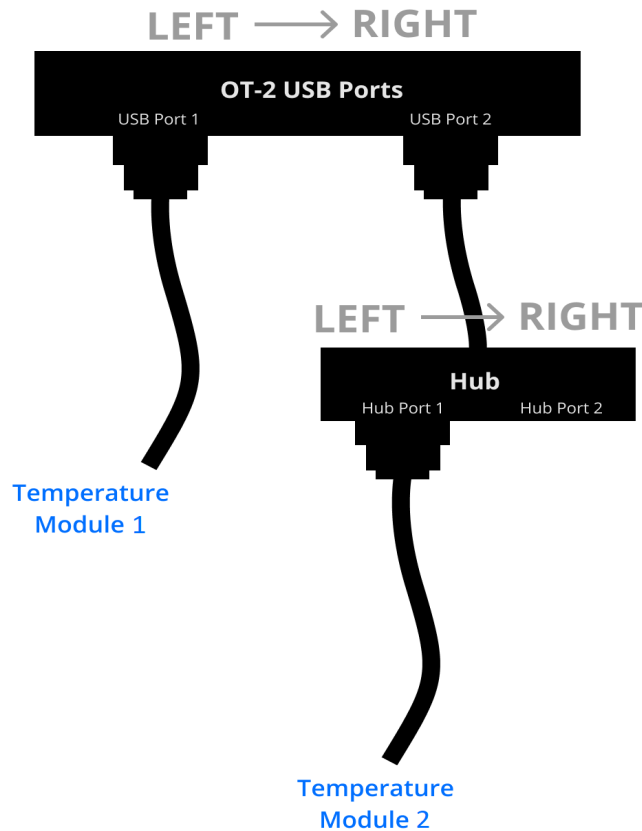
Using multiple modules of the same type in Python API protocols

To use multiple modules of the same type in your Python API protocol, load a module as you would normally.

Please see our API Version 2 Documentation on *Using Multiple Modules of the Same Type* for an example of how to incorporate this feature into your protocol.

Limitations of the multiple modules feature

Protocol Designer protocols do not support multiple modules of the same type.

Identifying USB port connections

Identify Through the Opentrons App
The Opentrons App will display USB port information while using modules in your protocol. This means you can see which module is plugged into which USB port.

Please keep in mind that the Opentrons App will **not indicate if anything is incorrect** in your configuration as it relates to this feature. Due to this, it is critical that you ensure you are working

with the correct configuration based on the order of modules being called in your Python API protocol.

However, the Opentrons App will provide you with on-screen guidance for how to use the feature. As shown in the images below, the Opentrons App will visually indicate the following as it relates to using multiple modules:

1. Deck layout (USB port order of modules and which slot each module is in)
2. Status of module (connected or not connected)

The image below shows an example of what the Module Setup section will look like when using multiple modules of the same type.
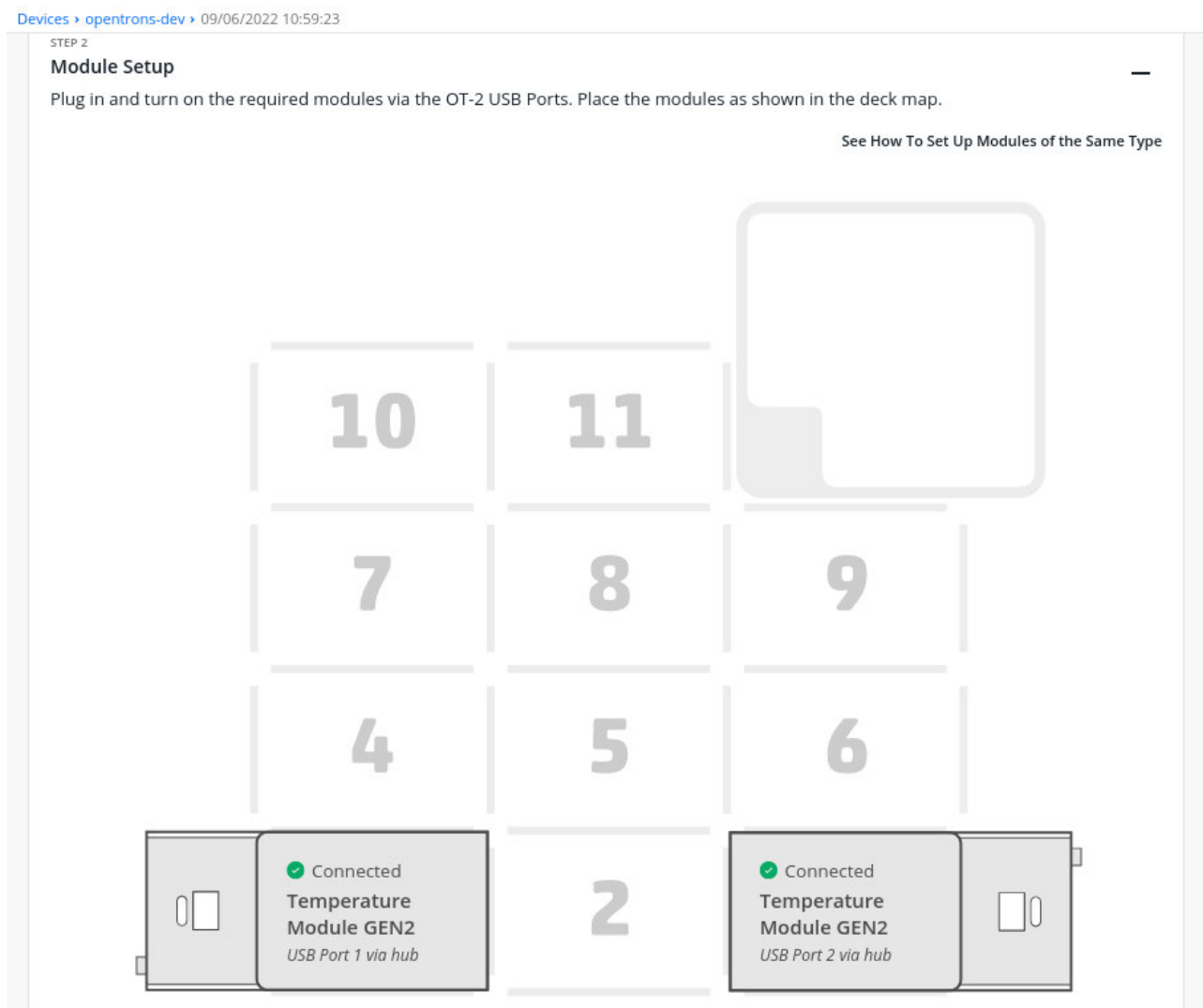
Identify Through Jupyter Notebook
Using multiple modules of the same type through Jupyter Notebook will **not** display USB port information, **however**, you can use the protocol load order as a map to the USB ports to utilize multiple modules in a protocol.

4.10 Power Consumption Considerations of the OT-2 and Hardware Modules

The total power consumption of the Opentrons modules depends on the specifics of the protocols run on the robot.

Estimated power consumptions

Temperature Module (GEN1 and GEN2)
The Temperature Module can consume close to its maximum power, approximately 180 W, when heating or cooling close to its maximum temperature limit. Protocols that involve temperatures closer to ambient will consume less power.

Magnetic Module (GEN1 and GEN2)
The Magnetic Module will consume a small amount of power while idle: approximately 10 W or less. Consumption may briefly increase to approximately 60 W while the motor is actively moving inside the device to engage or disengage the magnets.

Thermocycler Module (GEN1)
The Thermocycler will draw approximately 500 W when ramping through cycles with large temperature changes.

HEPA Module (GEN1)
The HEPA Module draws 145 Watts while the fan is blowing at full power. However, the unit will typically pull slightly less than 145 Watts, especially when running on lower intensity.

Uninterruptible power supply recommendations
In case of a power failure, an uninterrupted power supply (UPS) with battery storage can allow the OT-2 to continue running for a calculated amount of time. In order to calculate the time of continued operation, one can take the battery life of the UPS in watt-hours (Wh) and divide that by the number of watts (W) utilized by the machine. For example, the maximum power consumption of the OT-2 is 220W (voltage * current). If the UPS has runtime of 1 hour at 110 W, then the OT-2 can run for at least 30 minutes off the UPS. The machine can likely continue running far longer than the maximum calculated time as the power averages at a lower power value.

4.11 What Labware can you use with Opentrons Modules?

Labware Compatible with the Magnetic Module
Opentrons has tested with, and recommends, the following labware:

- ANSI/SBS format v-bottom conical 96-well plate
- *NEST 96 Well Plate 100 µL PCR Full Skirt*
- *NEST 96 Deep Well Plate 2mL*

How to determine if labware is compatible with the Magnetic Module

Other labware can be used with the Magnetic Module, but users should perform testing to check labware for fit and performance. It will be necessary to adjust the magnet engage height.

For labware to fit on the Magnetic Module, it should be a standard-sized 96-well plate with V-bottom (conical) wells. The standard plate bracket supports plates up to 22 mm tall. The deep-well bracket can support deep well trays up to 47 mm tall.

 Examples of compatible labware include:
- *Bio-Rad 96 Well Plate 200 µL PCR*
- *USA Scientific 96 Deep Well Plate 2.4 mL*

See the *Magnetic Module White Paper* for details and dimensions.

Labware Compatible with the Temperature Module

We strongly recommend placing labware on an *Opentrons aluminum block*. This provides the quickest time to temperature and ensures even heating and cooling across all wells.

Labware on an aluminum block

To check if your labware will fit on the aluminum block, consult the dimensions in the *Temperature Module White Paper*.

To use this setup in a protocol, you'll need a labware definition that encompasses the *combination* of the labware atop the aluminum block. The *Opentrons Labware Library* provides several such definitions for common labware. If the labware does not contain a definition for the labware , use the *Custom Labware Creator* to make a custom labware definition.

Labware without an aluminum block

Any ANSI/SBS compliant well plate or reservoir will fit on the Temperature Module deck.
NOTE: Labware will not heat or cool correctly without a proper fitting aluminum thermal adapter. Plates will not heat or cool correctly if sitting directly on top of the temperature module surface without a thermal adapter.

Labware Compatible with the Thermocycler Module

Recommended labware

Opentrons has tested with, and recommends, the following labware:
- *NEST 96 Well Plate 100 µL PCR Full Skirt*

You should use this labware to make sure your results match those demonstrated in the *Thermocycler Module White Paper*.

Other labware

The Thermocycler Module is compatible with standard-sized, 96-well, full-skirted PCR plates. Examples include:
- *Bio-Rad 96 Well Plate 200 µL PCR*

Chapter 5 | Operation

5.1 Turn On OT-2 Power

Connect the power supply to the wall. It has an LED that indicates it is receiving power correctly. When connecting the power supply to the robot, make sure it is in the correct orientation. Locate the power supply connector on the rear left-hand side and power on using the power switch.
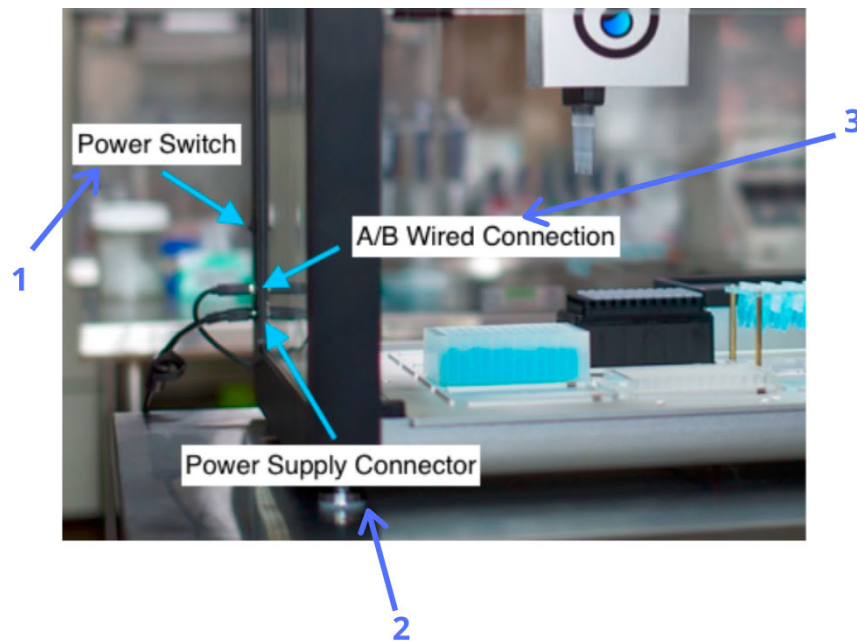
It takes several minutes for the robot to completely power on before you can connect to it. When the robot is powered on successfully the following will occur:
1. The blue light on the front of the robot will blink.
2. The blue light will then turn solid.
3. The robot will emit audible mechanical noises from the robot as the pipette mounts home.

If the blue light on the front of the robot does not turn solid after the robot has been powered on for more than five minutes, contact Opentrons Support for help, support@opentrons.com.

5.2 Turn Off OT-2 Power

To power down the OT-2 after the completion of a protocol, turn the power switch to the off position.

5.3 Launch Opentrons Apps

The OT-2 liquid handling robot is controlled by a computer running the Opentrons App. The Opentrons App runs on Windows 10, macOS, and Linux. Please see supported operating systems for more details about OS support.

5.4 Chemical Compatibility

Opentrons pipettes are designed and tested to handle a wide range of aqueous solutions common to biochemistry.

Material composition of Opentrons tips
Opentrons tips are made from 100% polypropylene.

Material composition of Opentrons pipettes
The parts of the Opentrons pipettes that interact with liquid or vapor during the run of a protocol are composed of:
- PC-ABS (external & internal housing)
- NBR (internal plunger O-ring)
- Silicone (external O-ring)

Only the pipette tips are designed to touch the liquid. If liquid enters the inside of the pipette, it can permanently damage it.

Acetone interaction with Opentrons pipettes
Never use acetone with Opentrons pipette tips. Acetone will permanently damage the Opentrons pipettes. The pipettes are composed of PC-ABS plastic, which is extremely sensitive to acetone.

Pipetting methanol or ethanol with Opentrons pipettes
Opentrons does not advise pipetting 100% ethanol or methanol because the vapor can damage the internal components of the pipettes over time.

If using methanol or ethanol, Opentrons recommends the preparation of a stock solution by hand at a lower concentration before using the solution in a protocol on the robot. Whenever possible, using diluted alcohol solutions will not only increase the life expectancy of the pipettes, but also improve volumetric accuracy, due to reduced vapor pressure.

If it is absolutely necessary to use 100% alcohol on the robot, we recommend you check the volumetric accuracy of that pipette more frequently, as the alcohol vapor will cause more wear than pipettes used only with aqueous solutions, and this can cause noticeably reduced volumetric accuracy over time.

Using volatile solvents with Opentrons pipettes

Opentrons does not advise the use of volatile solutions with the OT-2. The pipettes are constructed from PC-ABS, and PC-ABS has very limited chemical resistance, particularly to aliphatic, aromatic, and chlorinated hydrocarbons as well as alcohols, ketones, esters, and glycols.

Since the internal plunger of the pipette is made from PC-ABS, even though it does not contact the liquid, it is susceptible to damage over time from solvents with a high vapor pressure. Because the tips are polypropylene, it may be possible to handle some volatile solvents initially, but the vapor created during aspiration will damage the internal PC-ABS parts, degrading the performance of the pipette over time and eventually breaking it. Depending on the solvent, this might happen after the first use, or it might only happen after repeated use.

The external surfaces of the pipette are also made from PC-ABS and will be damaged if they accidentally touch your solvent.

Below is an OT-2 Pipette Chemical Compatibility Chart. Items outside of the information in this list have not been tested to support use on the OT-2.

	PC-ABS (external & internal housing)	NBR (internal O-ring)	Silicone (external O-ring)	Polypropylene (tips)
Acetone	X	X	X	✓
Ethanol	✓	X	✓	✓
Methanol	X	✓	✓	✓
Ethyl Acetate	X	X	✓	✓
MTBE	X	X	X	X
Petroleum Ether	✓	X	X	X
Hexanes	X	✓	X	✓
Methylene Chloride (DCM)	X	X	X	X
1,2 Dichloroethane (EDC/ethylene dichloride)	X	X	X	X
Acetonitrile	X	X	X	✓
Methyltetrahydrofuran (MeTHF) – Viridisol M	X	X	X	X
Dioxane	✓	X	X	✓
Tetrahydrofuran (THF)	X	X	X	X
Dimethyl formamide (DMF)	X	X	X	✓
Dimethyl sulfoxide (DMSO)	✓	X	✓	✓
Dimethylacetamide (DMA)	✓	X	✓	X
Chloroform	X	X	X	X
Carbolic Acid (Phenol)	X	X	X	✓

Legend:
  ✓ *Recommended for use on the OT-2*   *X   Not recommended for use on the OT-2*

5.5 UV Compatibility

Opentrons does not support the use of UV light inside the OT-2. All such applications should be performed at the customer's own risk and discretion.

The instrument is built with a polycarbonate enclosure, which can degrade under prolonged exposure to UV light.

In the event that a UV light is used inside the OT-2, appropriate PPE should be worn by the user as the OT-2 is not a closed system and will in no way protect the user from UV exposure.

5.6 Door Safety Switch

The OT-2 includes an auto-stop feature to reduce the risk of injury when either the top window or front door are open. The auto-stop feature is **disabled** by default.

Activating the Door Safety Switch

To activate the door safety switch, go to **Robot Settings > Advanced > Usage Settings** and turn on "Pause protocol when robot door opens":
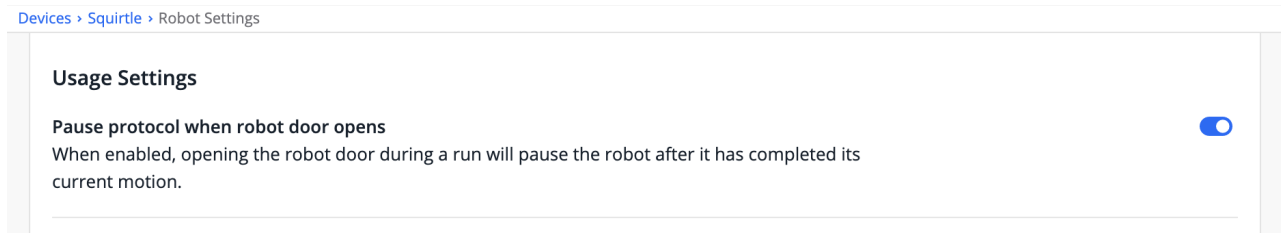
Using the Door Safety Switch

The OT-2 has two switches to detect when either the front door or top window are open:

Front door switch.
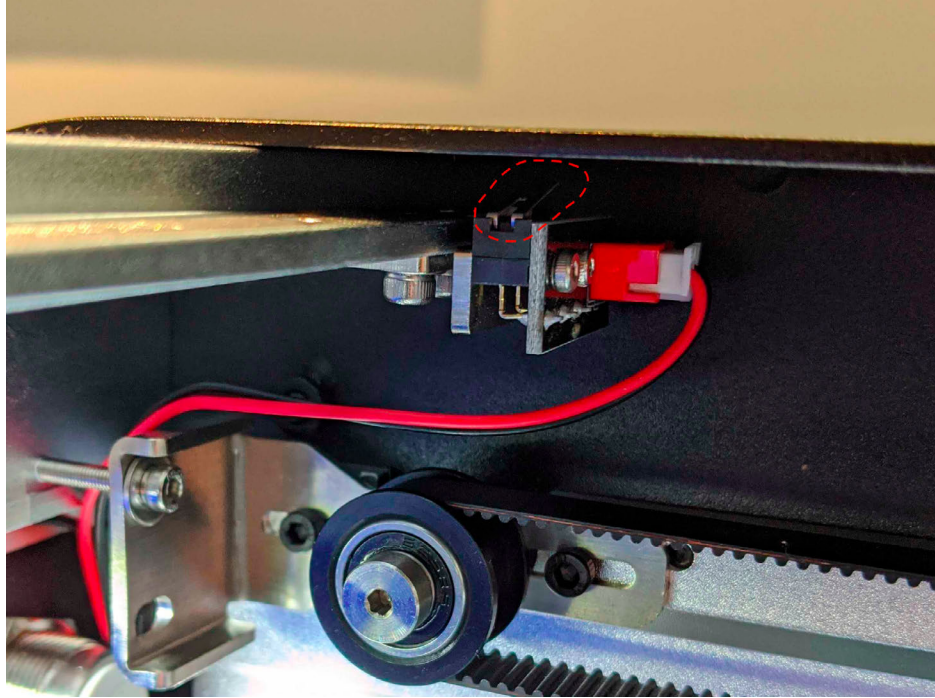
Top window switch.

If either of these are open:

- OT-2 will stop after completing its current movement
- OT-2 will prevent you from starting a protocol run
- Robot calibration and other processes requiring manual robot interaction are still enabled

The Opentrons App will inform the user if the door is open, and prompt the user to close the door or lid before starting or resuming a protocol:

Devices › Squirtle › 08/31/2022 16:06:03

Serial Dilution Tutorial

⚠ Close robot door to resume run

RUN	STATUS	RUN TIME	
08/31/2022 16:06:03	Paused - door open	00:00:42	▶ Resume run

PROTOCOL START	PROTOCOL END	
08/31/2022 16:07:02	--:--:--	Cancel run

Setup | **Run Log**

Run Log 994 steps total   Download run log

PROTOCOL SETUP   ⌄

1	Start: 00:00:20 End: --:--:--	Picking up tip from A1 of Opentrons 96 Tip Rack 300 µL in 1
2	Start: --:--:-- End: --:--:--	Aspirating 100.0 uL from A1 of NEST 12 Well Reservoir 15 mL on 2 at 92.86 uL/sec
3	Start: --:--:-- End: --:--:--	Dispensing 100.0 uL into A1 of NEST 96 Well Plate 200 µL Flat on 3 at 92.86 uL/sec
4	Start: --:--:-- End: --:--:--	Aspirating 100.0 uL from A1 of NEST 12 Well Reservoir 15 mL on 2 at 92.86 uL/sec
5	Start: --:--:-- End: --:--:--	Dispensing 100.0 uL into B1 of NEST 96 Well Plate 200 µL Flat on 3 at 92.86 uL/sec
6	Start: --:--:-- End: --:--:--	Aspirating 100.0 uL from A1 of NEST 12 Well Reservoir 15 mL on 2 at 92.86 uL/sec

Protocols
Labware
Devices

Chapter 6 | Software

6.1 Opentrons App

Several screens in the Opentrons App let you view information about a robot, change its settings, and control it.

Robot Settings

1. Robot name
2. Robot Server Version: This is the version number of the software that runs on the OT-2 itself, and is separate from the version of the Opentrons App. The two must match in order to run protocols. If the software versions don't match even after trying a manual update, please contact Opentrons Support at support@opentrons.com.
3. Firmware Version: The version of the motor controller.
4. Supported Protocol API Versions: The range of Python Protocol API versions the robot will accept. This does not affect JSON protocols created with Protocol Designer.

Devices › Squirtle › Robot Settings

Robot Settings

Calibration   Networking   **Advanced**

About

Robot Name   **Rename robot**
Squirtle

Robot Server Versions
v6.0.1   up to date   **Reinstall**
View latest release notes on **GitHub**

Robot Serial Number   **Firmware Version**   **Supported Protocol API Versions**
OT2CEP20210812R11   v1.1.0-25e5cea   v2.0 - v2.13
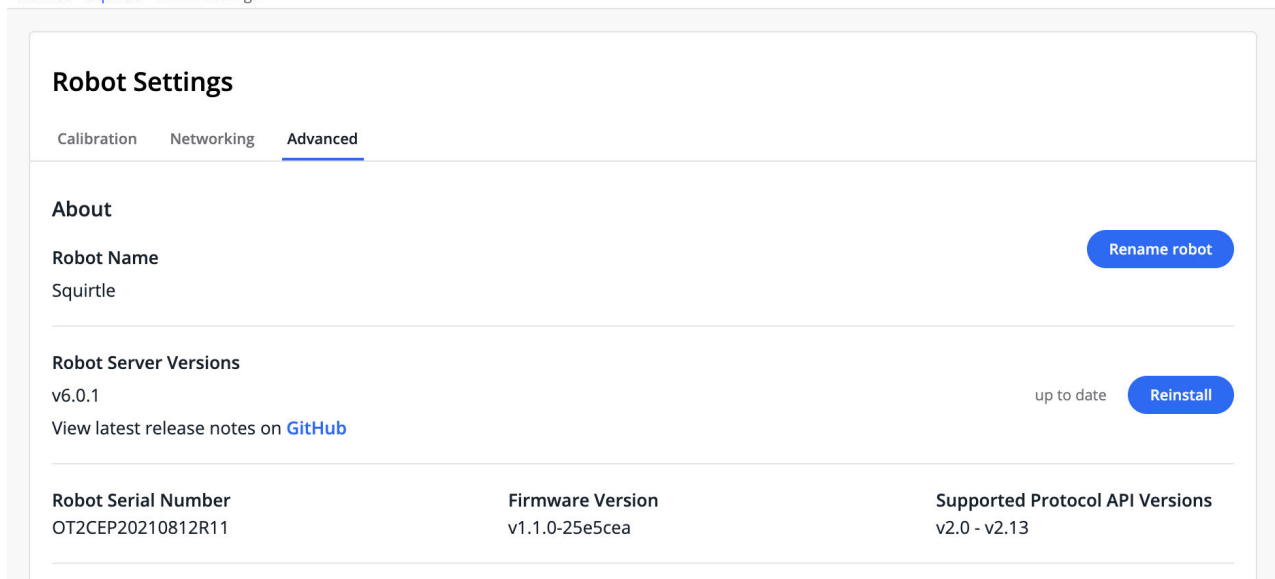
Device Details

Clicking on an available robot on the **Devices** screen will show the following:
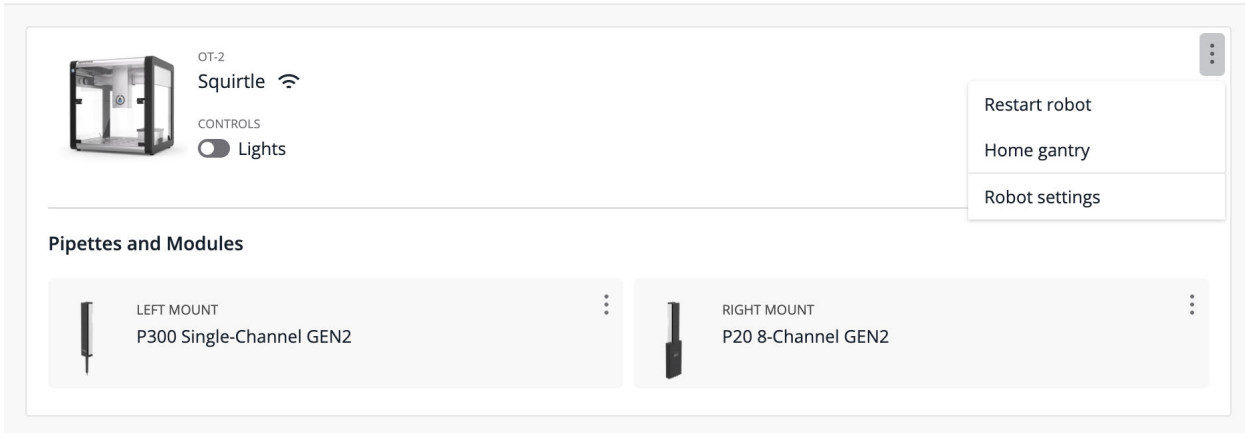
Robot controls

These controls are available from the three-dot menu on the device details screen:

1. **Restart robot:** Turn the OT-2 off and on again.
2. **Home gantry:** Move the gantry to the back right corner of the OT-2.

The **Lights** toggle under the robot name will turn the deck illumination on or off.

App Settings

Click the gear icon at the bottom left of the app window to access these settings:
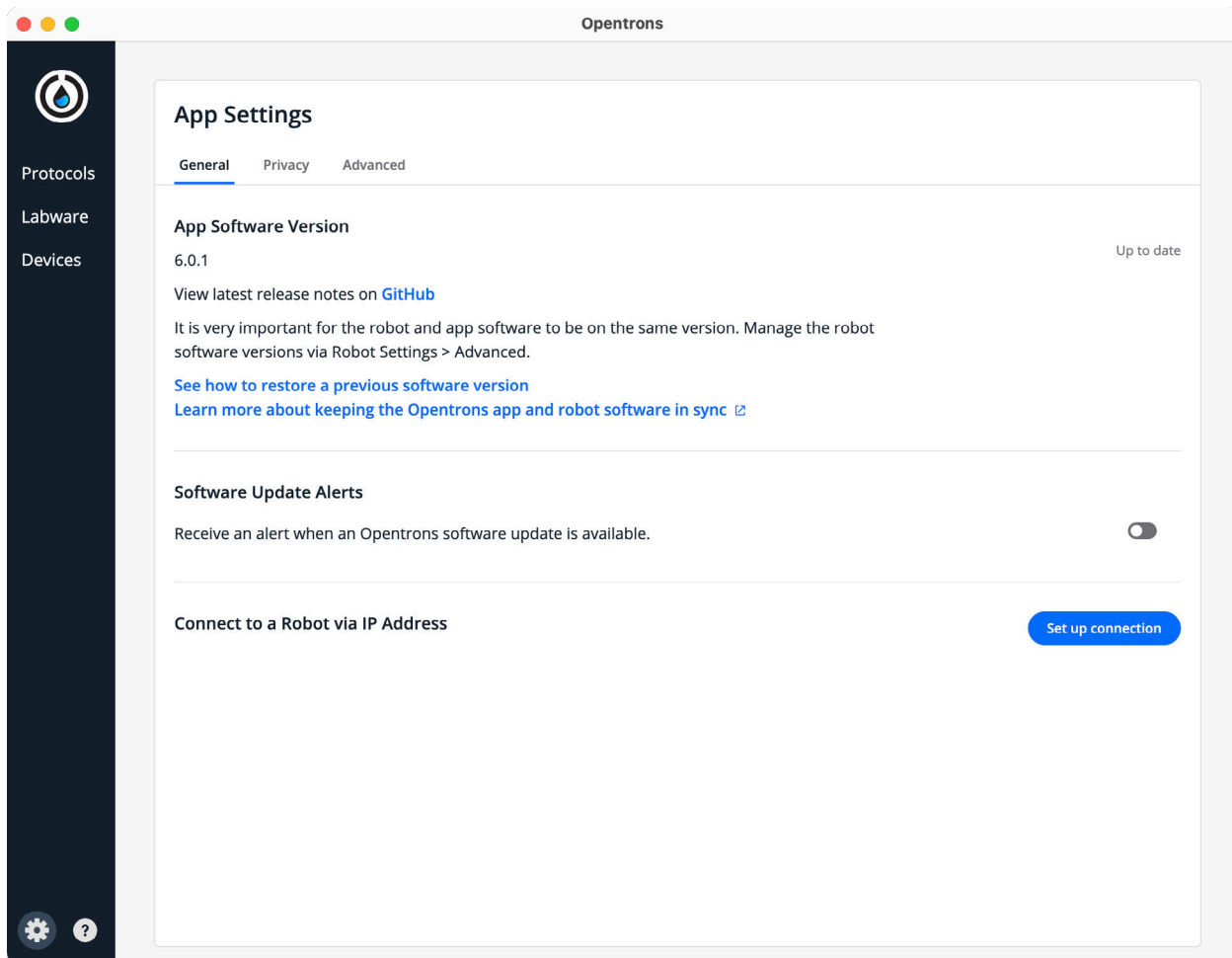
- **App Software Version:** This is what version of the app you are currently running on your computer. This is separate from the version of the robot server that runs on the OT-2, although the two must match to run protocols.
- **Software Update Alerts:** Whether the app will notify you if a newer version is available. You can turn this off, but we strongly recommend keeping your app up to date.
- **Connect to a Robot via IP Address:** If your network environment prevents automatic discovery of your OT-2, you can manually specify its IP address or hostname here.
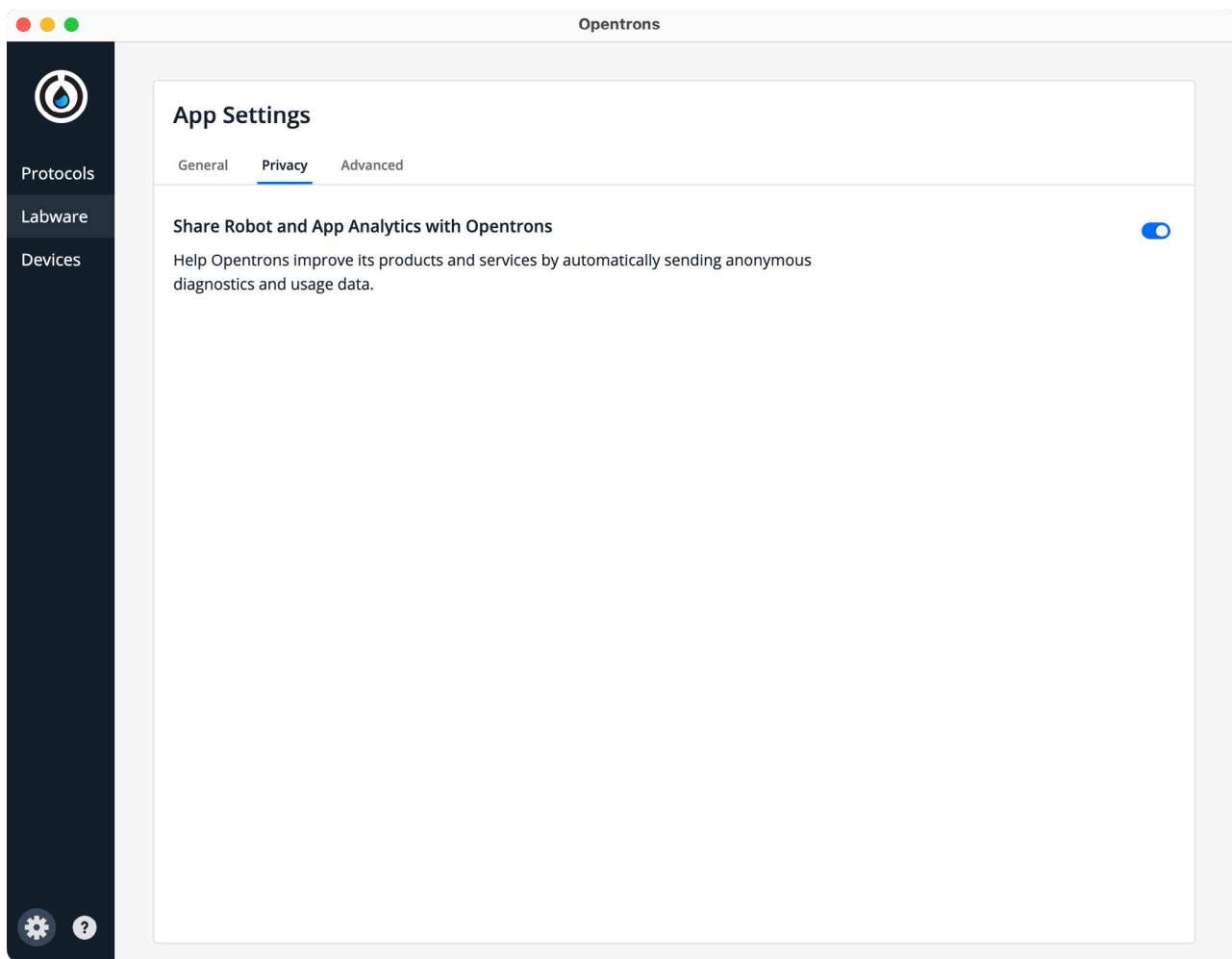
- **Share Robot and App Analytics with Opentrons:** When enabled, the app will send anonymized data to Opentrons about your use of the app and connected robots. We never sell this data, and you can always opt out.
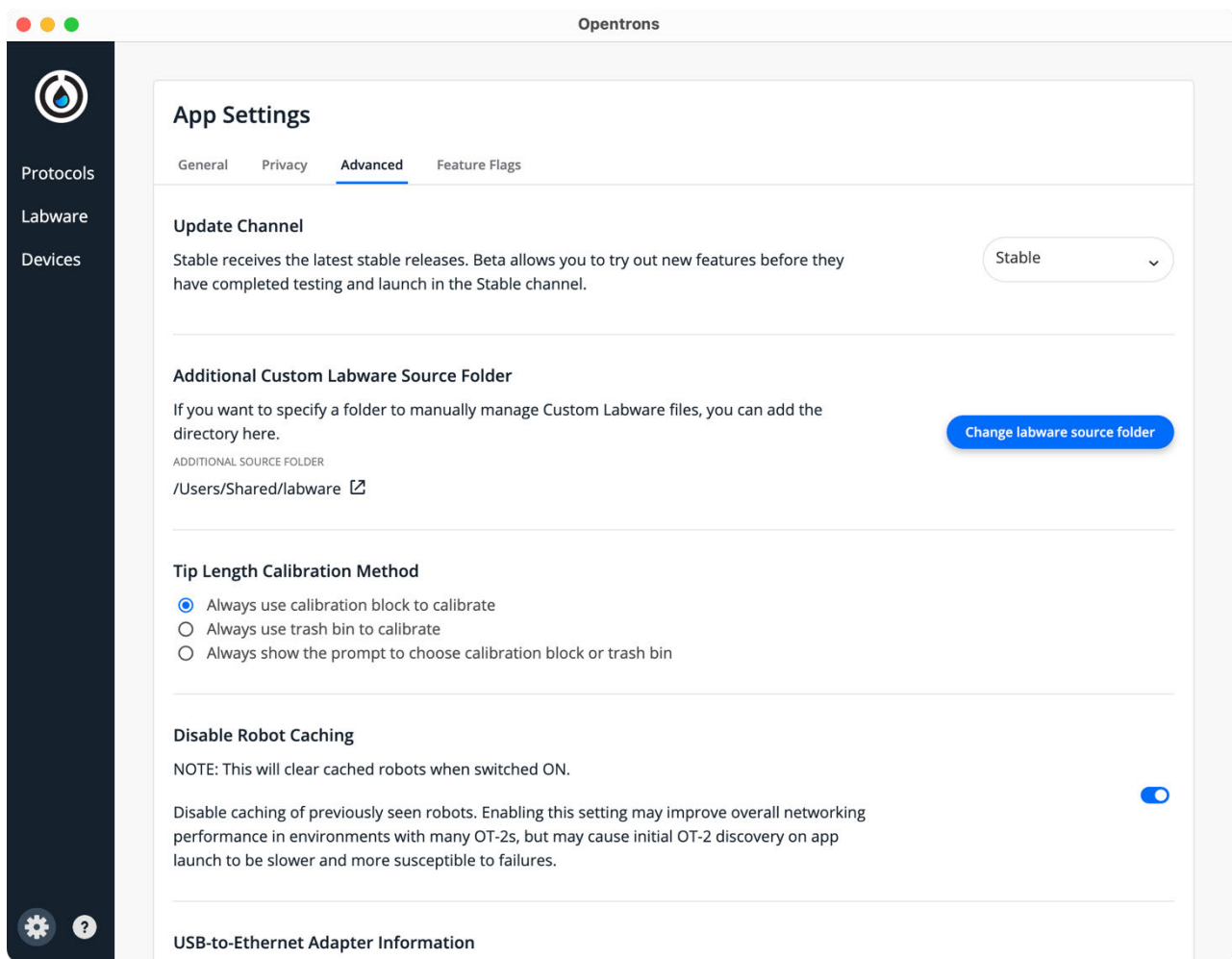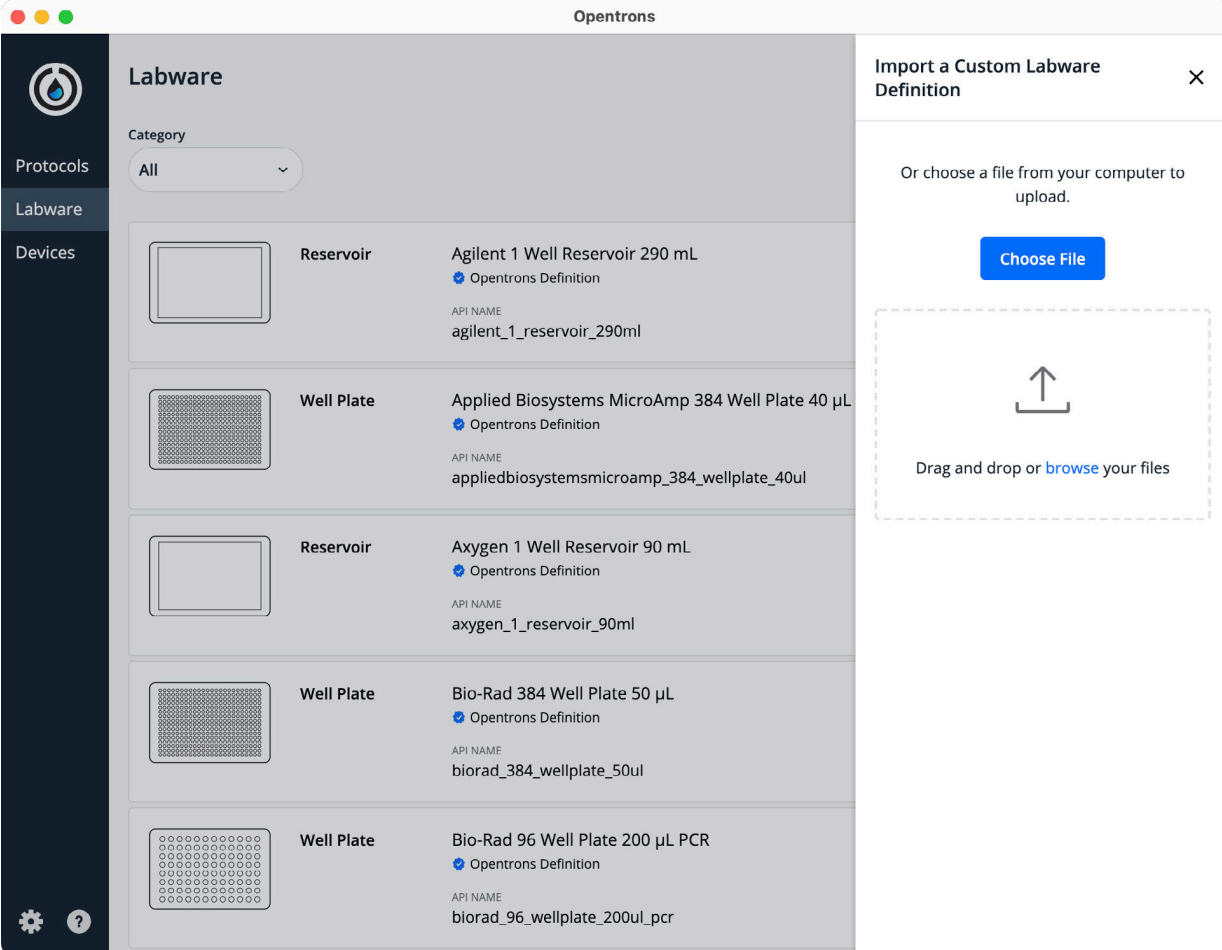
- **Update Channel:** Choose what versions of the Opentrons App you would like to update to: Stable, Beta, or Alpha (only available if the Enable Developer Tools setting is enabled). Most users should remain on the Stable channel unless you are participating in an Opentrons beta test program.
- **Additional Custom Labware Source Folder:** If you want to specify a folder to manually manage custom labware files, you can add the directory here.
- **Tip Length Calibration Method:** Users who *don't have a calibration block* can change this to use the trash bin for calibration until the calibration block arrives.
- **Disable Robot Caching:** The app will not remember previously connected robots when this is enabled.
- **USB-to-Ethernet Adapter Information:** If you have a robot connected with this type of adapter, details about it will appear here.
- **Show Link to Get Labware Offset Data:** Enable this setting to gain access to exporting Labware Offset data for use with Jupyter Notebook or other software.
- **Confirm Heater-Shaker Module Attachment:** By default, the Heater-Shaker Module will always remind you to confirm its attachment to the deck.
- **Clear Unavailable Robots:** One-time action to clear the list of unavailable robots.
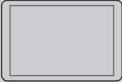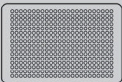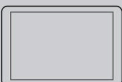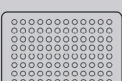
- **Override Path to Python:** The Opentrons App ships with its own version of Python for performing protocol analysis. If you need to use a different Python installation elsewhere on your computer for analysis (for example, if your protocols import Python modules not shipped with the app), you can specify it here.
- **Enable Developer Tools:** Display standard web developer tools every time the app launches. Turning this also enables additional settings only meant for those who are developing the Opentrons App. (The app is open-source and we welcome contributions from all!)

Custom Labware

To add custom labware, go to the **Labware** screen and click **Import** at the top right.

Custom Labware is stored on the computer you are using. If you move computers, you will need to load the custom labware on that computer as well.

6.2 Protocol Development

A user can run a target protocol on the OT-2 by using any of the following four protocol development pathways:

- The Protocol Designer Onboarding Protocol Tutorial
- The Python Protocol API Tutorial
- Download a protocol from the Protocol Library
- Purchase a Custom Protocol

Protocol Library

All OT-2 compatible workflows including our Featured and Verified protocols can be found in the Opentrons Protocol Library. All protocols in the Protocol Library are written using Version 2 of the Opentrons OT-2 Python API.

New to the Protocol Library? A great place to start is to determine whether your lab's workflow can be handled by the Featured or Verified protocols.

Featured Protocols in Protocol Library

Most lab workflows can be performed with protocols found in the Featured section of the Protocol Library.

Verified Protocols in Protocol Library

Opentrons Verified protocols are protocols which have been extensively tested on the OT-2 by Opentrons' in-house science team. Look for the "App Note" link in the following protocols to find more information on results and analysis using the OT-2. Opentrons Verified protocols can be found below:

- Swift 2S Turbo DNA Library Kit Protocol: Fully Automated
- Swift 2S Turbo DNA Library Kit Protocol: Semi-Automated
- NGS Cleanup and Size Selection with Omega Bio-tek Mag-Bind® TotalPure NGS
- OT-2 PCR Prep

6.3 API Documentation

The OT-2 Python Protocol API is a Python framework designed to make it easy to write automated biology lab protocols that use the OT-2 robot and optional hardware modules. We've designed the API to be accessible to anyone with basic Python and wet-lab skills.

6.4 Custom Protocol Development

Opentrons Standard Custom Protocol Development service provides customers with 1 brand new protocol and protocol optimization within 2 weeks to support their workflow needs.
Our Custom Protocol Development is a remote service that includes actual protocol development, personalized instructions for deck and reagent setup, code validation by an Applications Engineer before delivery, and protocol optimization within 1 week of initial script delivery (within 4 business hours).

The number of protocols does not generally impact turnaround time.

Opentrons Expedited Custom Protocol Development service provides customers with 1 brand new protocol and protocol optimization within 1 week to support their workflow needs. We offer both standard and expedited workflow development to offer flexibility to your organization's needs.

Custom Protocol Development Offerings

We offer a standard and expedited custom protocol development for 1 to 3 workflows to cater to the turnaround time that you need. Protocol validation and optimization occurs within the turnaround time listed below.

Product List Price	1 Workflow	3 Workflows	Turnaround Time
Standard Workflow Development	$1,000	$2,750	2 weeks
Expedited Workflow Development	$1,300	$3,350	1 week

Our Custom Protocol Development is available to all direct or non-direct customers purchasing Opentrons hardware including purchases made via a distributor or a secondhand purchase of an Opentrons system.

Requesting a Protocol
You can submit a custom protocol request to our Applications Engineering team by following the Custom protocol submission guidelines.

6.5 Protocol Designer

The Protocol Designer is meant to assist in building simple pipetting protocols for the OT-2 without code. Using the Protocol Designer, users can:
● Design protocols using standardized labware for the OT-2
● Assign liquids to source plates and track how they move
● Add and edit simple Transfer, Mix and Pause steps

- Track tips and usage
- Export protocols to run on the OT-2

What operating system is necessary to use Protocol Designer?

At this time Protocol Designer must be accessed using Chrome to function correctly. It also requires an active internet connection to work.

What labware is necessary to use Protocol Designer?

To see which labware is available in the protocol designer, check out our Labware Library. If the user's labware does not match any definitions in the library and the target protocol requires a custom labware definition create one using our Labware Creator.

Get started: Protocol Designer Onboarding protocol tutorial

Try this simple plate duplication protocol to get started with your OT-2.
Moving forward, you can use Protocol Designer to design your own custom protocols to run on the OT-2.
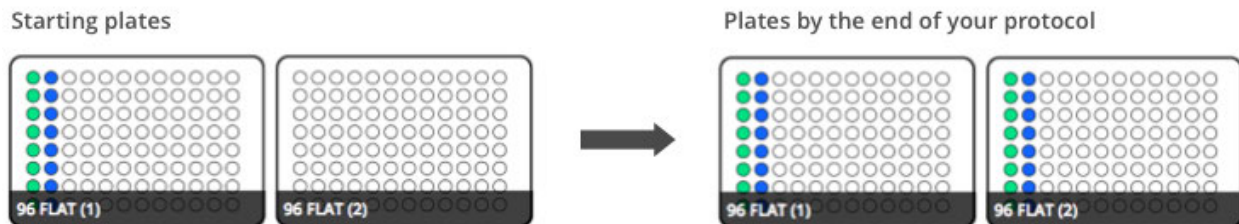
Step 1: Create a protocol
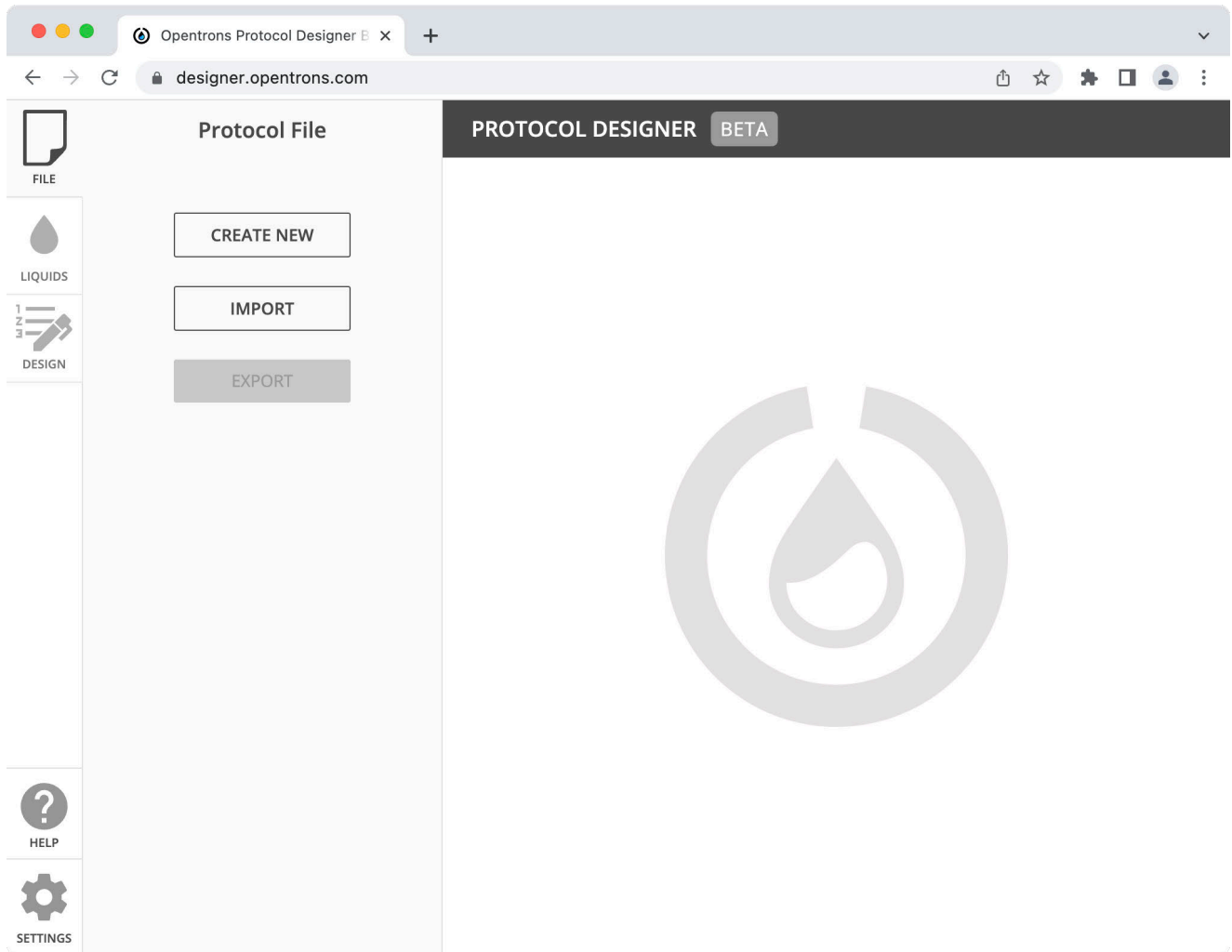
Select your pipettes and assign a tip rack type to each pipette.

Create New Protocol

Name:

Simple plate duplication

Pipette & Tip Setup

Left Pipette:

P20 Single-Channel GEN2 ▼

Right Pipette:

None ▼

Left Tiprack*:

Opentrons 96 Tip Rack 20 µL ▼

Right Tiprack*:

UPLOAD CUSTOM TIP RACK

Add Modules

☐ Magnetic

☐ Temperature

☐ Thermocycler

CANCEL   SAVE

Step 2: Define liquids
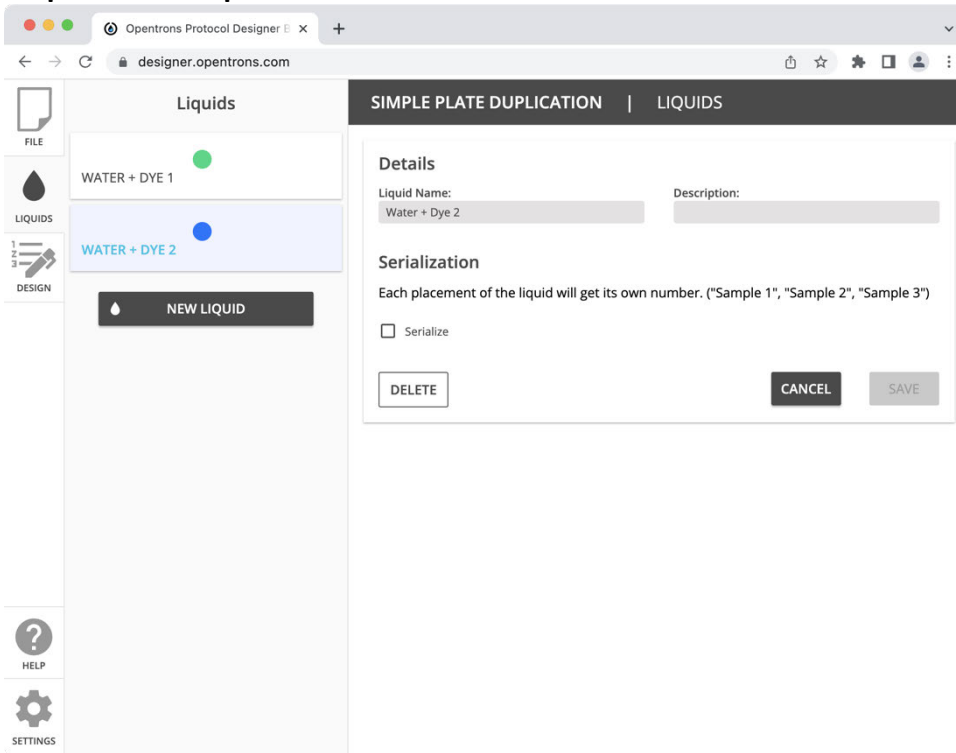
Head to the liquids tab. Add two liquids you want to use for a test run on your robot.

Step 3: Add labware
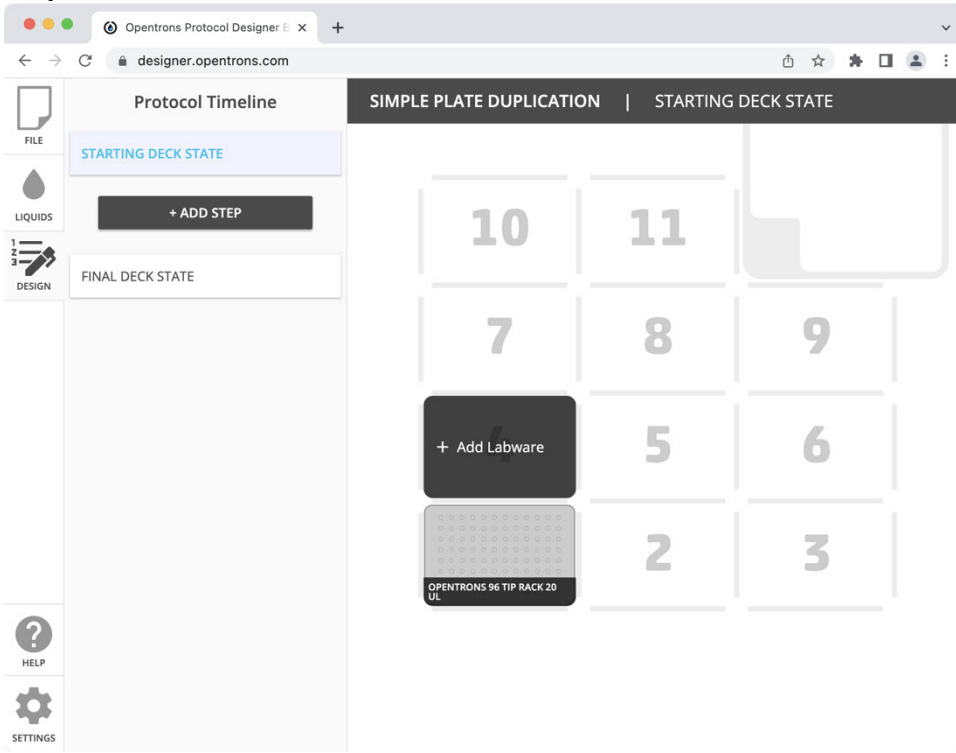
Go to the design tab. When you first go to the design tab, you will see the starting deck state. Hover over empty slots to add labware. Add one source labware and one destination labware. Leave both empty. Choose whatever labware you have on hand that is both in the Protocol Designer labware library and compatible with your pipettes.

This example will use 96 well plates.

SIMPLE PLATE DUPLICATION | STARTING DECK STATE

10	11	
7	8	9
NEST 96 WELL PLATE 200 UL FLAT	NEST 96 WELL PLATE 200 UL FLAT (1)	6
OPENTRONS 96 TIP RACK 20 UL	2	3
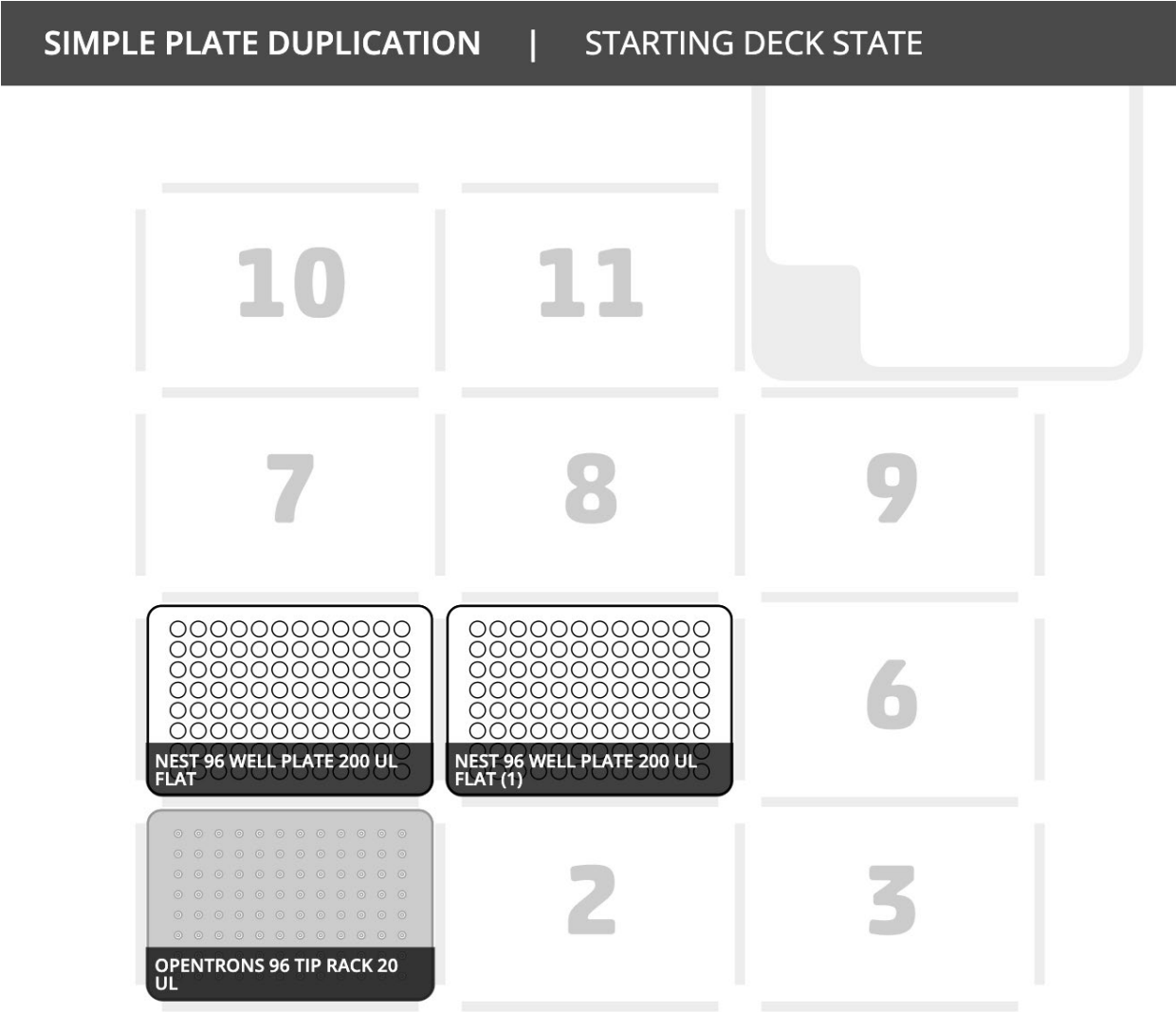
Step 4: Add liquids to their starting places.
Hover over your first labware and click **Name & Liquids**.
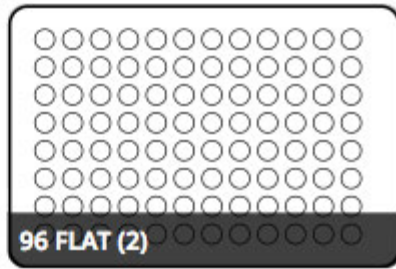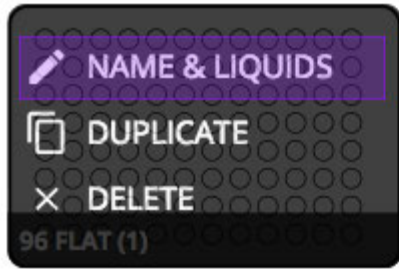
Click and drag to select all the wells in column 1 and fill them with your first liquid.

Select all the wells in column 2 and fill them with your second liquid. Click **Deck** in the navigation bar at the top to return to your deck view.

Step 5: Add a transfer

Protocol Timeline

STARTING DECK STATE

	+ ADD STEP
→	TRANSFER
↓↑	MIX
❚❚	PAUSE
C:	MAGNET
🌡	TEMPERATURE
↻	THERMOCYCLER

FIN
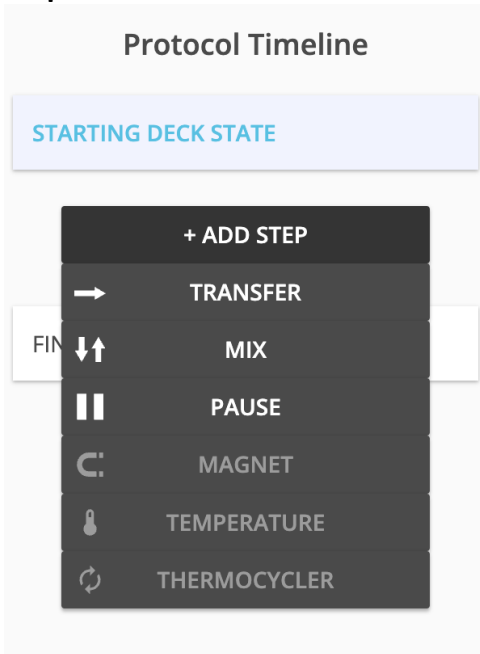
Once your selected labware is on the deck and liquids are in their starting location, it's time to add steps. You can hover on each of the step types to get a description of what they do.

Let's start with a transfer step.

Step 6: Make your step

SIMPLE PLATE DUPLICATION | TRANSFER

TRANSFER

Pipette:
P20 Single-Channel GEN2 ▼

Volume Per Well:
25   μL

ASPIRATE ⚙   **DISPENSE** ⚙

Source:	Wells:	Destination:	Wells:
NEST 96 Well Plate 200 μL Flat ▼	16	NEST 96 Well Plate 200 μL Flat ▼	16

STERILITY & MOTION

Change Tip:
Before every aspirate ▼

Path:
⌒ ⌒⌒ ⌒⌒

DELETE	NOTES		CLOSE	SAVE

Fill out the required fields in your step form. Aspirate from the wells in the first 2 columns of your source labware. Dispense into the wells of the first 2 columns of your destination labware.

Step 7: Customize your step

Flow Rate:	Tip Position:	Well Order:	Flow Rate:	Tip Position:	Well Order:
default µL/s	1 mm	⊠	default µL/s	0.5 mm	⊠

☐ Pre-Wet Tip   ☐ Delay

☐ Mix   ☐ Mix

☐ Delay   ☐ Touch Tip

☐ Touch Tip   ☐ Blowout

☐ Air Gap   ☐ Air Gap

Each step allows you to customize how the pipette aspirates and dispenses. To see aspirate and dispense advanced settings click the gear. Feel free to explore some of the available settings. Hit save at the bottom of your form.

Step 8: Keep track of your steps

When you hover on a step, a blue highlight will appear on the

- Source and destination labware
- Source and destination wells
- Tips used in the step

If you hover on a specific well in the sidebar, you will see

- Source and destination well highlighted
- Tip highlighted
- Contents of the well

When you're done looking, collapse your step by clicking on the arrow next to the title.

Step 9: Check your work

When you're done, click on the Final Deck State step. When you click on labware and hover on wells, you can see the final state of your wells at the end of your protocol.

Step 10: Export your protocol

In the File tab, select Export. Your protocol will be downloaded to your computer as a .json file that you can upload to a robot and run in the Opentrons App.

Chapter 7 | Consumables

The OT-2 is compatible with ANSI/SLAS-compliant labware and automation-friendly consumables. Our software has labware definitions for the following brands in our labware library.

- Agilent
- Axygen
- Bio-Rad
- Corning
- Eppendorf
- Falcon
- GEB
- NEST
- USA Scientific

7.1 Labware Definitions Explained

For the OT-2 robot to interact with any given labware, the robot needs to reference a corresponding JSON definition file which contains accurate measurements of that labware. This list of measurements informs the robot about the type and shape of the labware.

The OT-2 comes pre-loaded with a set of Opentrons and third-party labware definitions. The full list of these standard labware can be found in the Opentrons [Labware Library](). To include these labware in your protocols, browse for them in Protocol Designer or import them by name into your Python API protocols.

You can also load custom labware on your OT-2. Read more about [creating your own custom labware definitions]() or [requesting custom labware definitions from our team]().

Once you have a labware definition, we recommend you calibrate it for use on the OT-2. Calibration does not modify the labware definition, it only adjusts its position in relation to the deck of the robot. While calibration can be used to compensate for small differences between the height of the labware and the labware definition, we generally recommend using an exact labware definition to match your specific labware.

The following is the full list of data in a labware definition JSON file, which describes both its dimensions and properties such as manufacturer and labware category.

- **Count -** The total number of tubes or wells in your labware.
- **Max volume (µL) -** Total maximum volume of each well/tube/tip.
- **Shape -** The shape of the bottom of a well or tube.
- **Length (mm) -** When you look top down at your labware while on the robot's deck there should be a long and short side. Length is the long side.
- **Width (mm) -** When you look top down at your labware while on the robot's deck there should be a long and short side. Width is the short side.

- **Height (mm) -** The distance between the top of a tube/well (include any lip but exclude any cap) to the bottom of the labware.
- **Depth (mm)-** The distance from the top of the well/tube (include any lip but exclude any cap) to the bottom of the inside of the well/tube.
- **Total length (mm)-** The distance from the top of the tip to the bottom of the tip.
- **Diameter (mm) -** The inner diameter of a circular well, tube, or tip.
- **X-size (mm) -** The length of the x side of a rectangular well.
- **Y-size (mm) -** The length of the y side of a rectangular well.
- **X-spacing (mm) -** The spacing between columns, measured from the center of a well/tube/tip.
- **Y-spacing (mm) -** The spacing between rows, measured from the center of a well/tube/tip.
- **X-offset (mm) -** The distance from the center of well/tip/tube A1 to the left edge of the labware.
- **Y-offset (mm) -** The distance from the center of well/tip/tube A1 to the top edge of the labware.

7.2 Calibrating Labware

A labware definition informs the robot about the geometry of a physical piece of labware particularly (a) the well location (b) labware height and (c) labware inner well bottom. This is important as slight variations in labware definitions directly impact liquid handler performance and can cause problems such as crashes and missed wells.

Calibrating Labware for Protocols

Calibrate labware when you're running a protocol for the first time. If the run went well then you *do not* need to calibrate again in order to run this protocol again.

See [Do I have to do pipette and labware calibration every time I upload a protocol?](#) For more information

If someone else in your lab runs a different protocol with the same labware they may do a new calibration. This will overwrite the calibration for the labware that you previously saved. The current calibration status of the labware in each protocol is displayed in the app when the protocol is uploaded.

Troubleshooting Calibration

Troubleshooting a labware definition involves a confirmation of or an adjustment to:
1. Labware height (includes top of the well and inner well bottom definition)
2. X & Y offset (identifies a well's location within a deck slot)

Calibration does *not* account for errors in:
1. Well size (diameter, length, width, depth)
2. Spacing between wells

If the robot is still missing wells or crashing post-calibration, we recommend you double check if measurements in the labware definition match your labware. You can find all the measurements for labware in our API via the [Labware Library](#). If you're confident your physical labware and the labware definition match, but continue to experience issues, contact support@opentrons.com for further help in troubleshooting.

7.3 Labware Library

The Opentrons Labware Library is a visual index of all the labware natively supported by the OT-2. While your OT-2 can be adapted to use a wide range of labware, we've built this as a guide to labware that's been tested for compatibility and is ready to work with right out of the box. You'll find both Opentrons and 3rd party labware, with the information you need to incorporate them into your Protocol Designer or Python API protocols.

We strongly recommend using labware definitions that exactly match the labware you are using, and the Labware Library includes all the details you need to find automation-friendly labware, and to make sure you're using the matched labware definition.

Nested Labware
Sometimes you will need to stack labware on top of a second piece of labware to use it on the OT-2. Examples of this are when you use the Opentrons tube racks and Opentrons aluminum blocks with tubes or plates.

In these instances, you will require only one labware definition to represent the nested labware. For that reason, if you want to calibrate the labware or obtain a height measurement, you must stack the labware first.

Please note that each tube/plate variation used in conjunction with a tube rack or aluminum block will require its own unique definition to ensure accurate pipetting.

Matching labware definitions with labware

We strongly recommend you use labware definitions that are an exact match for your labware. A mismatch between the definition and the labware on the deck can cause problems such as robot crashes or pipette tips getting knocked off mid-protocol.

The easiest way to find the matching labware definition is by comparing the manufacturer ID from your labware to the Labware Library. If a manufacturer ID is unavailable, you can compare the dimensions of your labware to the dimensions available in the Labware Library.

7.4 Using Labware in Protocols

Using labware from the library in your protocols

The Opentrons Labware Library is available for use in Protocol Designer. Labware can be imported by name into your Python API protocols.

Standard labware

See the API documentation for instructions on how to load labware in a Python protocol. You will need to know the labware's *API name* to load it. The API name can be easily copied from the labware's listing in the Labware Library.

APIv2 protocols
Custom labware needs to be added through the Opentrons App before it can be used in a Python APIv2 protocol.

Note: Opentrons protocol development team uses Python APIv2 to write custom protocols.

You should already have a .json file exported from Labware Creator or provided to you by the Opentrons protocol development team.

To add the custom labware definition:
1. In the Opentrons App, go to **Labware** and click **Import**. A file browser will open.
2. Select the .json file representing your custom labware definition.
3. You should see your new labware in the labware list.
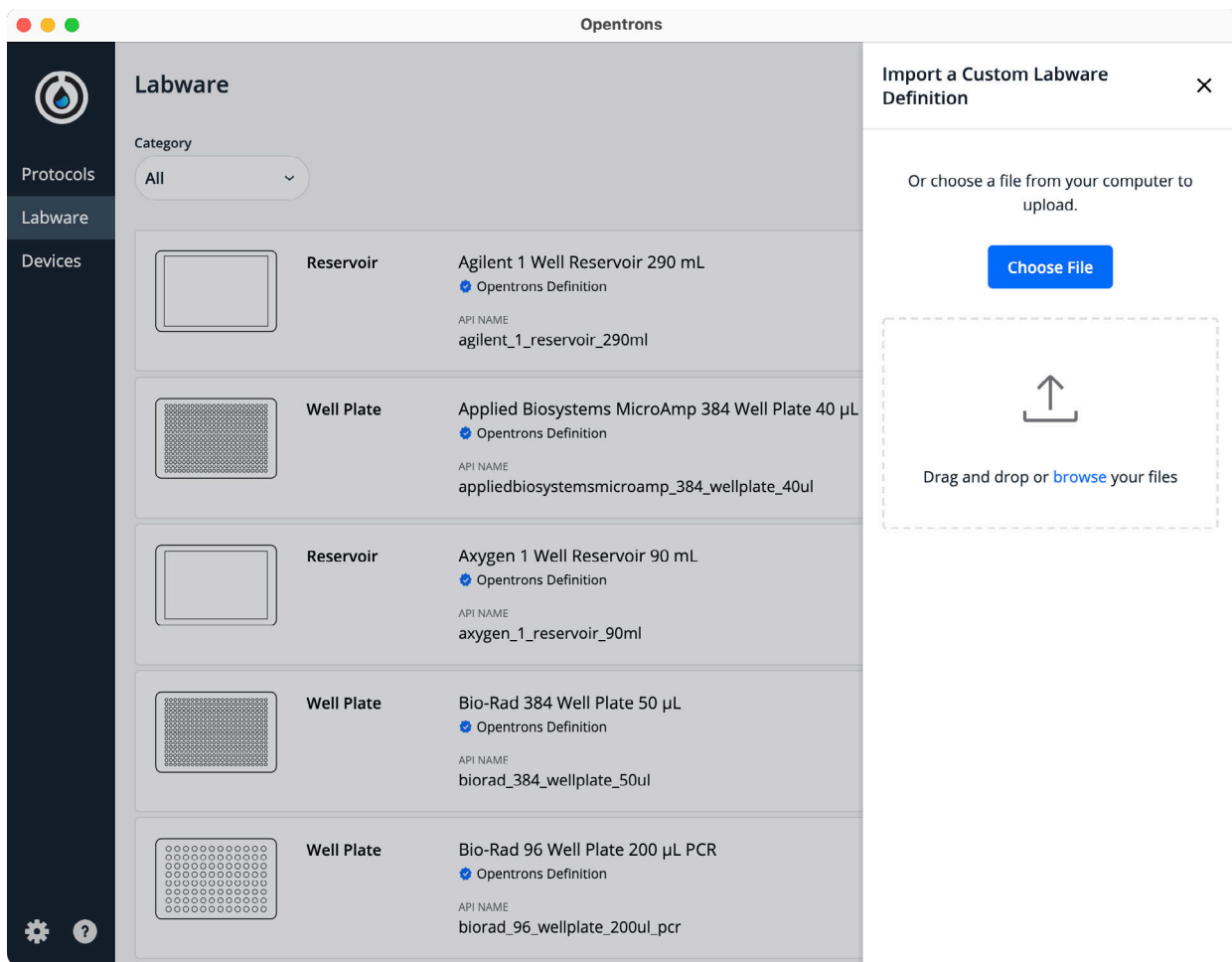
You will need to do this on each computer through which you upload protocols that use that custom labware.

Note: if you make any changes to the .json file on your local computer, you will need to add it to the app again to overwrite the old definition.

After adding the labware through the Opentrons App, protocols can reference it using its API name, like standard labware. See the API documentation for details.

Protocol Designer protocols

Standard and custom labware
Protocol Designer can use any labware in the Labware Library.

When you click **Add Labware** while hovering over a slot in the deck, you'll be presented with a list of labware to choose from. You'll also have the option to import a custom labware definition you've made with our Labware Creator.

Jupyter notebook protocols

Standard labware

As with regular Python protocols, use the `load_labware()` function to load labware.

Custom labware
To use custom labware in a Jupyter notebook, you need to take some extra steps.
1. Upload the definition
2. On the Jupyter Notebook homepage, click **Upload.**
3. Select the .json file corresponding to your custom labware definition.
4. The .json file will appear in the file list. Next to its entry, click **Upload** again.
5. Load the labware

Once you have the custom labware definition file in the Jupyter notebook directory, you can access it from within a Python notebook. Loading the labware for use in the Python Protocol API involves:
1. Loading the contents of the uploaded .json file representing the labware definition.
2. Parsing the JSON text into a Python dictionary.
3. Passing the dictionary to `load_labware_from_definition` to get the labware object.

For example:

```
import json
import Opentrons.execute
protocol = Opentrons.execute.get_protocol_api()
with open('InGen 96 Well Plate 200 µL.json') as labware_file:
    labware_def = json.load(labware_file)
well_plate = protocol.load_labware_from_definition(labware_def, 1)
```

Note that the *file name* `'InGen 96 Well Plate 200 µL.json'` is needed instead of the *API name* `'ingen_96_wellplate_200ul'`.

7.5 Creating custom labware definitions

Any labware that isn't in the Labware Library is custom. You are still able to use custom labware in your protocols by obtaining a custom labware definition.
Please note: We strongly recommend you reference mechanical drawings directly from your supplier/manufacturer and only rely on manual measurements (using calipers when possible) to supplement missing information.
There are two ways to get a custom labware definition.
1. If your labware fits the ruleset below then you can use the Labware Creator to make your own definition.
2. If you can't make your own definition then you can request a custom labware definition from our team.

Labware you can define on your own	Labware you'll need us to help make a definition for
Every well/tube is identical	Wells vary in shape/dimension
All rows are evenly spaced (the space between them is equal).	Not all rows are evenly spaced.
All columns are evenly spaced (the space between them is equal).	Not all columns are evenly spaced.
Fits perfectly in one slot.	Spans less than one slot (requires an adapter) or spans multiple slots.

It can be a bit hard to picture some of the scenarios above, so below we have included a couple examples.
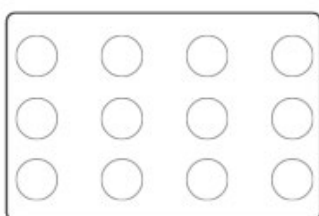
✅ Regular

All columns are evenly spaced and all rows are evenly spaced. Columns do not need to have the same spacing as rows.
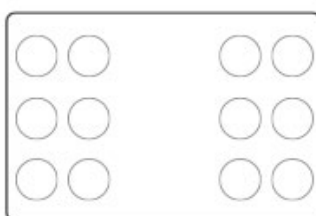
❌ Irregular

Rows are evenly spaced but **columns are not evenly spaced**.
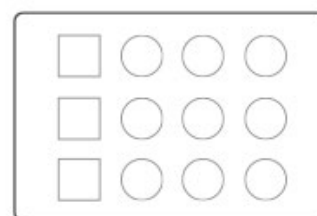
❌ Irregular

Columns/rows are evenly spaced but **wells are not identical**.
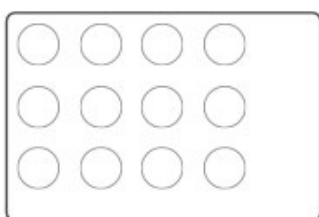
✅ Regular

The grid does not have to be in the center of labware.
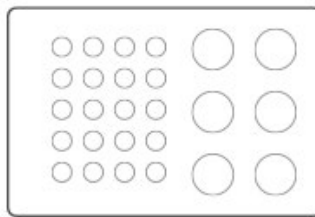
❌ Irregular

There is **more than one grid**.

You can often access technical drawings on the manufacturer product page for your labware. If it's not available there, then you can email the manufacturer and request the "technical drawings" to help with automation. Some brands (such as Agilent, Eppendorf, Corning, Bio-Rad, Thermo Fisher) are more likely to have technical drawings available on their website.

In some cases, the manufacturer's technical drawing may be missing key measurements. If you have to measure by hand, use the following techniques to increase your accuracy.

Tips for measuring labware

- **Use calipers, not a ruler** - We recommend using calipers, as they are more accurate.

- **Measure a couple times and average the result** - One way you can combat variance is to take the measurement a couple of times and then average the result.

- **Depth** - Your calipers may have a specific section used to measure vertical spacing that you can stick inside of a well or tube to measure depth. However, if you are using calipers without that feature or are using rulers you'll have to try a workaround. Stick something stiff (such as a cut out piece of paper or a paperclip, *not* string or thread) into the well/tube. Mark off where the top of the well/tube is. Remove your aid and use your caliper or ruler to measure from the end you stuck into your well/tube to the height.
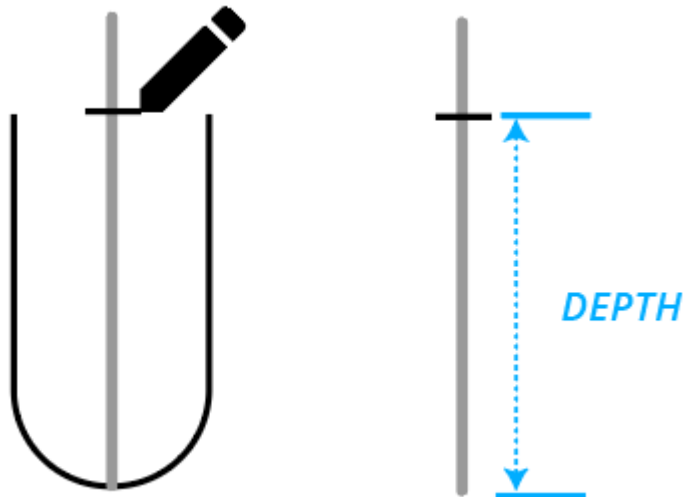
- **Well spacing** - Well spacing is the distance from one well center to the next. When measuring by hand it's advantageous to measure the distance across an entire column/row and then divide by the number of columns/rows.

(measurement / (# of rows - 1)) = Y_s

(measurement / (# of columns - 1)) = X_s
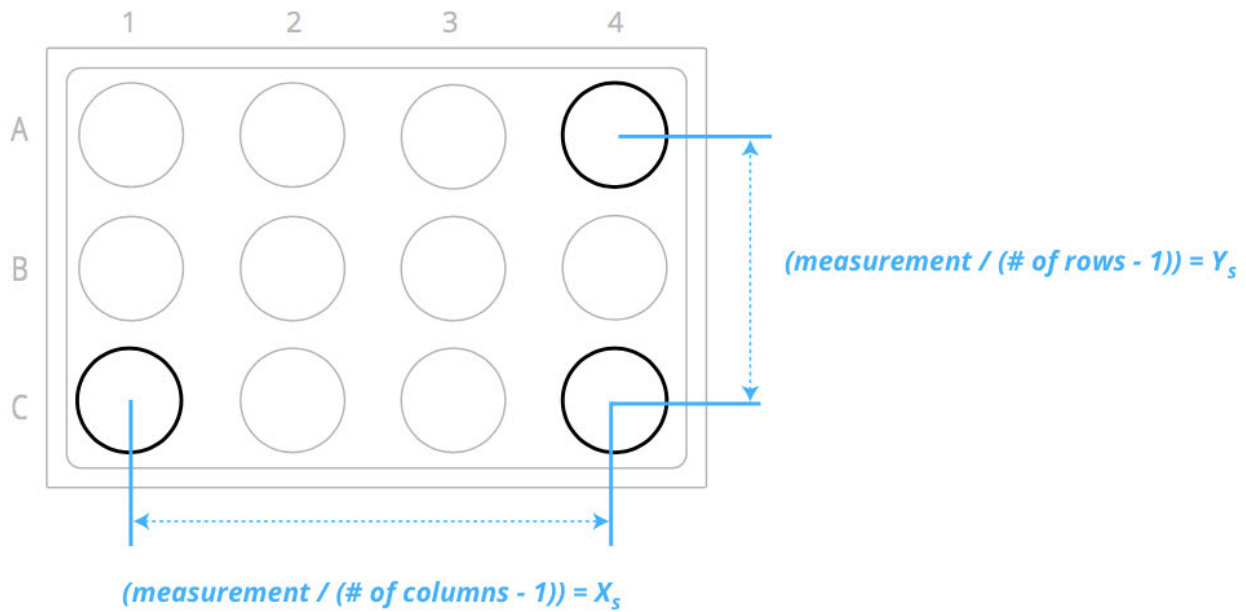
- **Offset -** Measuring the distance from a well's center to the labware's footprint can be difficult. You may find it helpful to push your labware up against a vertical flat edge, and then measure the distance from the well center to that flat edge.

OFFSET
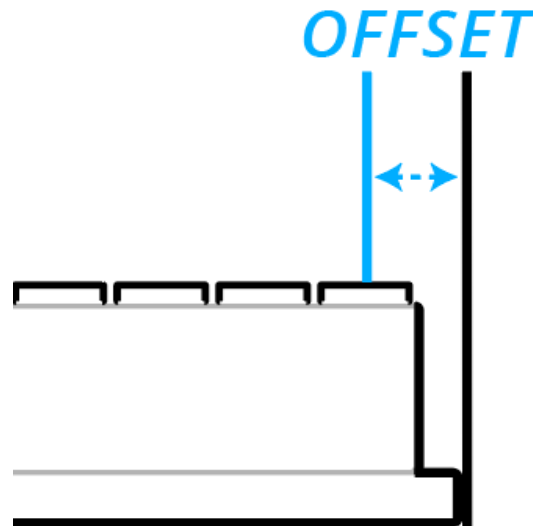
7.6 Requesting a custom labware definition

If the labware you would like to use is not available on the Opentrons Labware Library and you can't create a definition yourself, submit a custom labware definition request.

Please have the following information on hand:
1. The name of the labware
2. A SKU/catalog number
3. A link to where this labware can be found or purchased
4. A technical drawing or the labware including key dimensions

Chapter 8 | Maintenance and Service

8.1 Preventative Maintenance

Below is a recommended schedule for routine maintenance based on average usage. Average usage is estimated to include 4 hours per day, 5 days per week.

1. 20 hours/week
2. 80 hours/month
3. 1000 hours/year

Note that depending on actual usage, the following maintenance steps may be desired more or less frequently.

Maintenance Definitions and Schedule

Daily

Empty Trash Bin
Remove the trash bin, discard used pipette tips, and return the empty trash bin to the OT-2.

Inspect the deck
Check that metal labware clips are not damaged and clear the deck of any debris, liquid, and any labware/modules that will not be used in your protocol.

Weekly

Pipette visual check
Visually inspect the pipette nozzles for any deformations or rubber O-ring notches.

8.2 OT-2 Cleaning

Clean OT-2 as specified in the cleaning documentation. **Do not clean linear bearing rails** or electronics.

8.3 Pipette Cleaning

Clean pipette as specified in the cleaning documentation. Please make sure to remove the pipette from the OT-2 before starting the cleaning procedure.

8.4 Module Cleaning
To clean the Temperature, Magnetic, and Thermocycler Modules, follow the instructions laid out in this chapter.

Clean the Temperature Module

Follow the steps below to clean the Temperature Module and make sure to **not** douse or spray the module directly with cleaning solution:

1. Power down the module.
2. Unplug all module cables.
3. Spray a paper towel with alcohol or bleach and wipe down the module. **Avoid electronics.**
4. Wipe down the following parts of the module with either isopropyl alcohol, ethyl alcohol, or bleach:
    - Aluminum blocks
    - Heating/cooling top plate
    - Module housing/body
    - Sheet metal housing
    - Insulators
5. Do not clean the temperature display, power button, power input port, or the USB port of the Temperature Module.

Clean the Magnetic Module

Follow the steps below to clean the Magnetic Module and do not douse or spray the module directly with cleaning solution:

1. Power down the module.
2. Unplug all module cables.
3. Spray a paper towel with alcohol or bleach and wipe down the module. **Avoid electronics.**
    a. Wipe down the following parts of the Magnetic Module with either isopropyl alcohol or ethyl alcohol:
        - Magnet platform
        - Magnet
    b. Wipe down the following parts of the Magnetic Module with either isopropyl alcohol, ethyl alcohol, or bleach:
        - Module housing/body
        - Sheet metal sides and back
4. Then, wipe down with a dry paper towel.

Clean the Thermocycler Module

When proceeding with the cleaning procedure do **not** douse or spray the module directly with cleaning solution. Do not clean the temperature indicator LED strip, power button, power input port, or the USB port of the Thermocycler Module. Follow the steps below to clean the thermocycler module:

1. Power down the module.

2. Unplug all module cables.
3. Wipe down the following parts of the Thermocycler Module with either isopropyl alcohol, ethyl alcohol, or bleach:
    - PCR top plate
    - Lid heating plate
    - Module housing/body
    - Lid plastic housing
    - Sheet metal housing
    - Rubber seal
4. Then, wipe down with a dry paper towel.

Disinfecting Thermocycler Module seals

The Thermocycler's seals are made of silicone. To clean them:
1. Affix the seal to the thermocycler's lid.
2. Wipe the seal with a 1:10 dilution of household bleach. Let it dry.
3. Wipe the seal with molecular biology grade water. Let it dry.

Note: Thermocycler seals cannot be autoclaved, and each Thermocycler seal should be replaced with a fresh one after approximately 20 Thermocycler runs.

8.5 Maintenance Procedures

Monthly

Check Single-Channel Pipette's Ejection Mechanism
While the OT-2 is powered off, open the door and pull down the tip eject sleeve of the Single-Channel pipette to check that it moves freely.

Power Cycle OT-2 and Modules
Turn the OT-2 and modules off, wait 10 seconds, and then back on again.

Quarterly

Robot Calibration Health Check
Check your OT-2's positional accuracy by running a Calibration Health Check. This procedure identifies if any calibration needs to be updated.

Bi-Yearly

Deck Calibration
Deck calibration adjusts liquid handler motion due to any component wear or major relocation of the instrument within or between labs.

Pipette Offset Calibration
Pipette offset adjusts the calibration of the pipettes relative to the liquid handler deck.

Tip Length Calibration
Tip length calibration checks or adjusts tip length. This is also recommended to be performed if installing a new pipette.

Labware Calibration
Labware calibration adjusts the calibration of the liquid handler to specific labware being utilized. Or perform if liquid handler positioning to any designated labware appears to be offset.

Yearly

Pipette Replacement
Replace pipette entirely with a new unit.

Air Clean Electronics Board
Applying compressed air or electronics duster to the electronics board on the OT-2.

As Needed

Replace the O-rings on your 8-Channel GEN2 pipette

Replace the rubber *O-ring* wrapped around the outside of each channel if you notice any nicks or tears.

Adjusted Schedule

Action	Recommended Schedule	Adjusted Schedule
Empty Trash Bin	Daily	If the trash bin reaches capacity.
Inspect the Deck		N/A
Pipette visual check	Weekly	If pipette aspirate or dispense actions appear to show any signs of malfunction or pipetting inaccuracy.
OT-2, Module, and Pipette Cleaning		Immediately after contamination or spillage occurs, or if the pipette and/or module look dirty.

Check Single-Channel Pipette's Ejection Mechanism	Monthly	If the pipette does not eject/pick up tips consistently.
Power Cycle OT-2 and Modules		N/A
Robot Calibration Health Check	Quarterly	If movement appears to be malfunctioning or incorrect. Or after deck calibration, pipette offset calibration, or tip length calibration.
Deck Calibration	Bi-Yearly	If OT-2's Calibration Health Check shows inaccurate calibration results. Or if installing a new pipette. Or if pipette positioning appears to cause crashing into the deck or labware.
Pipette Offset		If OT-2's Calibration Health Check shows inaccurate calibration results. Or if installing a new pipette. Or if pipette positioning appears to cause crashing into the deck or labware.
Tip Length Calibration		If installing a new pipette.
Labware Position Check		If liquid handler positioning to any designated labware appears to be offset. Or if installing and using new types of labware from previous runs.
Pipette Replacement	Yearly	If pipette aspirate or dispense actions appear to show any signs of malfunction or pipetting inaccuracy.
Air Clean Electronics Board		If noticeable dust and debris have accumulated on the black electronics cover.

Maintenance that should not be done on the OT-2

Do not clean or grease the linear rail

The OT-2 linear rails come with a specific oil on them. Removing this oil decreases the longevity of the rails/bearings.

Do not grease lead screws

Greasing the lead screw can introduce particulates or contaminants, which can decrease the longevity of the lead screw.

8.6 Maintenance Tools and Materials

- Torx key (T10)
- Hex key set (1.5 mm, 2 mm, 2.5 mm, 3 mm)
- M3x14mm socket head screws (2x/pipette), M3 hex nut
- Super Lube (silicone-based lubricant) - reserved for specific troubleshooting and when instructed to be used by the support team
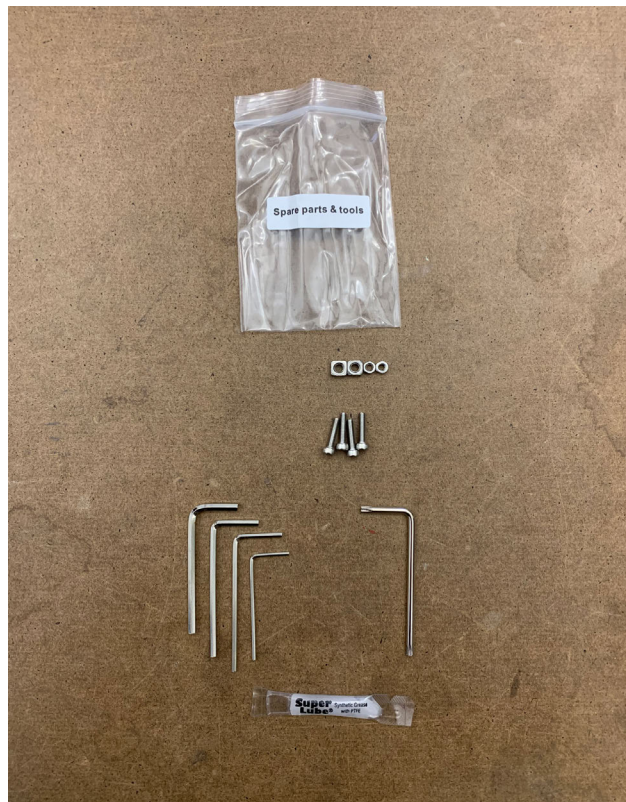
Chapter 9 | Troubleshooting

9.1 Eight Channel Pipette Rotated Relative To Labware

What does this issue look like?

When the pipette moves to some labware, it's visibly rotated around its vertical axis. If you look at the pipette from above, it would look like this:
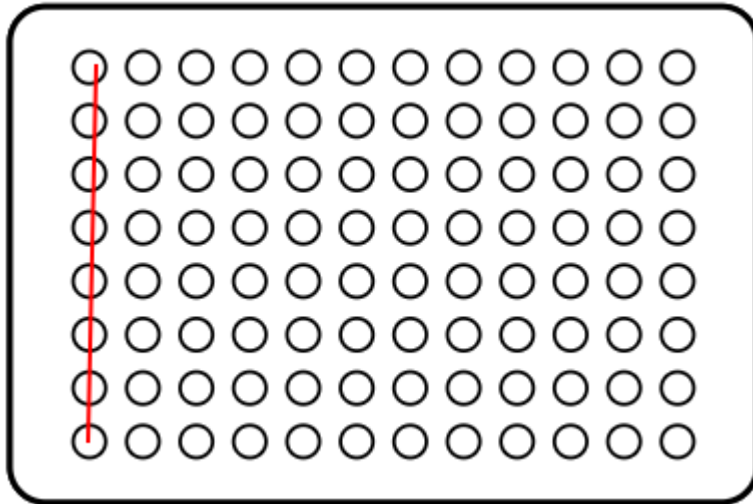
This issue is not to be confused with the pipette being tilted around the *x-axis,* which is a different problem. If you are experiencing that issue please see our guide on Leveling 8-Channel Pipettes.

When can this happen?

This potentially can happen with any 8-Channel pipette. This could technically happen with a Single-Channel pipette, too, but it would not impact the performance of the pipette.

What are the possible causes and how do I solve them?

Cause	Resolution
The labware is inserted incorrectly in its deck slot, leaving it rotated.	Reinsert that labware correctly so that you see the aluminum deck slot dividers around all 4 sides of the labware.
The pipette is unevenly screwed onto its mount.	1. Remove and reattach that pipette. 2. Make sure the pipette is screwed in evenly. 3. Make sure there is nothing sandwiched between the pipette and mount. 4. If this fixes the issue you can redo pipette offset calibration.
The whole removable deck is slightly rotated relative to the gantry.	Loosen and rotate the whole *deck.* If this fixes the issue, redo deck calibration when you're done.
The two pipette mounts are not perpendicular to each other.	Please contact Opentrons Support ✉

9.2 Accessing and replacing hardware

Accessing the OT-2's motor controller board

The motor controller board is responsible for electrically driving the motors that move the pipettes around the OT-2. If you're working with Opentrons Support to resolve certain technical problems, we might ask you to inspect or replace it.
This section walks through:

1. Removing the cover of the pipette carriage to access the board.
2. Disconnecting, removing, and replacing the board.

Accessing the board
Warning: Turn off the OT-2 before beginning.

1. The board is mounted inside the pipette carriage. Using the driver that came with the OT-2, remove the 4 screws labeled below from the top and back of the pipette carriage.
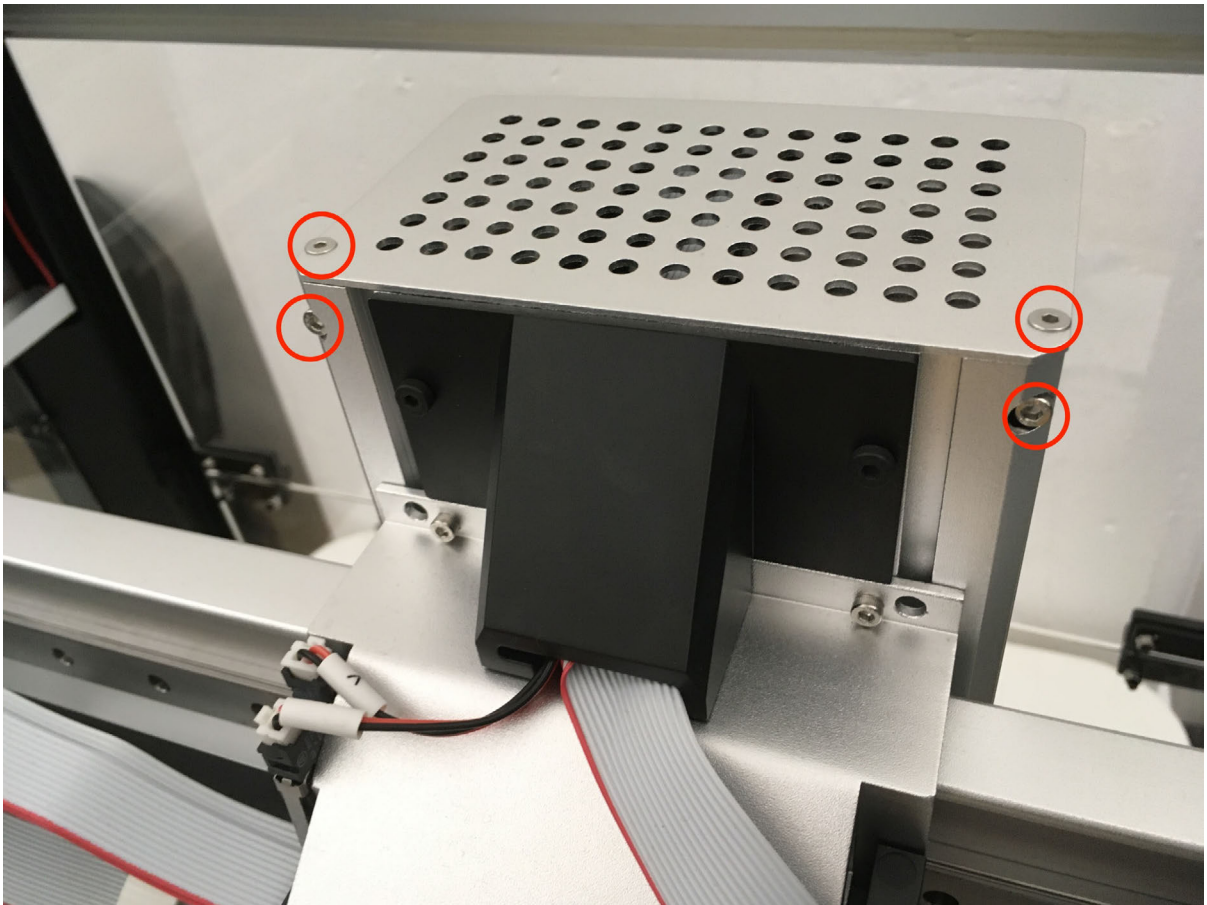
2. Remove the remaining 4 screws, labeled below, from the back of the pipette carriage.
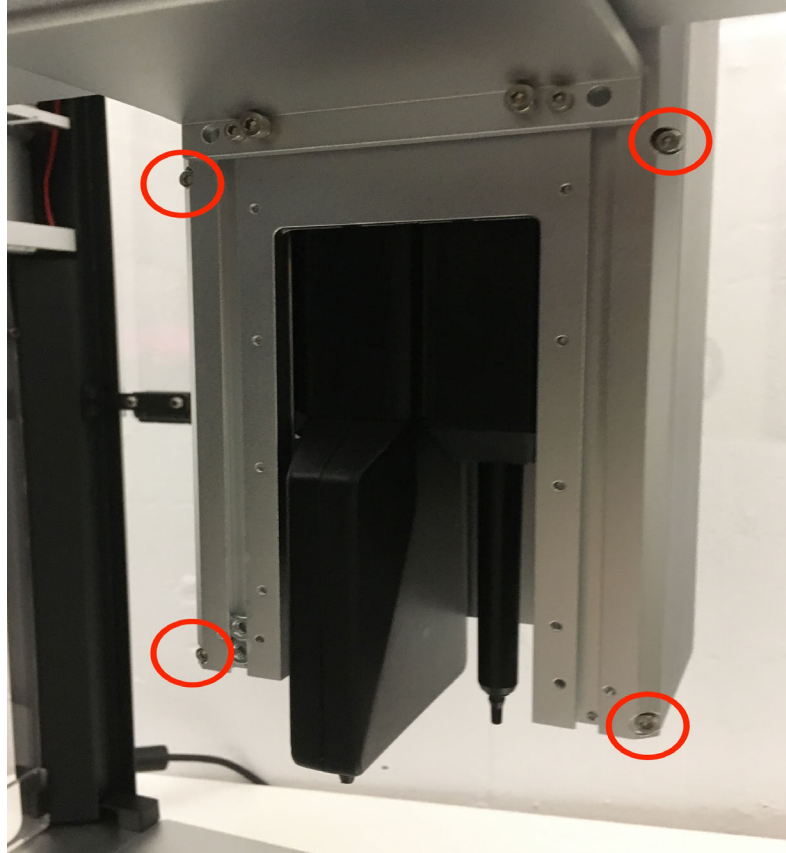
3. Slide the front cover of the pipette carriage down, and then pull it off. The motor controller board will be exposed.

Removing the board for replacement

⚠️ **Warning:** Turn off the OT-2 before beginning.

1. Unplug the 8 connectors circled below:
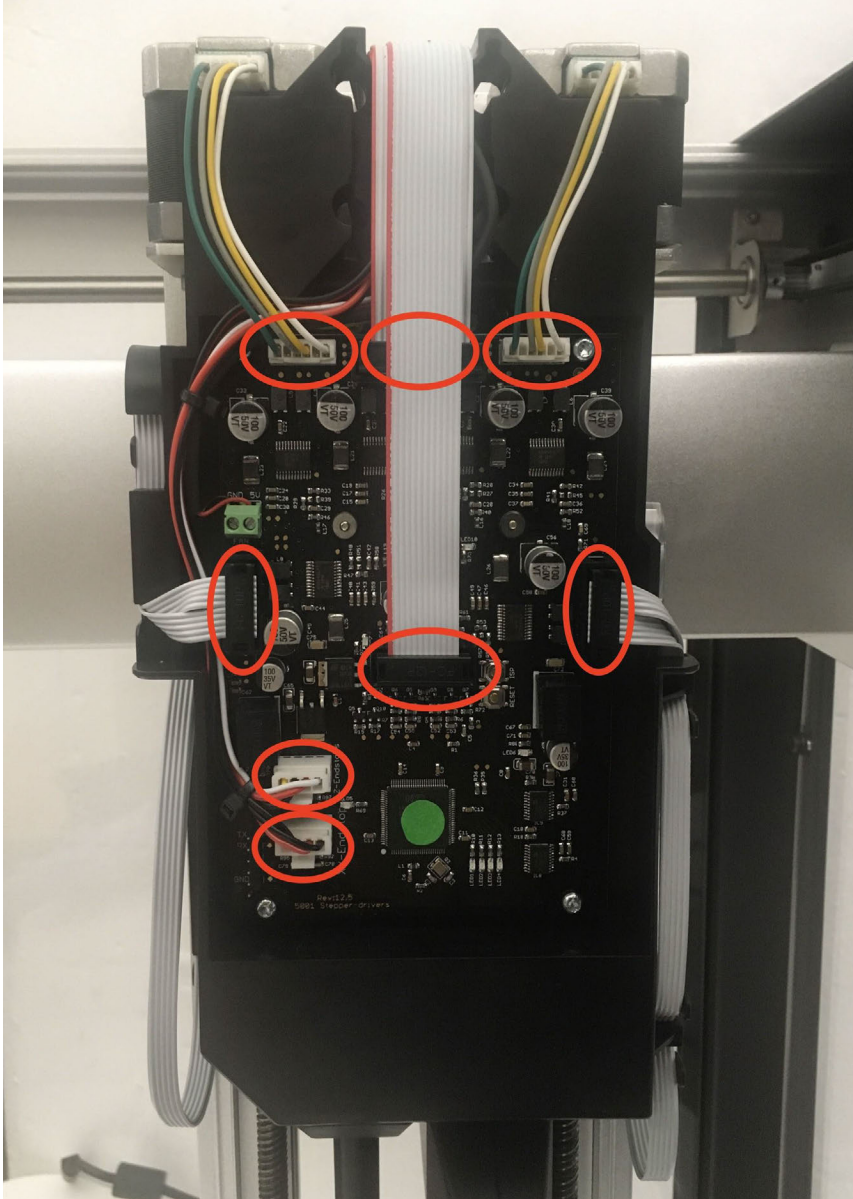
2. Unscrew the remaining 2 wires with a flathead screwdriver:
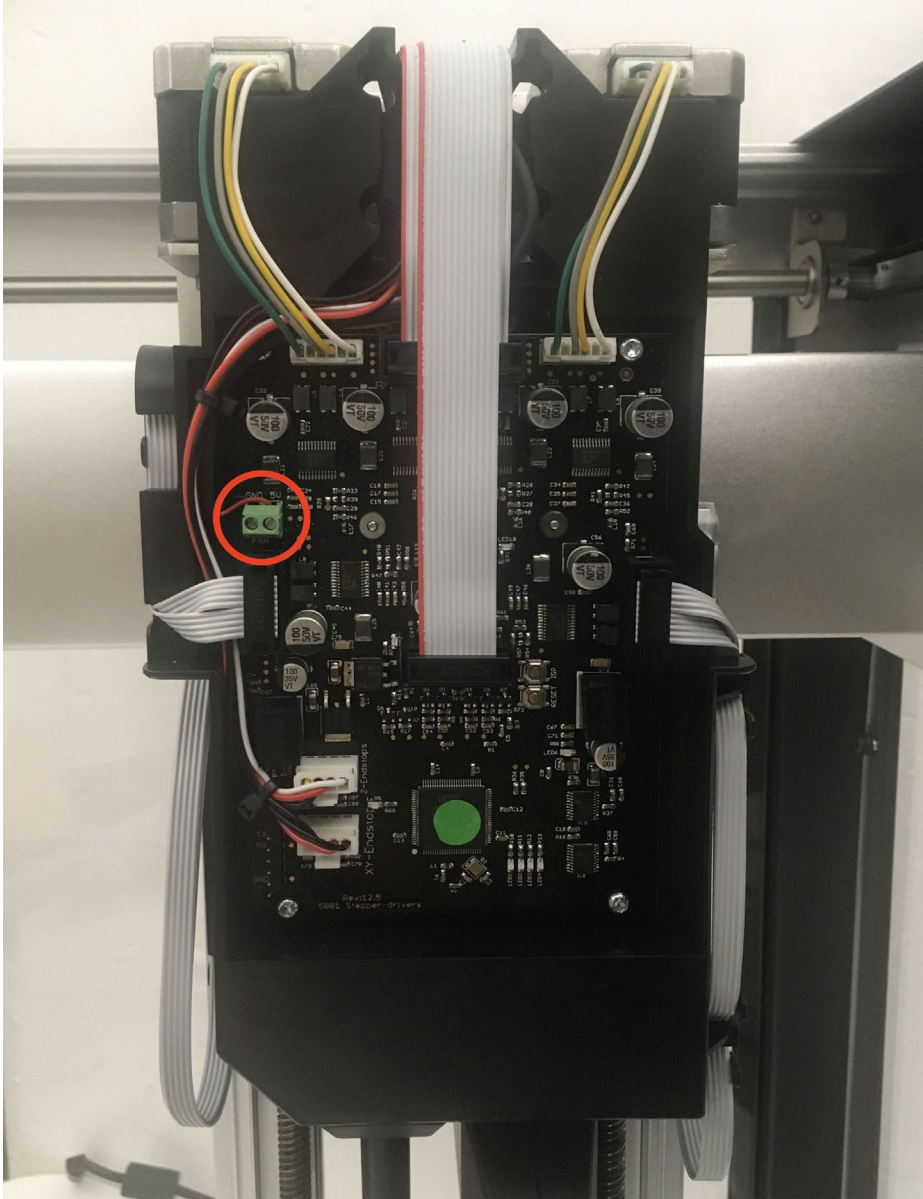
3. Once all cables are unplugged, unscrew the 4 screws securing the board to the carriage:
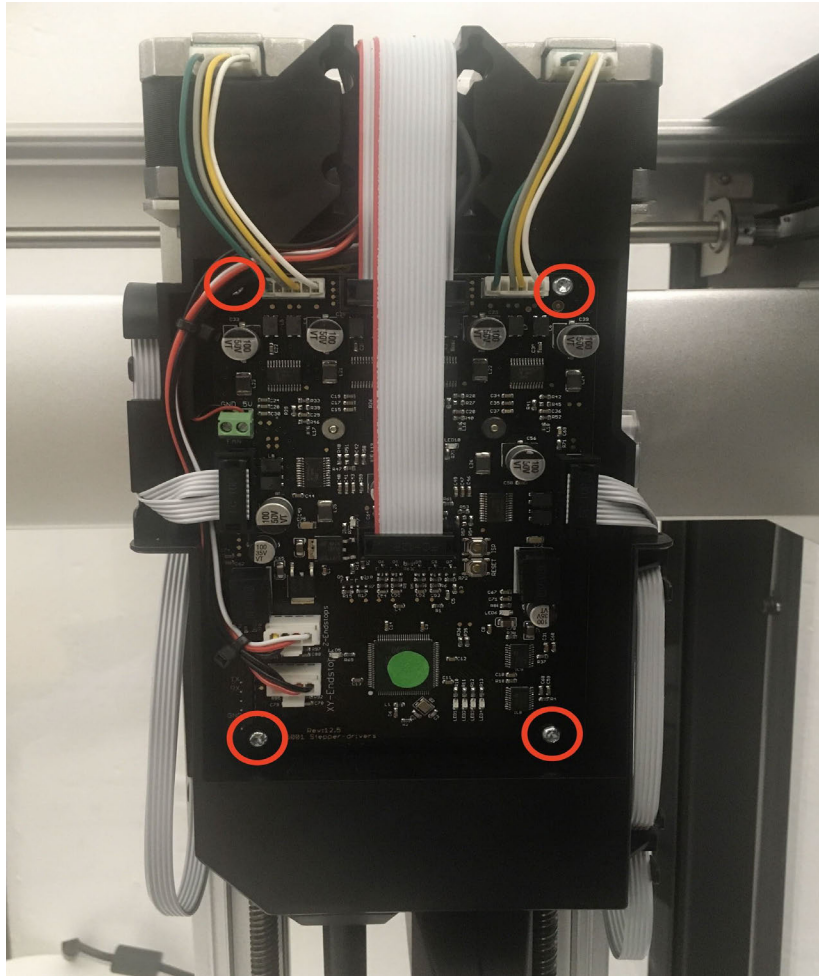
4. Pull the board out.

Installing a new board

Install a new board by following the above steps in reverse.

1. Insert the board and secure it by screwing in the 4 corner screws.
2. Insert the two bare-ended wires into the screw terminal. Use a flathead screwdriver to tighten the screws. The black wire goes into the left slot (labeled *GND),* and the red wire goes into the right slot (labeled *5V).*
3. Insert the 8 push connectors.
4. Make sure the OT-2 works with the new board before reattaching the cover.

Accessing the OT-2's rear electronics

On the back of the OT-2, there are some electronics that are normally concealed by a black cover. While troubleshooting, Opentrons Support might request for this cover to be removed to expose the electronics.

Materials

- T10 Torx (star) key provided with the robot

Before beginning to remove the electronics cover

Warning: Turn off the OT-2 before beginning.

1. **Unscrew the back screws.**
   Find the vent on the back, upper portion of the robot. Using the T10 Torx driver, remove the nine screws shown below to open up the electronics enclosure cover. Note that some of the screws will be shorter than the others.
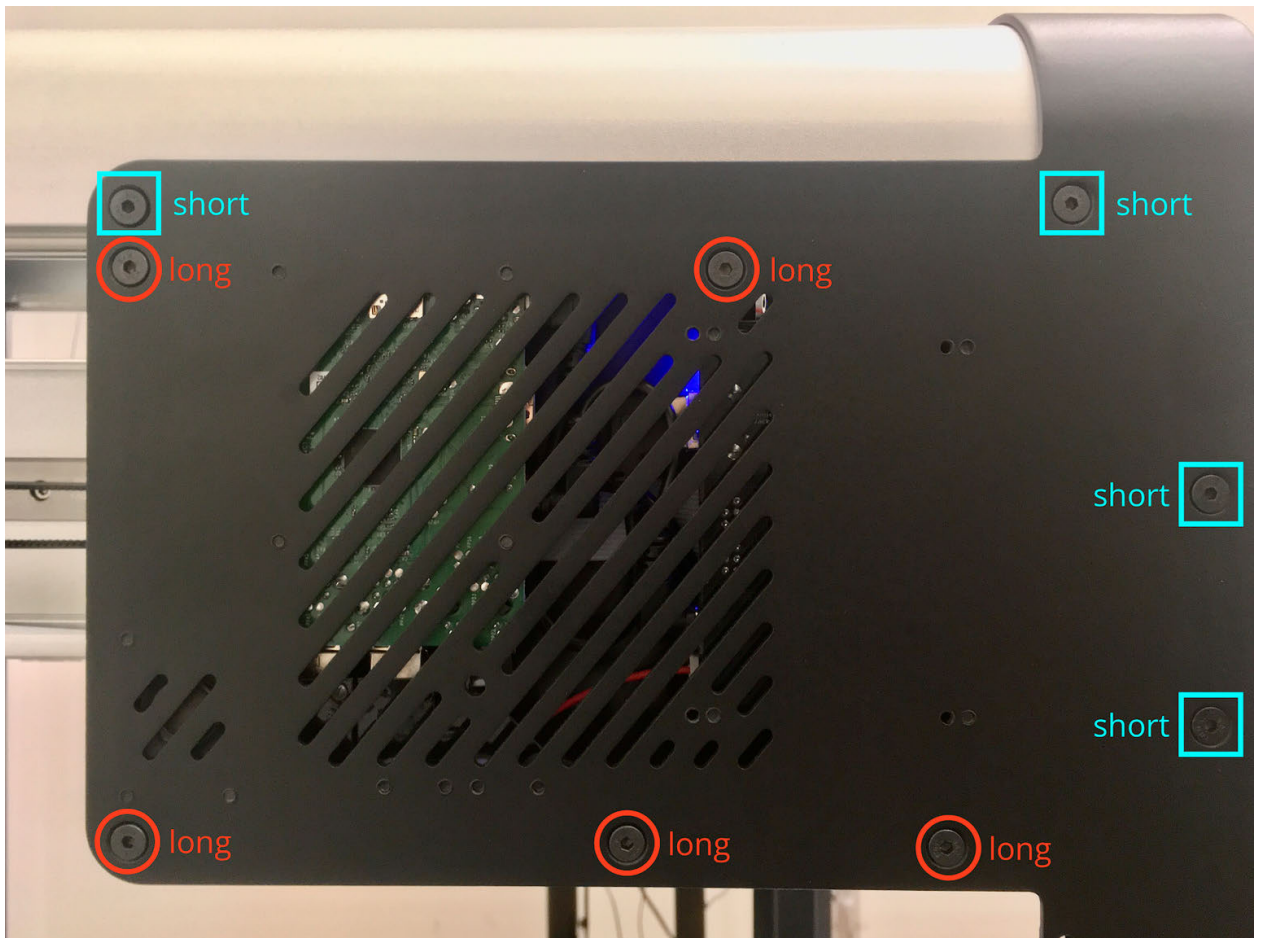
2.

3. **Remove the metal panel.**
4. Pull the black metal panel off.
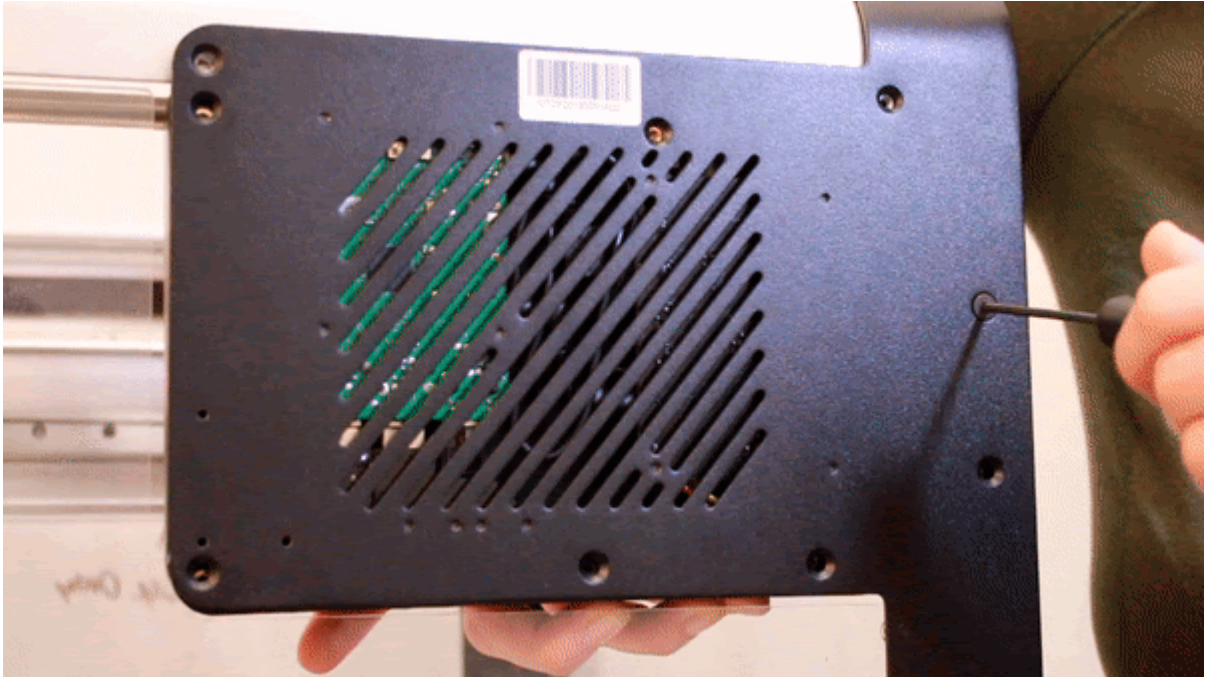
5. **Electronics layout**
   - The black central routing board (right)
   - The green Raspberry Pi (left)
   - Various cables
   - **Update**: The gray grounding strap is now located in the upper right (previously it was in the upper left as shown in the image below)
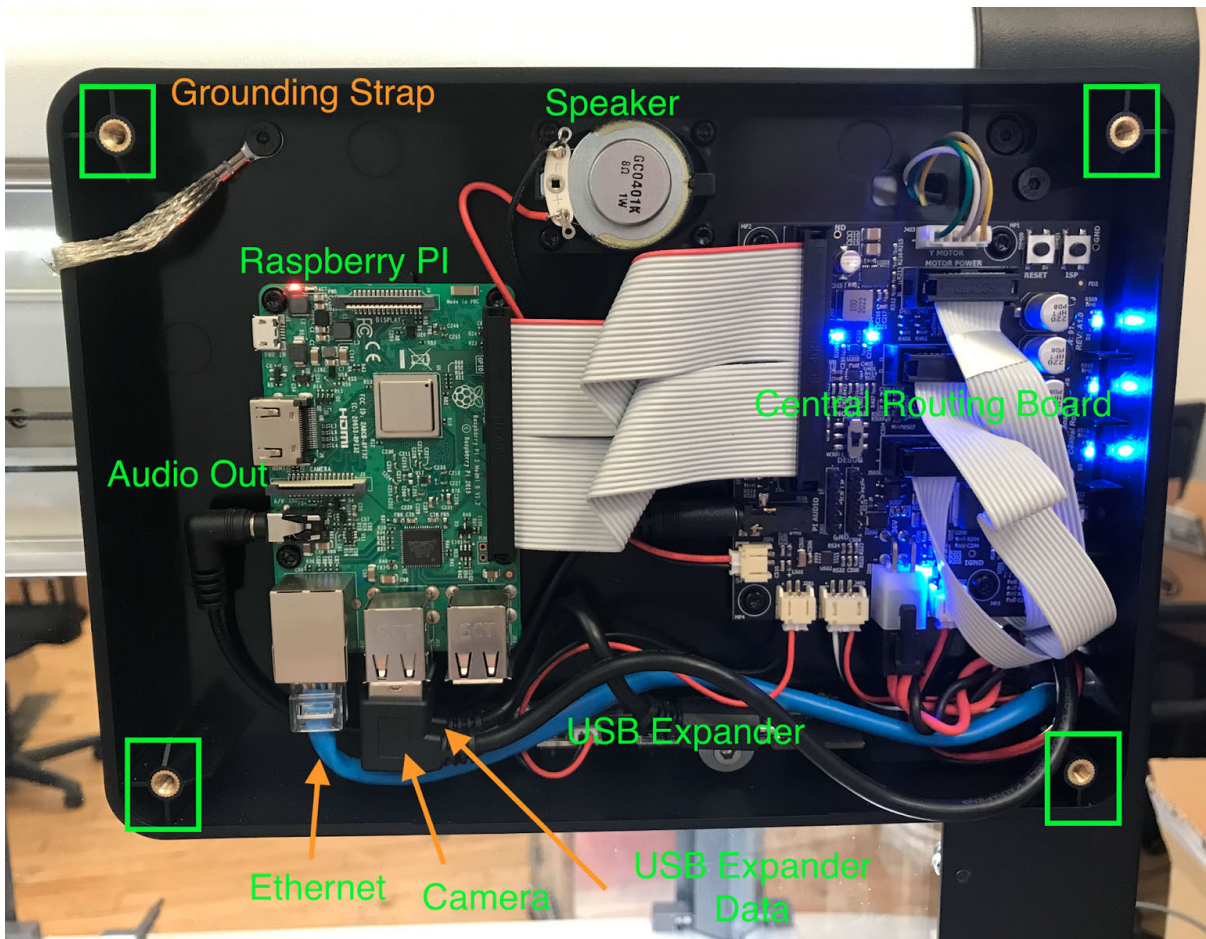
6. **Put the metal panel back on.**
   Important: Before putting the metal panel back on, notice that the inside of the panel is partially unpainted. This is to establish a solid electrical connection from the panel to the gray grounding strap cable (located in the upper right within the electronics enclosure). When putting the panel back on, it is important to **check that the gray grounding strap cable is sandwiched between two pieces of metal** when it is fastened. Re-fasten the electronics enclosure while ensuring that the gray grounding strap cable is sandwiched between two pieces of metal.

9.3 Hardware troubleshooting on the OT-2

Although hardware issues are not always linked to *SmoothieErrors*, if you do happen to encounter one, it is helpful to determine if you are experiencing a *homing fail* or *hard limit* error before proceeding with the troubleshooting process. Additionally, you might need to *investigate the electronic boards* for further troubleshooting.

Hardware overview

Constant or repeating SmoothieErrors on one of or both mounts can be an indication of a hardware failure, specifically with the motor controller board, central routing board (CRB), and/or Raspberry Pi (RPi) with which it communicates. To diagnose these, you can investigate the boards yourself and send the Support team images, videos, and information about what is happening to expedite troubleshooting.

Data collection procedure

These are instructions you can follow to start examining the internal hardware:

1. Power off the OT-2 and access the motor controller board and the rear electronics (CRB and RPi boards are housed here).
2. Follow the cables leading from the rear electronics to the motor controller board to check for damaged cables (see pictures below for specific cables and connections). If there are any damaged cables, send some pictures over to the Support team showing both the damage and the slot into which the cable connects, so we can properly identify the cable.
3. Take a video of the 3 boards (motor controller board, CRB, and RPi) powering on when you turn on the OT-2. Send this over to Support so we can confirm everything is powering on correctly. You can also refer to our article on investigating the electronic boards for information on the LED indicators.

Hardware details

Motor controller/stepper driver board

The motor controller board is responsible for electrically driving the motors that move the pipettes around the OT-2. As a result, there are a lot of cables and connections that each control a specific movement. The above schematic outlines the connection slots on the board and what motor/logic this connection is monitoring.
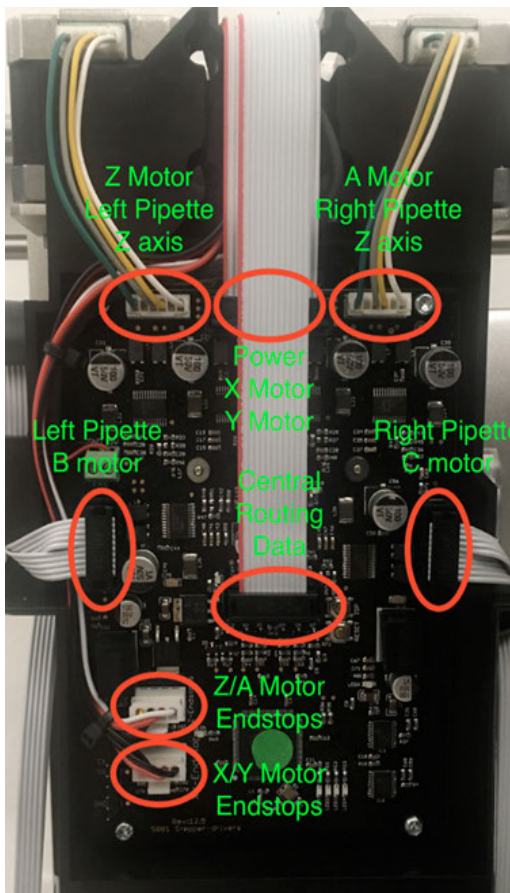
If experiencing a SmoothieError:

1. Given the mount/axis shown in the actual error (such as Hard limit +B or ...G28.2Z returned ALARM: Homing fail, etc.), you can narrow down which cables would most likely be responsible for throwing an error if there was a hardware failure on that connection.
2. For example, with a **Hard limit +B**, we know our area of interest is the **plunger of the pipette attached to the left mount** and, thus, should first check that the **Left Pipette B motor** is properly connected and not damaged.

Raspberry Pi (RPi) and Central routing board (CRB)
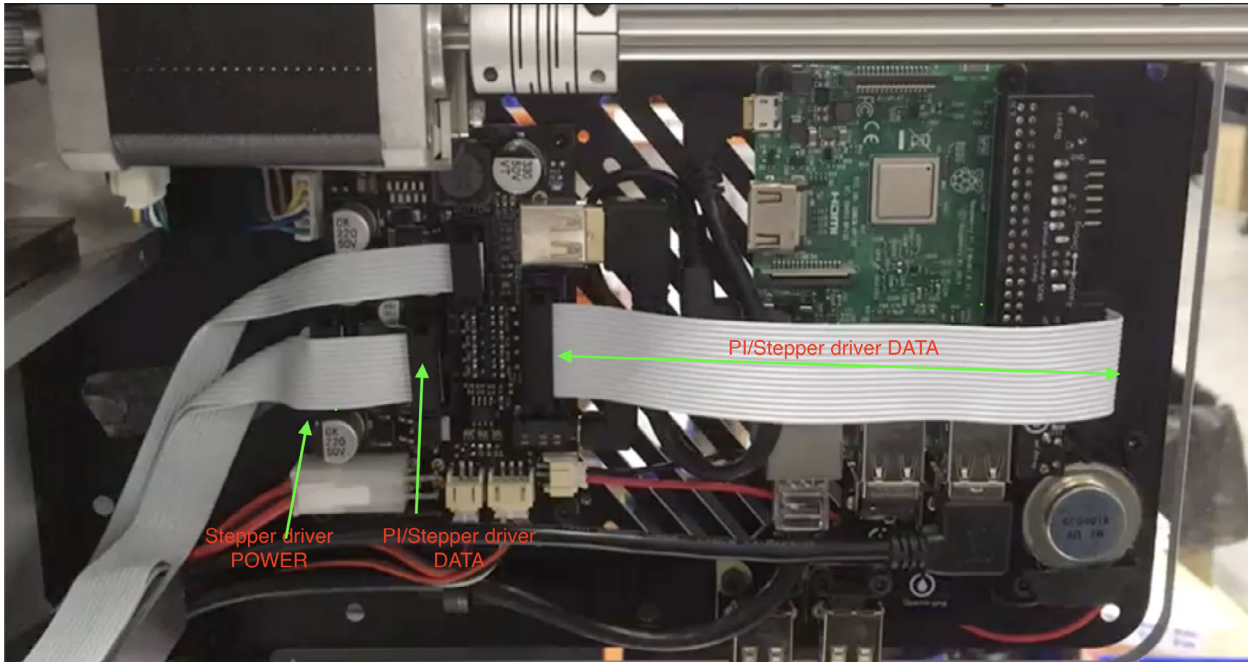
The RPi uses a serial protocol to command the motor controller board, the software running on the RPi sending G-code to the motor controller board. The motor controller board then receives it, interprets it, and electrically drives the gantry and pipette motors accordingly. It also handles connectivity with the USB/Ethernet adapter and Wi-Fi as well as monitoring the USB ports for the modules. The CRB helps deliver the data from the RPi to the motor controller board as well as deliver power to the motor controller board.

Troubleshooting steps

With this chain of data transmission, any improperly connected cable, broken cable, or damaged part of any board can be responsible for the error the OT-2 is experiencing. **Thus, it's best to check your cable connections before going straight to replacing the boards.**

1. **Reconnect motor controller board cables**. Unplug and reconnect all the cable connections shown on each of the three boards. As mentioned above, you can unplug and reconnect specific connections on the motor controller board first depending on the type of error you received, and then test to see if that solved the problem. If not, you can try reconnecting all the cables.

2. **Replace the boards of the OT-2.** If you need to replace the boards, it is best to try and see which board actually resolves the problem, as we can use this to identify the exact origin of the problem for future reference. It would be helpful, for future records, to check through each board to see which replacement solves the issue. Opentrons recommends the following:
   a. Swap the Raspberry Pi + Raspberry Pi shield and test if the robot powers on, homes, and connects to your computer properly.

b. If that doesn't work, swap in the new CRB and test again.
c. If that still doesn't resolve it, swap in the motor controller board.

If these new boards do not resolve the error, please reach out to our Support team.

9.4 GEN2 P300 8-Channel pipette isn't ejecting tips

When you attach a GEN2 P300 8-Channel pipette to your robot for the first time, you may encounter an issue with the pipette not ejecting or dropping tips if you are using an OT-2 with the incorrect bottom pipette attachment screw. This can be resolved by following the instructions below.

The OT-2 introduces a new mount design and requires the **6 mm bottom pipette attachment screw** (**not** the 14 mm screw used with OT-2s) to properly function with Opentrons' 8-Channel pipettes:
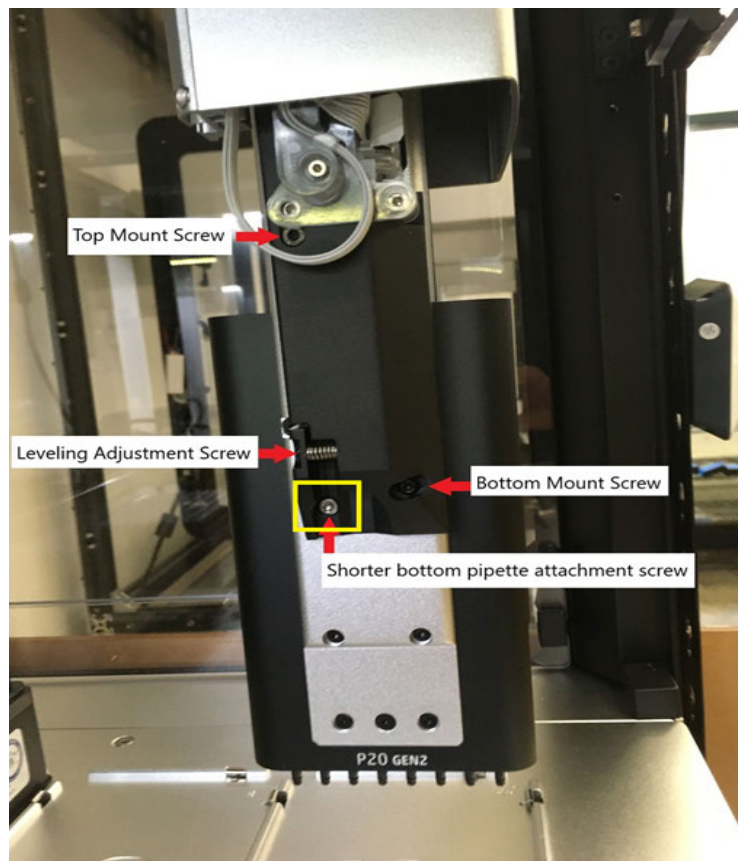
If you use the 14 mm screw as the bottom pipette attachment screw on an OT-2 mount, the screw will make contact with the plunger of the pipette and prevent it from dropping tips. Putting in the new (6 mm) screw will allow the 8-Channel to work properly. Be sure to confirm you are using the shorter screw while attaching an 8-Channel pipette to an OT-2 robot.

9.5 How to download the OT-2 logs

A log file is a computer-generated data file that contains information about usage patterns, activities, and operations within an operating system, application, server, or another device. The OT-2 generates three different log files (API, serial, and server) that we will explore in this article.

API logs

Similar to serial logs, the OT-2 keeps a record of the API level commands. The commands are more intuitive to read. If you are using our Python API, they should be easier to understand.

How to understand API commands

The messages that are printed on the API logs closely resemble the commands you would use when developing protocols with our Python API. Our API was designed to make writing protocols easy and accessible to everyone.

Not only do these logs resemble our Python API, but they also resemble the actions that are being performed by the OT-2 during a protocol.

Starting from:
Dec 14 20:20:55 falling-forest opentrons-api[233]: Updating working volume on pipette mount:RIGHT, tip volume: 300 ul
Dec 14 20:20:55 falling-forest opentrons-api[233]: InstrumentContext.aspirate: {'volume': 50, 'location': Location(point=Point(x=63.88, y=42.74, z=5.550000000000001), labware=A1 of rez on 1), 'rate': 1.0}

1. Updating working volume on pipette mount:RIGHT, tip volume: 300 ul is telling us the pipette on the right mount will be used in the following commands.
2. InstrumentContext.aspirate is telling us that the pipette is going to aspirate with this pipette.
3. 'volume': 50, 'location': Location(point=Point(x=63.88, y=42.74, z=5.550000000000001), labware=A1 of rez on 1), 'rate': 1.0 is telling us the volume, the location, and the rate of the aspirate step.

This line of logs says the pipette on the right mount is going to aspirate 50 μL of liquid from well A1 of the labware in slot 1.
 Example
Below is a snippet of what the API log file would look like in an aspirate/dispense cycle.

1Dec 14 20:20:55 falling-forest opentrons-api[233]: Updating working volume on pipette mount:RIGHT, tip volume: 300 ul

2Dec 14 20:20:55 falling-forest opentrons-api[233]: InstrumentContext.aspirate: {'volume': 50, 'location': Location(point=Point(x=63.88, y=42.74, z=5.550000000000001), labware=A1 of rez on 1), 'rate': 1.0}

3Dec 14 20:20:55 falling-forest opentrons-api[233]: InstrumentContext.dispense: {'volume': 50, 'location': Location(point=Point(x=146.88, y=74.24, z=1.42), labware=A1 of NEST 96 Well Plate 100 µL PCR Full Skirt on 2), 'rate': 1.0}

4Dec 14 20:20:55 falling-forest opentrons-api[233]: InstrumentContext.aspirate: {'volume': 50, 'location': Location(point=Point(x=63.88, y=42.74, z=5.550000000000001), labware=A1 of rez on 1), 'rate': 1.0}

5Dec 14 20:20:55 falling-forest opentrons-api[233]: InstrumentContext.dispense: {'volume': 50, 'location': Location(point=Point(x=155.88, y=74.24, z=1.42), labware=A2 of NEST 96 Well Plate 100 µL PCR Full Skirt on 2), 'rate': 1.0}

6Dec 14 20:20:55 falling-forest opentrons-api[233]: InstrumentContext.aspirate: {'volume': 50, 'location': Location(point=Point(x=63.88, y=42.74, z=5.550000000000001), labware=A1 of rez on 1), 'rate': 1.0}

7Dec 14 20:20:55 falling-forest opentrons-api[233]: InstrumentContext.dispense: {'volume': 50, 'location': Location(point=Point(x=164.88, y=74.24, z=1.42), labware=A3 of NEST 96 Well Plate 100 µL PCR Full Skirt on 2), 'rate': 1.0}

Serial logs

The Raspberry Pi on the OT-2 uses a serial protocol to command the motor controller board. The OT-2 logs these commands in the form of G-codes. G-code is a common language for controlling things like 3D printers and CNC liquid handlers.

For example, you might give a liquid handler the G-code command G0 X10 to tell it to move 10 mm along its X-axis, or the G-code command G28.2 Y to tell it to home its Y-axis.

The OT-2 uses G-code internally for controlling its gantry, pipettes, and modules.

- The software running on the Raspberry Pi sends G-code to the motor controller board. The motor controller board then receives it, interprets it, and electrically drives the gantry and pipette motors accordingly.
- It also sends G-code over a USB cable to modules, which interpret it and perform their module operations.

How to understand G-code
Starting from:
1M907 A0.1 B1.0 C0.05 X0.3 Y0.3 Z0.1 G4P0.005 G28.2B

1. M907 is code for "set the motors' electrical current." A0.1 B1.0 C0.05 X0.3 Y0.3 Z0.1 specifies the new current for each axis (0.1 amps on the A-axis, 1.0 amps on the B-axis, and so on).
2. G4P0.005G4 is code for "dwell" (pause). P0.005 specifies that the dwell should be 0.005 seconds.
3. G28.2B G28.2 is code for "home." B specifies that the home should be on the B-axis.

So, in short, this line of G-code sets the motor currents, waits very briefly, and then homes the B-axis.

Example

1Feb 17 21:40:05 blue-cloud opentrons-api-serial[237]: smoothie: Write -> b'M400\r\n\r\n'
2Feb 17 21:40:05 blue-cloud opentrons-api-serial[237]: smoothie: Read <-
b'M400\n\no\r\nk\r\nok\r\nok\r\n'
3Feb 17 21:40:05 blue-cloud opentrons-api-serial[237]: smoothie: Write -> b'M18ZACB\r\n\r\n'
4Feb 17 21:40:05 blue-cloud opentrons-api-serial[237]: smoothie: Read <-
b'M18ZACB\r\no\r\nk\r\nok\r\nok\r\n'
5Feb 17 21:40:05 blue-cloud opentrons-api-serial[237]: smoothie: Write -> b'M400\r\n\r\n'
6Feb 17 21:40:05 blue-cloud opentrons-api-serial[237]: smoothie: Read <-
b'M400\ro\n\r\nk\r\nok\r\nok\r\n'
7Feb 17 21:40:05 blue-cloud opentrons-api-serial[237]: smoothie: Write -> b'G4P0.1\r\n\r\n'
8Feb 17 21:40:05 blue-cloud opentrons-api-serial[237]: smoothie: Read <-
b'G4P0.1\r\n\r\nok\r\nok\r\n'
9Feb 17 21:40:05 blue-cloud opentrons-api-serial[237]: smoothie: Write -> b'M400\r\n\r\n'
10Feb 17 21:40:05 blue-cloud opentrons-api-serial[237]: smoothie: Read <-
b'M400\r\n\n\rk\n\r\nok\r\nok\r\n'
11Feb 17 21:40:05 blue-cloud opentrons-api-serial[237]: smoothie: Write -> b'M369R\r\n\r\n'

G-code references

Consult the OT-2's Python source code to see what each G-code command means.

What the G-code is controlling	The relevant source code file
OT-2 gantry and pipettes	api/src/opentrons/drivers/smoothie_drivers/ driver_3_0.py

Magnetic Module	api/src/opentrons/drivers/mag_deck/driver.py
Temperature Module	api/src/opentrons/drivers/temp_deck/driver.py
Thermocycler Module	api/src/opentrons/drivers/thermocycler/driver.py

In these files, try Ctrl+F or Command+F searching for the first part of the command (like G0 or M907).

Server logs

On software versions 4.1+ you will be able to download a third log file referred to as the server log. These logs encompass both the built-in response logs (very useful for figuring out interaction patterns) and errors that bubble up to the server logging handlers.

Example:

1Feb 11 17:57:56 CA9520E280A2E1B4A5E280A2CA94 uvicorn[239]: INFO: - "GET /labware/calibrations HTTP/1.1" 200 OK
2Feb 11 17:57:59 CA9520E280A2E1B4A5E280A2CA94 uvicorn[239]: INFO: - "GET /calibration/tip_length HTTP/1.1" 200 OK
3Feb 11 17:57:59 CA9520E280A2E1B4A5E280A2CA94 uvicorn[239]: INFO: - "GET /labware/calibrations HTTP/1.1" 200 OK
4Feb 11 17:57:59 CA9520E280A2E1B4A5E280A2CA94 uvicorn[239]: INFO: - "GET /pipettes HTTP/1.1" 200 OK
5Feb 11 17:58:00 CA9520E280A2E1B4A5E280A2CA94 uvicorn[239]: INFO: - "GET /modules HTTP/1.1" 200 OK
6Feb 11 17:58:01 CA9520E280A2E1B4A5E280A2CA94 uvicorn[239]: INFO: - "GET /sessions HTTP/1.1" 200 OK

Downloading log files via Opentrons App

1. Make sure you are connected to your OT-2
2. Scroll down in the main connection page
3. Under **Advanced Settings,** you will see a **DOWNLOAD** button
4. Select the **DOWNLOAD** button
5. Follow the operating system specific on-screen directions to save the log files
   a. There will be 3 log files total but you will be prompted to save one at a time. View video.

9.6 Installing Linux commands on Windows

Most Windows computers do not automatically come installed with Linux based commands such as ssh or make. This article aims to give a simple option for installing these commands.

Instructions

Install chocolaty

To install, you can either use cmd.exe or powershell In either scenario you should make sure you are running as administrator. [Search and Right Click]
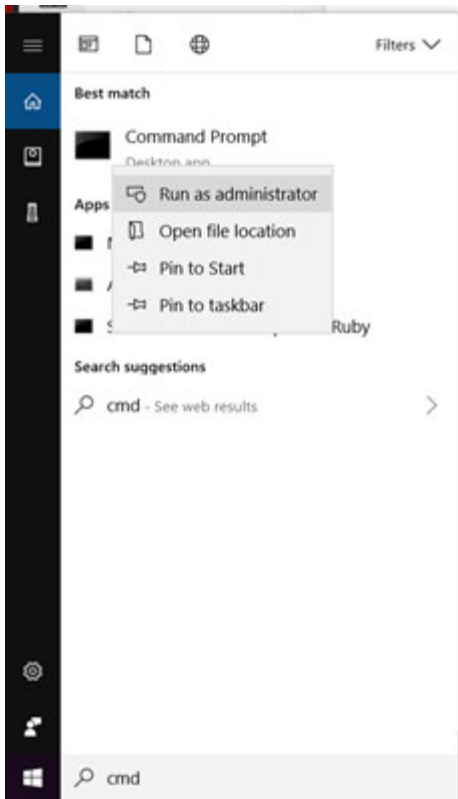
Make sure you are connected to a reliable internet.

For cmd.exe use the following command:

1@"%SystemRoot%\System32\WindowsPowerShell\v1.0\powershell.exe" -NoProfile -InputFormat None -ExecutionPolicy Bypass -Command "iex ((New-Object System.Net.WebClient).DownloadString('https://chocolatey.org/install.ps1'))" && SET "PATH=%PATH%;%ALLUSERSPROFILE%\chocolatey\bin"

For powershell use the following command:

1Set-ExecutionPolicy Bypass -Scope Process -Force; iex ((New-Object System.Net.WebClient).DownloadString('https://chocolatey.org/install.ps1'))
2

Type the following commands:

1choco install git.install
2choco install python --version=3.5.4
3choco install nodejs.install --version=8.9.4
4choco install yarn
5choco install make
6choco install openssh
7choco install winscp

You should now be able to use linux commands in your native terminal.

9.7 Movement, positioning, and calibration issues

Check OT-2 Calibration Health

You can make sure your OT-2's *positional calibration* is accurate by running a *Calibration Health Check.* If it isn't, the check will help you identify which parts need to be fixed.
Calibration Health Check guides you through checking the accuracy of your OT-2's:

- *Deck calibration*.
- *Pipette offset calibrations*, for your currently attached pipettes.
- *Tip length calibrations*, for the tips that you used to calibrate your attached pipettes' offsets.

It will *not* help you check the accuracy of:

- Pipette offset calibrations for pipettes that *aren't* currently attached.
- Tip length calibrations for tips *other than* the ones that you used to calibrate your attached pipettes' offsets.
- Any of your *labware calibrations*.
- Any of your custom *labware definitions*.

Before beginning

Required software versions

Make sure that both your OT-2 and Opentrons App are *updated* to at least **v4.0.0.**
Note: In earlier versions, positional calibration and the Calibration Health Check *worked much differently*.

Required materials

- Opentrons tips in Opentrons tip racks, for each attached pipette
- Calibration Block

Procedure

Step 1: Begin
In the **Robot** tab, go to your OT-2's home page.
In the **Robot Calibration** section, near **Calibration Health Check,** select **Check health** to open the Calibration Health Check.
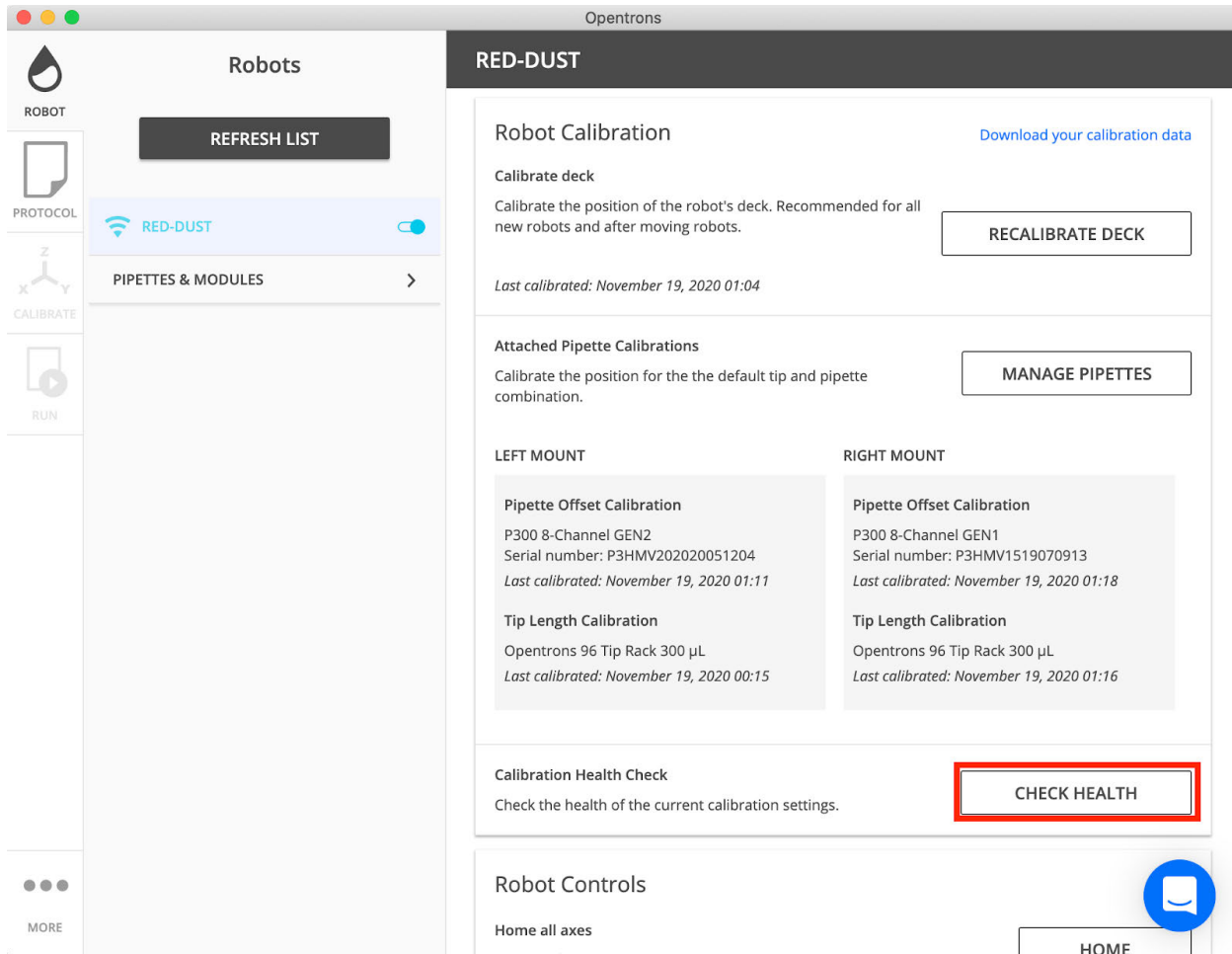
Can't find the "Check health" button? The button will be hidden if your OT-2 is missing a deck or pipette offset calibration. Look for "Calibration required" errors and resolve those first.

Step 2: Take measurements

Closely follow the Opentrons App's instructions. Take the measurements by manually jogging each pipette to positions designated by the Opentrons App. The Opentrons App will compare those measurements to your OT-2's current deck, tip length, and pipette offset calibrations.

Step 3: View the results and download the details

The Opentrons App will summarize which parts of your OT-2's positional calibration looked good and which parts looked bad. While you're here, download the **detailed JSON Calibration Check summary**. If you reach out to Opentrons Support for help, it will help us troubleshoot. You won't be able to download it after you exit the Calibration Health Check.

Note: For background on what each of these parts does and what the differences are between them, see *How positional calibration works on the OT-2*.

Next steps

If all the checks result in a good calibration
Congratulations: your OT-2 can accurately move both of its pipettes relative to the deck. As you continue using it, though, be on the lookout for other problems. The Calibration Health Check can't identify every kind of calibration problem.
If something still seems wrong even though all the checks passed, see if the problem is specifically mentioned in *Troubleshooting movement, positioning, and calibration problems*. Otherwise, reach out to Opentrons Support with the detailed JSON calibration check summary that you downloaded.

If the checks find problems:

1. If there's a problem with deck calibration, *recalibrate the deck*.

2. If there's a problem with tip length calibration, *recalibrate tip length*.
3. If there's a problem with pipette offset calibration, *recalibrate pipette offset*.

Doing things in this order minimizes the extra work that you'll have to do because of how the different parts of calibration *build on top of each other*.

After recalibrating everything, run the Calibration Health Check again to make sure the problems have been resolved.

Resolving a Bad Deck Calibration

If the following error occurs in the Opentrons App when connecting to the OT-2:

Bad deck calibration data detected! This robot is likely to experience a crash. Please perform a deck calibration prior to uploading a protocol.
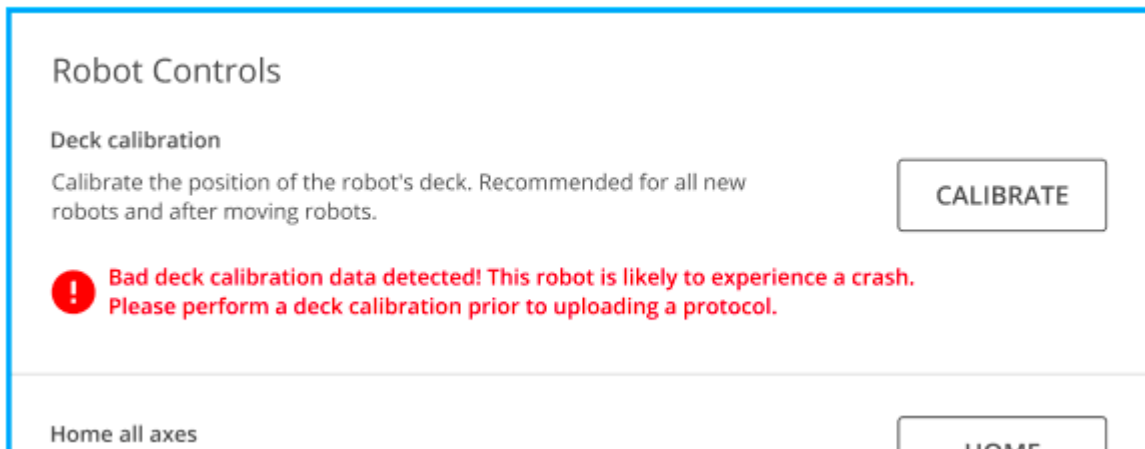
What this error means:

The OT-2 proactively detects some basic calibration problems, including:

- The deck has been calibrated too high or low.
- Deck calibration has never been performed.

Until the issue is resolved, the OT-2 will be unable to upload or run any protocols. This check helps avoid pipette collisions.

Resolve the problem

1. Update both the Opentrons App and the OT-2 robot software to the latest version. This is important, since older software versions have known bugs that make this problem difficult to resolve.
2. Calibrate the deck.

After completing the steps above, Opentrons recommends checking the OT-2's robot calibration, which can diagnose many more problems than this simple automatic check can.

9.8 Common Problems and Concerns

Same error occurs after deck recalibration

- Ensure the Opentrons App and OT-2 are both [updated](#) to the latest software version. Older software versions had [known bugs](#) that could make this problem difficult to resolve.
- Use *Opentrons* tips for the deck calibration. If non-Opentrons tips are used, the deck calibration will be inaccurate, and this error will persist.

If things still don't seem right after checking those things, please reach out to Opentrons Support. We'll help work out what's going on.

I didn't see this error message until I updated it. Did updating break something?

There was probably something wrong with the OT-2's calibration all along. When you updated your OT-2, it just learned how to automatically detect the problem.
The OT-2 might have appeared to work normally, before. The [interplay between the different parts of calibration](#) can hide problems. For example, if deck calibration was 10 cm too high, that might have been masked by a pipette calibration that was 10 cm too low.
These hidden problems tended to manifest in confusing ways later on—like collisions the first time you used a new pipette, or new labware.

Can I temporarily override this check, so I can run my protocol?

This check can't currently be overridden.
It's important to fix the underlying problem with the OT-2's calibration. But if this is not possible—for example, if your lab's standard operating procedures limit when you're allowed to—reach out to Opentrons Support. We can help downgrade your OT-2 to a previous software version that doesn't have this check.

9.9 How to correct attachment of two pipettes when one is off position

During a protocol run, when the pipettes move to labware, you might notice:

- One pipette goes to the correct place.
- The other pipette is up to 1-2 mm off, and that error is consistent throughout the protocol.

Why this happens

This is a known limitation in software versions earlier than 4.0.0.
On the OT-2, labware is only calibrated with one pipette. To move the *other* pipette to the same position, the OT-2 relies on knowing the exact offset between the two pipettes' tips. It measures this with pipette offset calibration. (See [How calibration works on the OT-2](#).)

Before software version 4.0.0, the OT-2 took that measurement by automatically tapping its pipettes against five *tip probe switches* hidden under the OT-2's trash bin. This could be inaccurate and imprecise. So, whichever pipette you didn't calibrate labware with would be slightly off-position when it moved to that labware during your protocol.

How to fix it

Update your Opentrons App and OT-2 to the latest software version. In v4.0.0, we improved how calibration works to make it more accurate, make it more precise, and fix problems like this.

9.10 OT-2 and Computer Connection Issues

Troubleshooting problems with USB connection

If your OT-2 isn't showing up in the OT-2 App even though it's connected to your computer by a USB cable, you're experiencing connection problems.

Make sure your OT-2 is turning on

Power-cycle your OT-2. (Use the power switch to turn it off, wait several seconds, and then turn it back on.)

As it restarts, look for the following:

1. The blue light on the front of the OT-2 should blink, then turn solid.
2. The pipettes should rise to their maximum height, if they are not already there, and make a whirring noise.

If those events don't happen within 5 minutes of powering on the robot, contact Opentrons Support

Try waiting after plugging your OT-2 into your computer

If you click the **Try again** or **Refresh list** button too quickly after plugging in the USB cable, your OT-2 might not show up in the Opentrons App. (This is a known issue that we plan to fix in a future software update.)

To work around this problem:

1. Power on the OT-2 and wait for the ring light to light up solid blue.
2. Plug your OT-2's USB cable into your computer.
3. Wait 60 seconds.
4. Then, click **Try again** or **Refresh list.**

Your OT-2 should then be available to connect. If it doesn't, continue with the following troubleshooting steps.

Troubleshoot based on computer operating system

Further troubleshooting steps depend on your operating system.

- Troubleshoot USB on Windows 10
- Troubleshoot USB on macOS
- Troubleshoot USB on Ubuntu

Troubleshooting problems with Wi-Fi connections

If you can't connect to a Wi-Fi network, we recommend the following steps:

1. Make sure your app and robot are fully up-to-date.
   a. Note: You can perform a manual update if needed.
2. If you are connecting to an open network, please make sure the network does not use a captive portal. The OT-2 does not support captive portal networks.
3. Double check your network credentials. The "Show Password" toggle can be useful to ensure you are typing any passwords in correctly.
4. If you are having trouble connecting to a WPA2 Enterprise network, please consult your IT department. Your IT department will know more than us about your particular network setup and credentials. They may need to whitelist your robot's MAC address or something similar.
5. If you're still unable to connect, connect our Support Team with the error message(s) you are receiving along with any details from your IT department (if applicable).

If you are able to connect your OT-2 to Wi-Fi but your app is unable to find your robot on your Wi-Fi network, please ensure that your network and firewall(s) are set up to allow traffic as described in the network requirements for the OT-2 Support article. If you are on an enterprise network, you may want to check in with your IT department.

Troubleshooting Windows blue screen errors
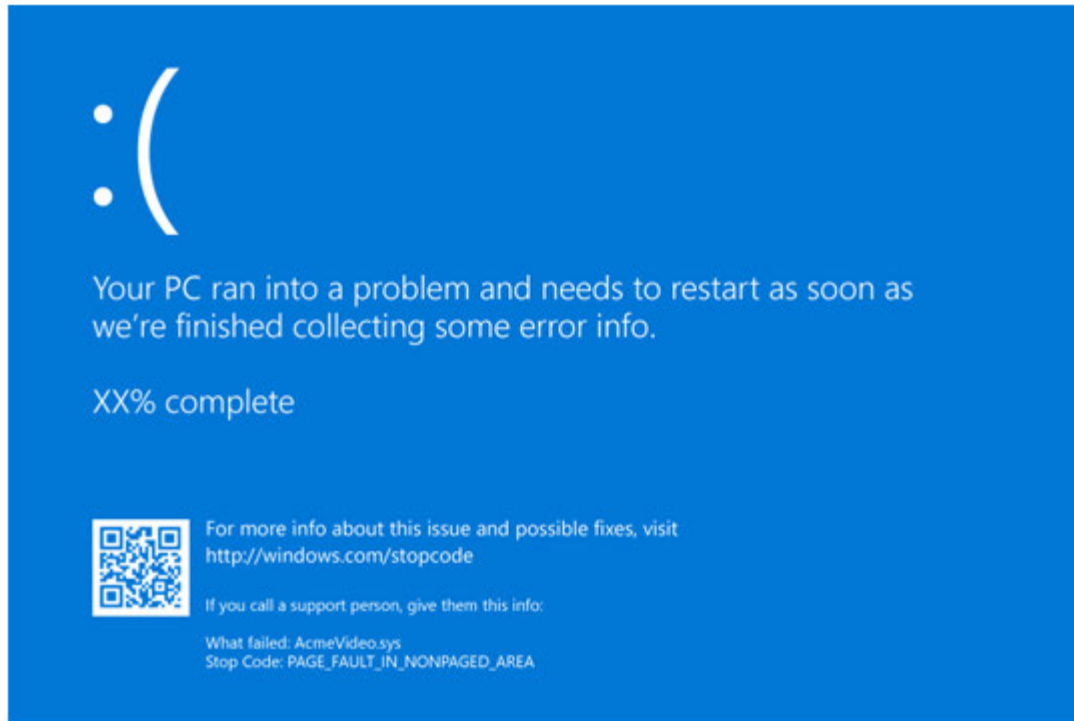
Windows blue screen (BSOD) errors have rarely been reported while using the OT-2. You might see them when uploading protocols or updating your OT-2's software.

Why do blue screens happen?
Although a blue screen may appear to be triggered by your OT-2, it actually originates from a problem somewhere else. Blue screens can happen when there's a problem:

- in Windows,
- in a device driver (a special program that helps run a hardware device),
- or, in antivirus software.

Generally, apps like the Opentrons App aren't capable of causing blue screen errors.

How do I fix the problem?

1. Ensure Windows is up to date
See the Microsoft article: Update Windows 10.

2. Update the Realtek USB FE Family Controller driver

We've found that the version of this driver that's installed automatically can be outdated and problematic.

Manually download and install [the latest driver from Realtek's website](#).

The download will be provided as a .zip file. Open it and run the **setup** program, which will guide you through installation.

3. Update your other device drivers

Most other device drivers can be updated through Windows Update.

Open **Start > Settings > Update & Security > Windows Update** and select **Check for updates.**

4. Check your antivirus

Customers have reported certain antivirus software causing blue screens when trying to use the OT-2.

Try temporarily disabling or uninstalling your antivirus software. If that resolves the problem, contact your antivirus vendor to report the bug with them.

5. Follow Microsoft's troubleshooting guide

See the Microsoft article: [Troubleshoot blue screen errors](#).

6. Try a different computer

If you're still seeing blue screen errors after trying all the above steps, you can still use your OT-2. A computer of a different make and model is unlikely to see the same error

9.11 Bridge network connections for the OT-2

This document presents a way to connect an OT2 connected to a computer with the USB/ethernet adapter to the internet by bridging the USB/ethernet adapter wired connection to the computer's internet connection.

IMPORTANT: On most operating systems, creating a bridge adapter that includes your internet connection will disconnect your computer from the internet while the bridge is active (removing the bridge will allow you to regain connectivity). Make sure this guide is saved in a location accessible without internet connectivity before following these steps.

Background

Under normal circumstances, connecting the OT2 to a computer using the USB interface creates a network connection between only the OT2 and the computer. Because the computer is not running the software that a router or switch would, this connection is known as a *link-local connection* and does not allow the OT2 to connect to the internet.

Windows, OSX, and Linux however have the capability to create something called a *bridge network adapter*, which connects in software two network adapters, so they seamlessly push traffic between them. Creating this network bridge between the OT2's USB/ethernet adapter and

the computer's network connection allows the OT2 to use the computer's connection to the internet and retrieve the online-only portion of Opentrons release 3.3 update.

Create Bridge for OSX

This guide is sourced from [Apple's knowledgebase](#).

Creating the bridge

- Choose Apple menu > System Preferences, then click Network.
- Click the Action pop-up menu, then choose Manage Virtual Interfaces.
- Click the Add button, choose New Bridge, then select the interfaces to include in the bridge. The interfaces should be the way your Mac connects to the internet (probably the adapter named Wi-Fi) and the OT2's USB/ethernet adapter, which will be called something like USB 10/100 LAN. You can tell for sure which adapter is the OT2's by noticing which adapter only has a green dot when the USB cable is plugged in. Name the bridge something you will remember, like Opentrons Bridge.
- Click Create, then click Done.
- To apply the changes, click Apply. Once you do this, your OT2 will be connected to the internet, but your Mac will no longer have internet connectivity.

Removing the Bridge

- Choose Apple menu > System Preferences, then click Network.
- Click the Action pop-up menu, then choose Manage Virtual Interfaces.
- Select the bridge you created above, and click the Delete button
- Click Done
- Select the bridge you created above in the list on the left of Network Manager and click the Delete button. You can select either option in the confirmation window.
- On the bottom left of Network Manager, click the Add button and select Wi-Fi from the dropdown menu. Click OK.
- On the bottom left of Network Manager, click the Add button and select USB 10/100 LAN from the drop down menu. Click OK.
- On the bottom right of Network Manager, click Apply. The Wi-Fi and USB 10/100 LAN adapters will reconnect.

Windows 7, 8, and 10

This guide is sourced from [Windows Central](#) (not affiliated with Microsoft).
If you are an advanced user, you can follow the steps in the linked guide to set up and save a static IP address. This may allow your computer to remain connected to the internet while the bridge adapter is active; however, if you have an IT department you should contact them about the

settings to use and depending on their configuration, the steps outlined in the linked guide may not be possible.

Creating the Bridge

- Open the Network Connections control panel pane:
- On Windows 7, 8, and 10 navigate through Control Panel to Network Connections
- On Windows, 7, 8, and 10 in the search bar type "Network Status" and select "Change Adapter Settings"
- On Windows 10, press Windows+X and select Network Connections, or (on other Windows versions) navigating through Control Panel.
- Select both the network adapter that connects to the internet and the OT2's USB/ethernet adapter. The OT2's network adapter should look something like this: The name may be different, but the bottom line should read something like Realtek USB FE Family Controller. To be certain you have the right one, disconnect and reconnect the OT2's USB cable and find the adapter that changes.
- Right-click the selection and select Bridge Network Connections.
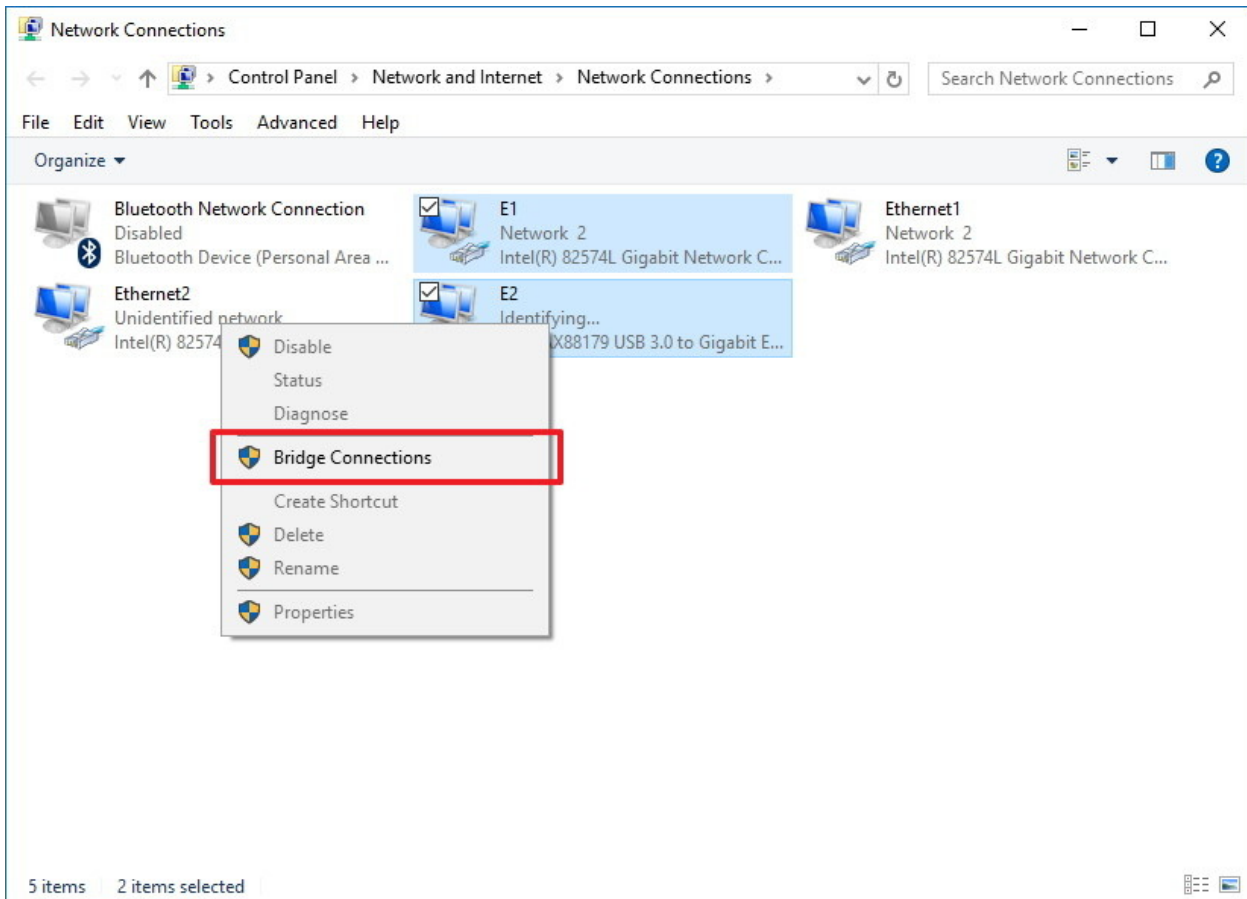
Removing the Bridge

- Open the Network Connections control panel pane by (on Windows 10) pressing Windows+X and selecting Network Connections, or (on other Windows versions) navigating through Control Panel.
- Select the two adapters that are in the bridge you created above (they should say enabled, bridged in the second line of text beside the icon), right click them, and select Remove from bridge.
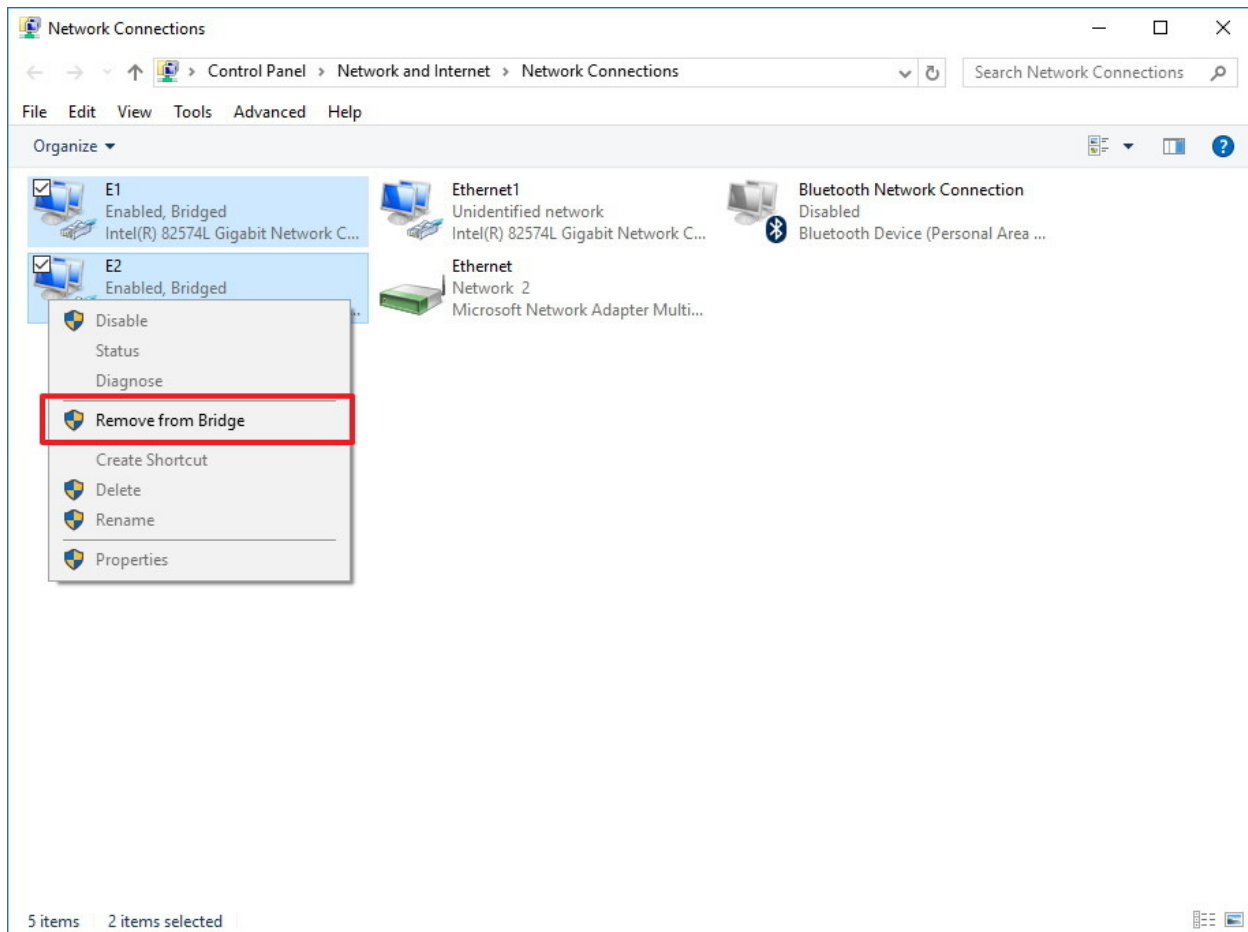
Linux

Network administration in Linux can vary widely between distros. This guide describes how to set up a network bridge in an Ubuntu desktop liquid handler; for other distributions or headless servers, contact us, your IT department, or set up your own bridge connection between the OT2's USB/ethernet adapter and your internet connection.

This guide is sourced from ask.xmodulu.com. It contains instructions for configuring a bridge using nm-connection-editor's GUI. If your system does not use NetworkManager or you do not

want to use the GUI, check the [Ubuntu community help wiki](#) (or your preferred distro's documentation) which contains guides to other methods of making the bridge.

Creating the Bridge

- Open a terminal and run ifconfig to find the interface name of the OT2's USB/ethernet adapter. If you can't find it, try disconnecting the cable, running ifconfig, reconnecting it, running ifconfig again, and noting down the interface that only appears when the cable is plugged in. Note down the interface name ethX and the MAC address.
- Open nm-connection-editor
- Click the add button in the top left
- Select the Bridge connection type from the dropdown
- Add a connection
- Select the Opentrons USB/ethernet adapter that you noted down above.
- Add another connection and this time select the adapter that connects your computer to the internet.
- For both connections, allow any user to connect to them and enable when the computer turns on
- Go to the IPv4 configuration tab of the bridge adapter settings and configure it in the same way that your internet connection is configured - probably DHCP.
- In Network Connections, delete the old adapter that connected you to the internet (or disable it). If you delete it, note down the way it was configured.

Removing the Bridge

- Open nm-connection-editor
- Edit the bridge connection created above and remove the two connections contained in it
- If you deleted the connection that connects your computer to the internet, recreate it with the settings you noted down above. If you disabled it, re-enable it now.
- Delete the bridge connection

9.12 Connect the OT-2 to a router

The electronics of the OT-2 have been specifically designed to be connectable via Ethernet. This is possible due to the removal of the internal USB-to-Ethernet converter which is present in the OT-2 but not the OT-2. This means it is easier to connect your OT-2 to a router due to direct access to the Ethernet port.

Materials needed:

- Router, ideally with a [DHCP server](#) (which most have)
  - **Note**: No specific routers are recommended because most should work.
- Ethernet cable

1. [Locate the Ethernet port on your OT-2 and plug in your Ethernet cable](#)

2. Plug in the other end of the Ethernet cable to your router

Follow the instructions provided by your router for establishing a connection with your computer. See below for a couple examples of how your network setup might be configured with the OT-2 connected.

Router only:

Internet

Firewall

Router

OT-2R
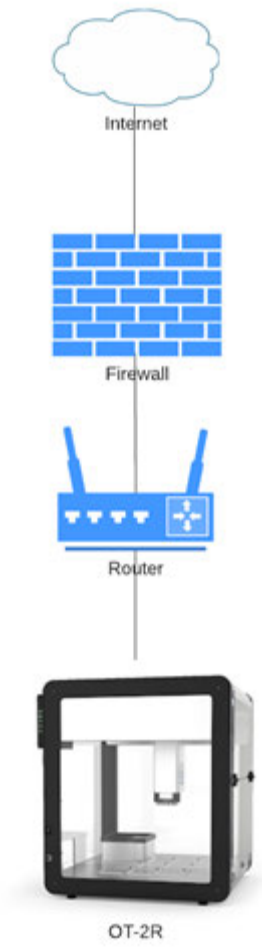
Router → server:
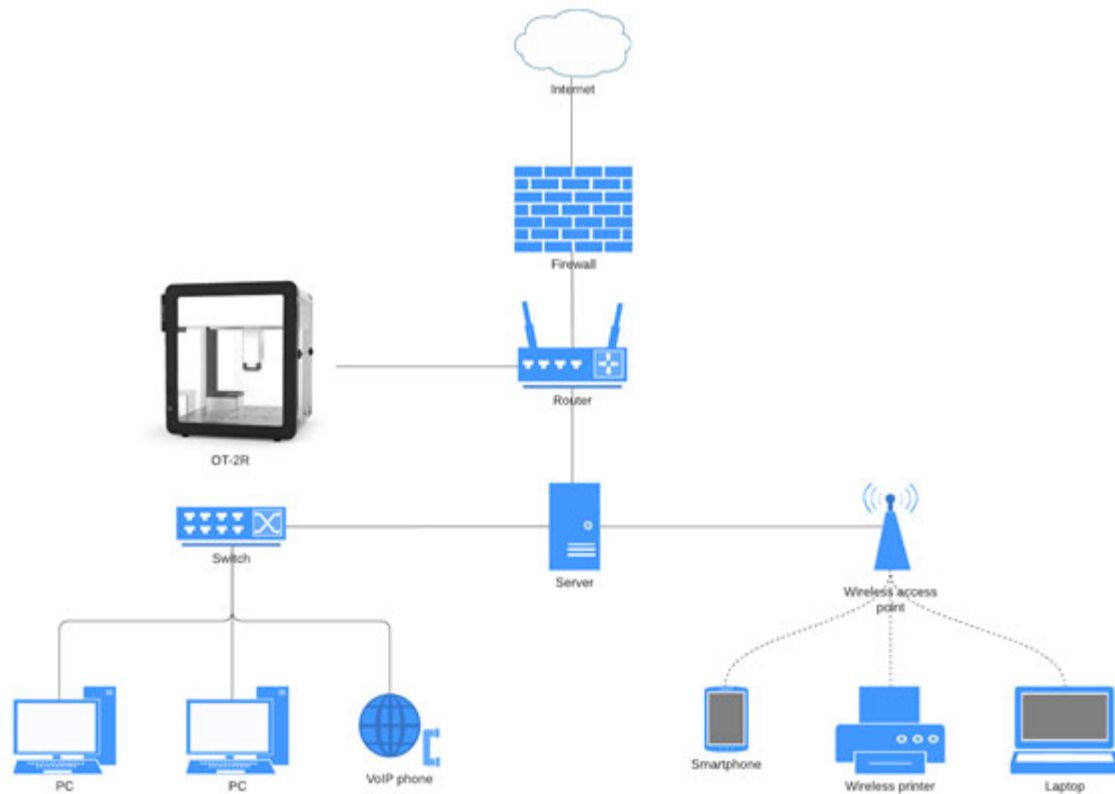
Troubleshooting internal network requirements for the OT-2

- Ensure that you are able to meet the network requirements for the OT-2. Your company might have its own specific network requirements to be able to connect to your robot.
    - In this case, we recommend reaching out to your IT department as they might need to help you bypass a firewall or modify specific settings on your computer.
- See our article on how to read the OT-2 status lights. These lights provide diagnostic feedback for troubleshooting.
- You may or may not need to use a separate modem with your router because they might be part of the same physical device.

9.13 Opentrons App shows no steps

Are you seeing this error when uploading a protocol to your robot? The basic [structure of protocols changed in APIv2](#) and if your robot software is too old, it will mistakenly see these new protocols as empty.

Your protocol could not be opened.

This protocol has no steps in it - there's nothing for your robot to do! Your protocol needs at least one aspirate/dispense to import properly

It looks like there might be a problem with your protocol file. Please check your file for errors. If you need help, contact support and provide them with your protocol file and the error message above.

If you didn't write the protocol yourself:
Don't worry, the protocol *probably* has steps in it, and updating the robot software so it can understand APIv2 protocols should resolve the issue.

1. [Update your App and the Robot](#) to the most current version of the Opentrons software.
2. Verify that the robot is running a server version greater than 3.14.0

Information

Robot name:	Server version:
OT2CEP20190528A04	3.17.1
Firmware version:	Max Protocol API Version:
v1.0.8-1f0a3d7	2.4

If the error persists after updating your robot, contact Opentrons Support and send us the protocol file causing the error (and tell us where you found it!).

If you wrote the protocol yourself:
In addition to ensuring your robot software is up to date, you should double check your protocol for other syntax errors. Sometimes an unexpected indent or another syntax error can cause the

robot to miss the required run(protocol) function or metadata variable. See the API documentation for [more information on the required syntax for APIv2 protocols](#).
The API also requires you to have an executable step in your protocol to run on the robot. Any command that moves the pipettes will satisfy this requirement. If the robot still thinks the protocol is empty even though you are able to successfully simulate it, try including a protocol.comment("message") in your code.

9.14 Pipette connection issues

If you see a message like this, your OT-2 is having trouble communicating with the pipette that you just attached.
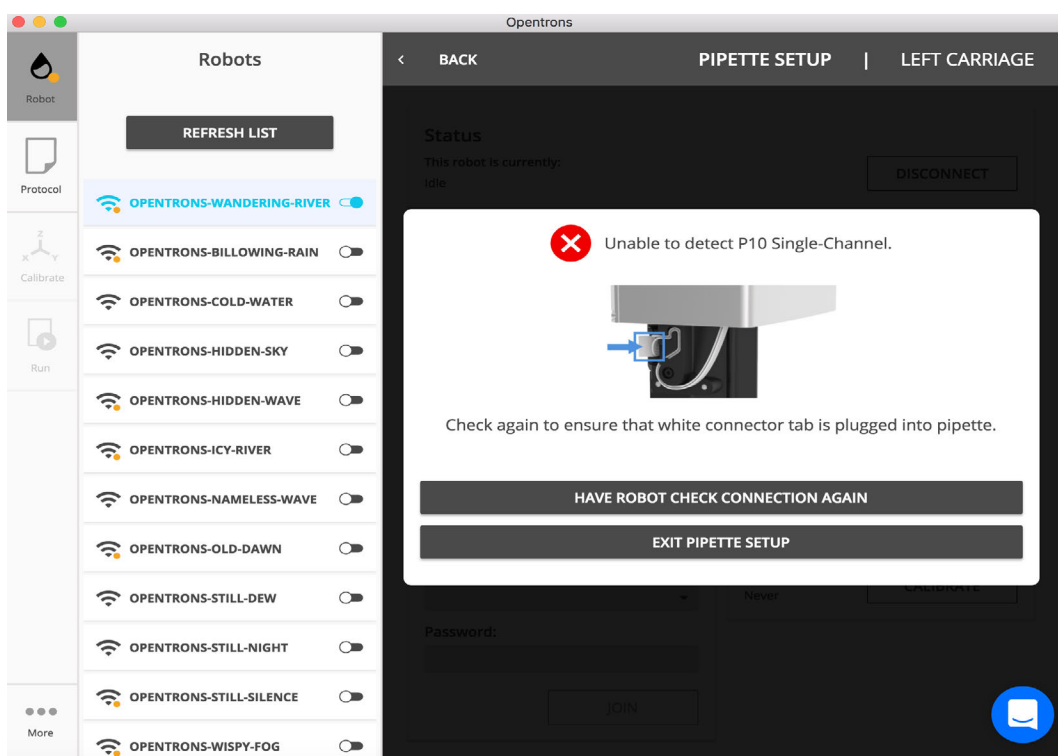
First, try unplugging the white connector tab from the pipette and re-inserting it. Ensure that the connector is plugged in securely. Then, click **Have robot check connection again.**
If the pipette still cannot be detected, there may be a problem with the pipette, or with one of the OT-2's cables. To narrow things down:

1. See if the pipette can be detected in the other mount, leaving the original mount empty.
2. See if a different pipette can be detected in the same mount.

9.15 Reflashing the OT-2's SD card

Normally, to update an OT-2, the Opentrons App uploads a small update package to it while it's running. The OT-2 opens the package and updates itself, carefully preserving things like calibration files and anything that you manually saved on the filesystem. You may need to reflash the SD card on the OT-2 if instructed to do so by a member of the Support team.

1. Get the URL for the current robot OS
2. Reflash the SD card
3. Next steps

Get the URL for the current robot OS

1. Go to Opentrons App on our website. To get the most current version of **ot2-fullimage.zip**, navigate to the section **Robot Operating System Update** and copy the URL for that link (where it says "here" shown below).
   a. **Note**: The URL hash changes every software release, so we are unable to provide a static link at this time.

2. Once the link address has been copied, paste it into your browser and change the file name at the end of the URL to **ot2-fullimage.zip**
   a. After changing the end of the URL, it should look similar to: https://opentrons-buildroot-ci.s3.amazonaws.com/719186c6-3b06-4a7c-95c6-60d882de8167/opentrons-buildroot/ot2-fullimage.zip
3. Select **Enter** to download the .zip file and continue reading.

Reflash the SD card

1. You will need to [remove the SD card from the OT-2](#).
2. If you have not done so already, download the current robot OS image following the [instructions above](#).
3. Download and install [balenaEtcher](#).
4. Follow the instructions for this tool to etch the new robot OS onto the SD card.
5. Re-insert the SD card into the OT-2 following the instructions for [removing the SD card](#), but in reverse order.
6. Power on your OT-2 and wait for the robot to initialize and boot. **This step could take up to 10 minutes.**

Next steps

1. If the new SD card image is for a version older than v4.0.0, you will need to [perform a factory calibration on the OT-2](#).
2. Set the OT-2's [serial number](#) by modifying and running the **set_serial_number.py** script below.
   a. To do this, you can either 1) download the .py file **here** and proceed to step b or 2) copy and paste the following script into a [source-code editor](#) of your choice.

```
# This script sets an OT-2's serial number.  It's meant to be used after you
# replace or reflash an OT-2's SD card.  This script is based on some of the factory
# provisioning scripts.
#
# Change the following to your OT-2's serial number.
# For example, in the script below: NEW_SERIAL_NUMBER = "<your serial number goes here>"


from opentrons import protocol_api
import os
metadata = {"apiLevel": "2.8"}

def run(ctx: protocol_api.ProtocolContext):

    #Please replace what is between the quotation marks!!

    NEW_SERIAL_NUMBER = "set_your_serial_number_here"
```

```
ctx.comment(f"Setting serial number to {NEW_SERIAL_NUMBER}.")

if not ctx.is_simulating():
    with open("/var/serial", "w") as serial_number_file:
        serial_number_file.write(NEW_SERIAL_NUMBER + "\n")
    with open("/etc/liquid handler-info", "w") as serial_number_file:

serial_number_file.write(f"DEPLOYMENT=production\nPRETTY_HOSTNAME={NEW_SERIAL_NUMBER}\n")
    with open("/etc/hostname", "w") as serial_number_file:
        serial_number_file.write(NEW_SERIAL_NUMBER + "\n")

    os.sync()


    ctx.comment("Done.")
```

    b. ❗**Important: Before you can use the script, you will need to modify it by using a [source-code editor](#)** to include your OT-2's serial number where the script's instructions indicate and save it as a .py file.

    c. Finally, you need to run the script on your OT-2 using the Opentrons App. To do this, open the Opentrons App. After connecting to your OT-2, go to the **Protocol** tab and [upload](#) the .py file you modified and saved. Navigate to the **Run** tab and select **Start run**.

9.16 Replacing the OT-2's internal USB-to-Ethernet adapter

If you're working with Opentrons Support to troubleshoot certain connection problems, we might ask you to replace your OT-2's internal USB-to-Ethernet adapter.

Required materials

- 2.5 mm hex driver (included with your OT-2).
- Replacement USB-to-Ethernet adapter (from Opentrons Support).

Before beginning

- Turn off your OT-2.

- Unplug your OT-2's power cable.

Disconnect the Ethernet cable

To give yourself enough slack to pull out the USB-to-Ethernet adapter, you'll first need to unplug an internal Ethernet cable.

Follow the steps in Accessing the OT-2's rear electronics.

You should see an Ethernet cable plugged into a Raspberry Pi circuit board:
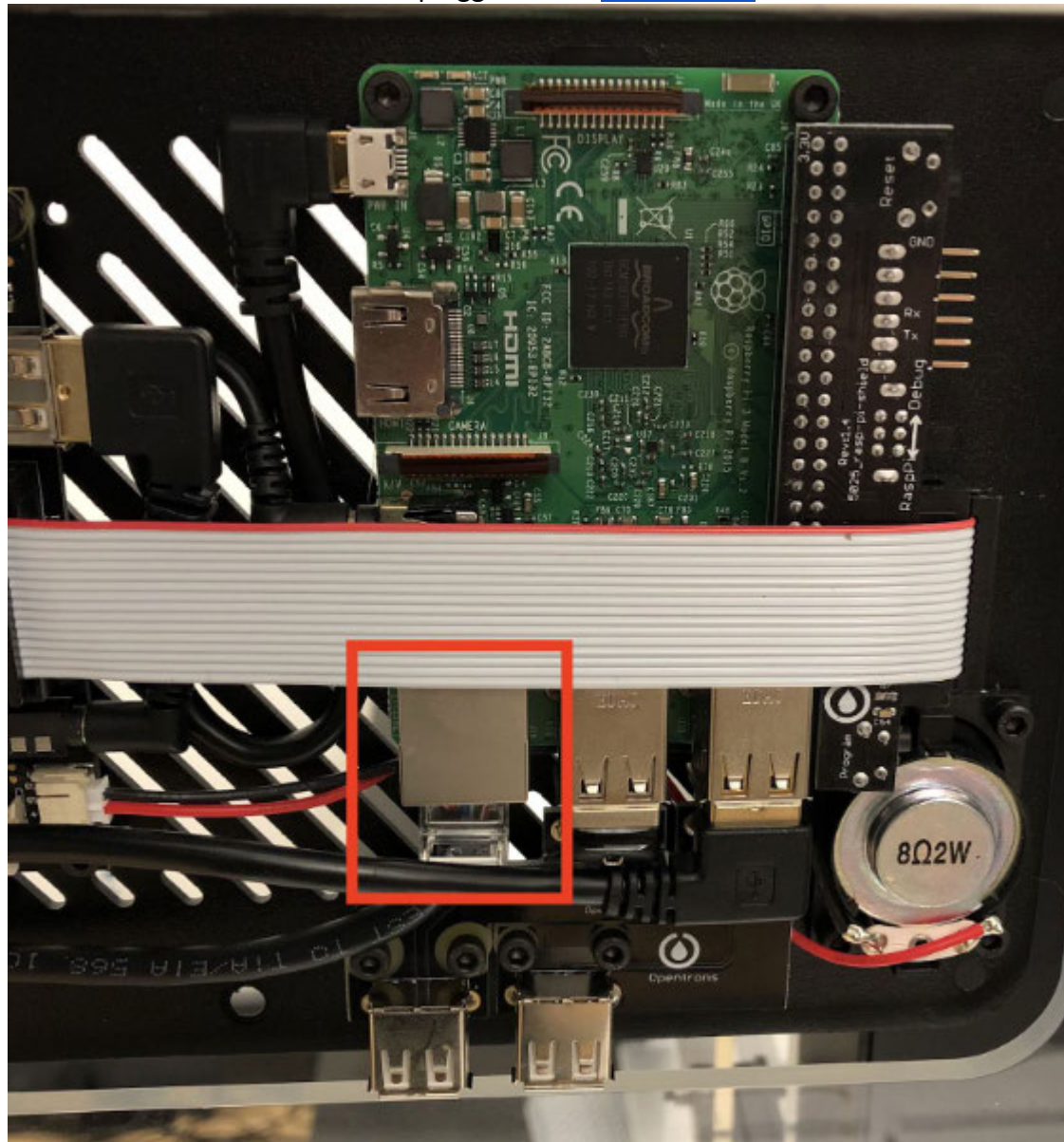
A tab on the back of the connector locks it in place. To unplug the connector, you will need to squeeze that tab.

On the back of the robot, there is a hole. Stick a hex driver through the hole to squeeze the tab on the connector. While doing that, use your other hand to pull the cable down and out.
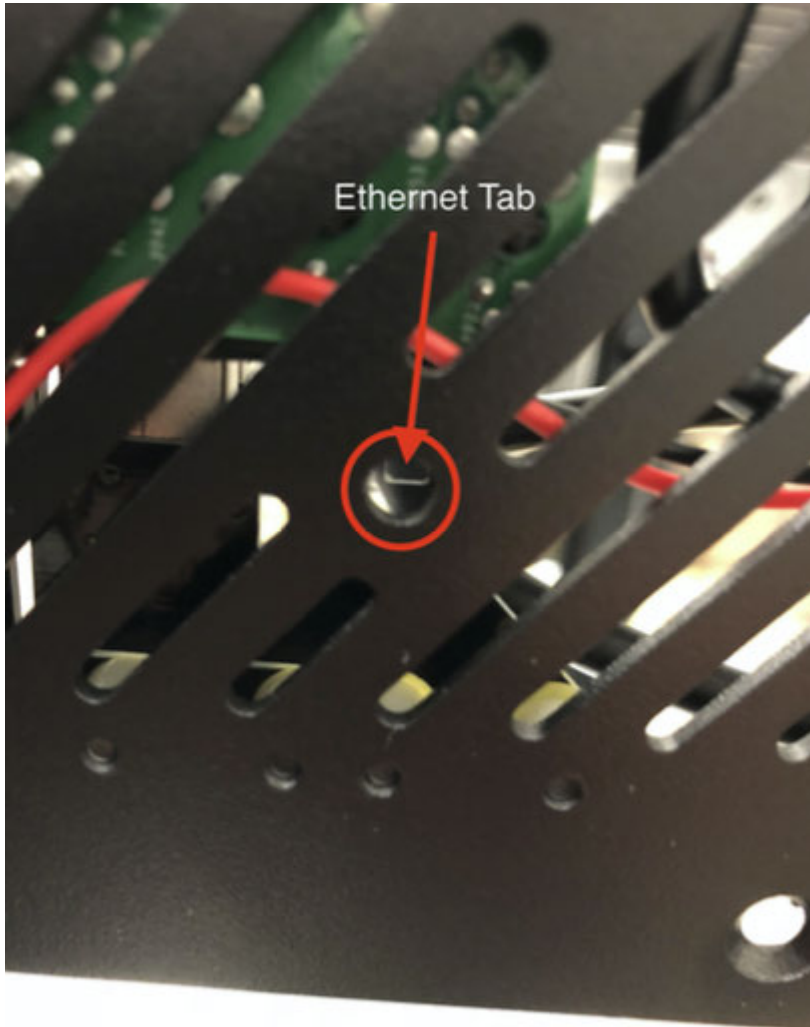
After unplugging it, leave that end of the Ethernet cable hanging loose.

Remove the OT-2 side panel

To reach the USB-to-Ethernet adapter itself, you'll need to remove the OT-2's left side panel.

On the left side of the robot, towards the back, unscrew the 2 hex screws.

Caution: The screws might be tight. Be careful not to strip them.

With those screws removed, you can bend the panel back slightly. This exposes some electronics, including the internal USB-to-Ethernet adapter.

Pictured below: the side panel of the OT-2 partially removed, with the USB end of the adapter unplugged from the robot (the adapter will be plugged in when opening the panel).

Tip: Optionally, to give yourself more room, you can remove the side panel entirely instead of just bending it back. Unscrew the 2 remaining side panel screws, towards the front of the robot. (They might be difficult to unscrew.)

Accessing the USB-to-Ethernet adapter

First, disconnect the USB-to-Ethernet adapter from the USB port in the side panel of the OT-2. An unrelated connector, with red and black wires, is in the way of accessing the USB-to-Ethernet adapter. You'll need to unplug it to get it out of the way.

This can be a little tricky. Hold the latch on the right open with one finger. Lift from the other side (circled) with another finger.

Note: You can use a screwdriver to help open the latch.

Once unplugged, it should look like this:

Pull the white USB-to-Ethernet adapter down, just far enough to let you unplug the Ethernet cable from it.

Caution: Don't pull the USB-to-Ethernet adapter too far. The other end of the Ethernet cable, which you unplugged from the Raspberry Pi earlier, will follow it. If it slips down into the side of the OT-2, it might get stuck there.

Finally, disconnect the USB-to-Ethernet adapter and replace it with the new one.

Reinstall the USB-to-Ethernet adapter

1. Move the USB-to-Ethernet adapter back up into the side of the OT-2.
2. Reconnect the connector with the red and black wires.
3. Connect the other end of the Ethernet cable back to the Raspberry Pi.
4. Reconnect the USB-to-Ethernet adapter to the USB port in the OT-2's side panel.
5. After making sure that the replacement USB-to-Ethernet adapter works, replace all the panels and covers.

9.17 Replacing the OT-2 SD card

Materials needed:

- Allen key set provided with your robot (specifically, the 2.5mm hex driver)
- Tweezers
- New Micro SD card
- Step stool (if your robot is placed on a tall bench)

SD Card Removal Procedure

1. Power off your robot and disconnect from any cables.
2. Turn your robot around so that you can access the back electronics enclosure panel.
3. Using your 2.5mm hex driver, remove the 4 screws which fasten the top acrylic panel and remove the panel by sliding and lifting.
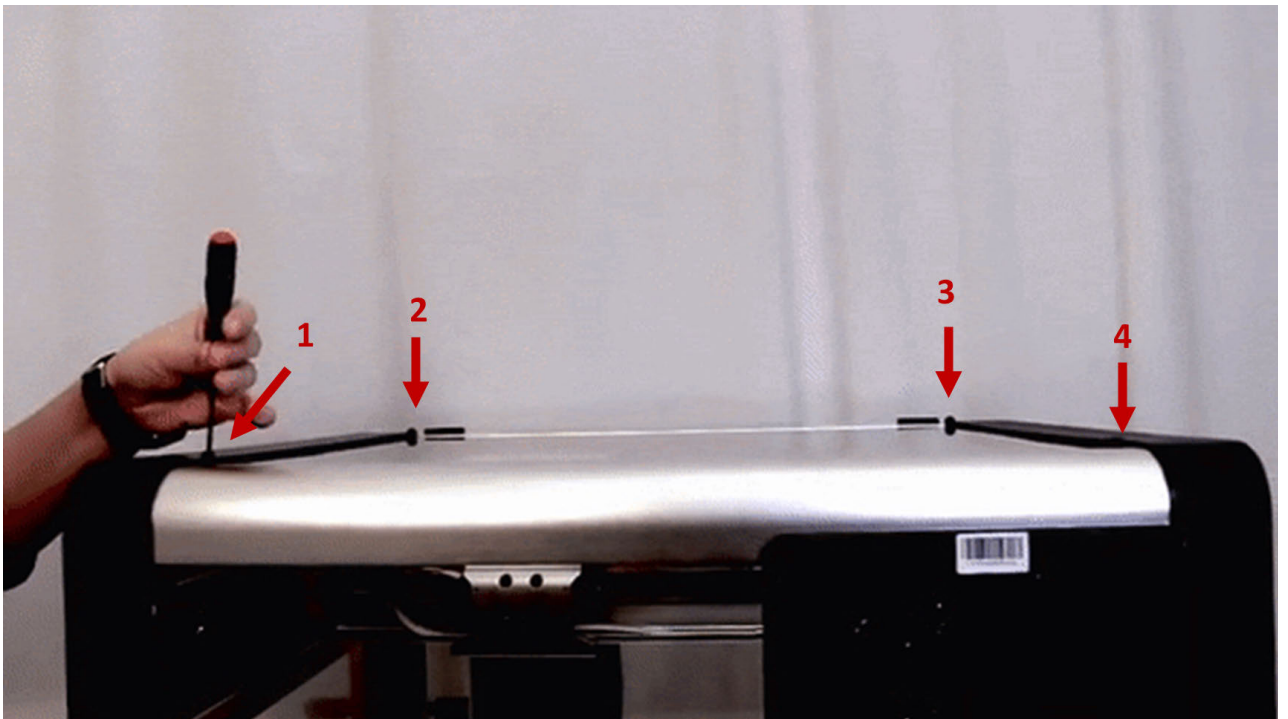
Remove 4 screws.
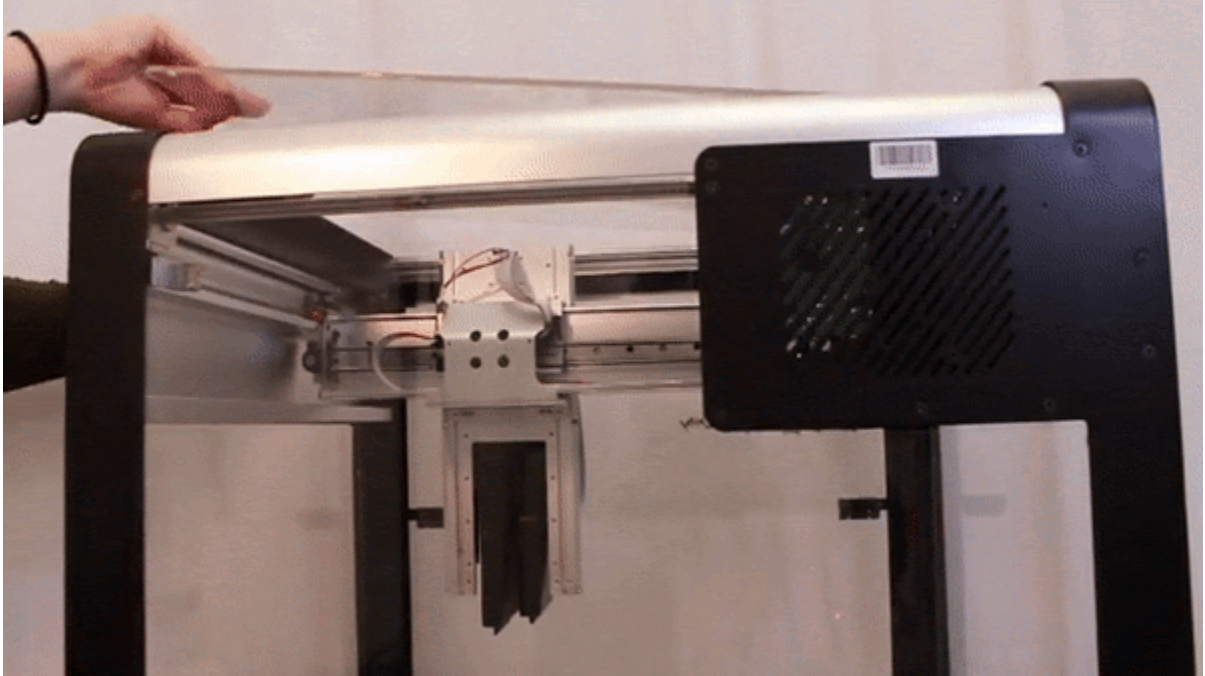
Remove the panel by lifting and sliding.

4. Now with your 2.5mm hex driver, remove the following screws to loosen the electronics enclosure cover. Once you reach the last two screws, place your hand on the top of the robot to remove the casing.
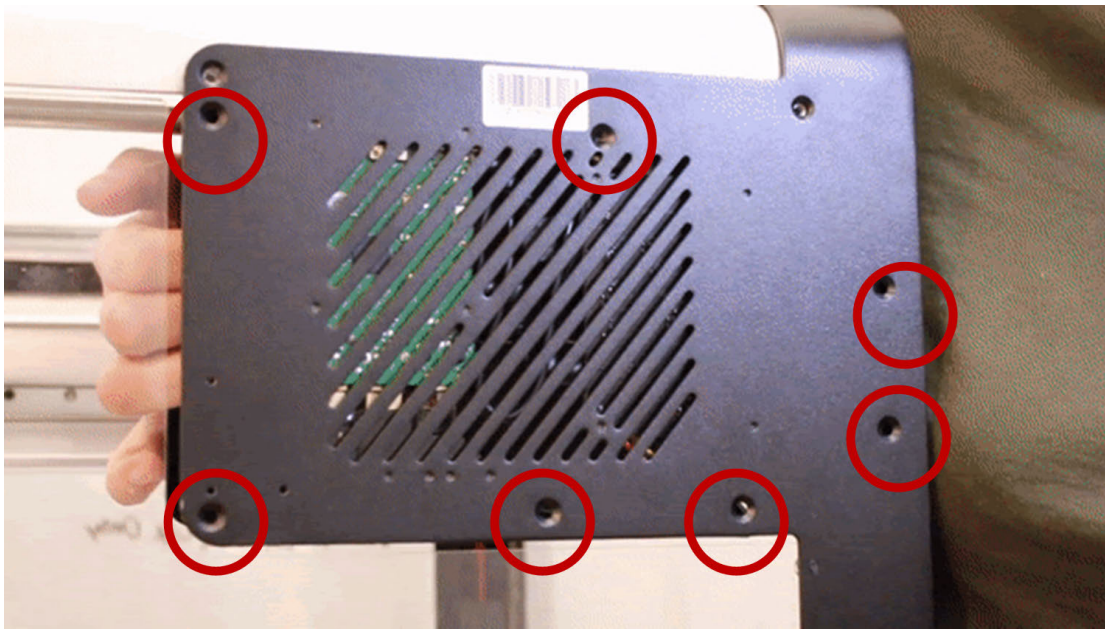
Screws off.
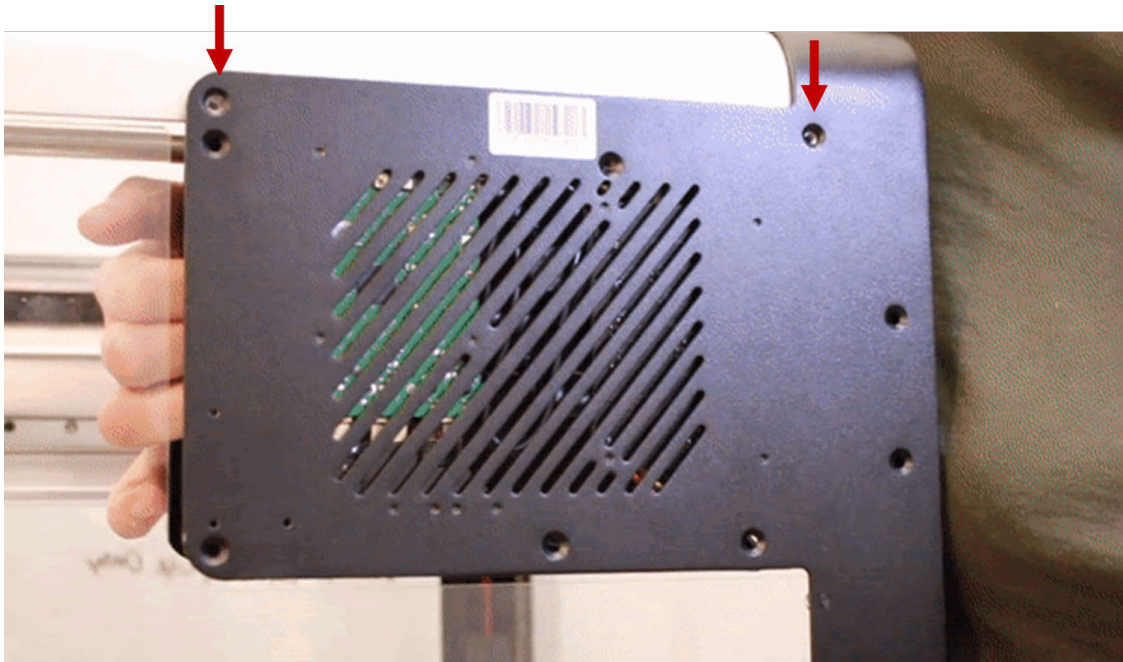
Leave these screws on.
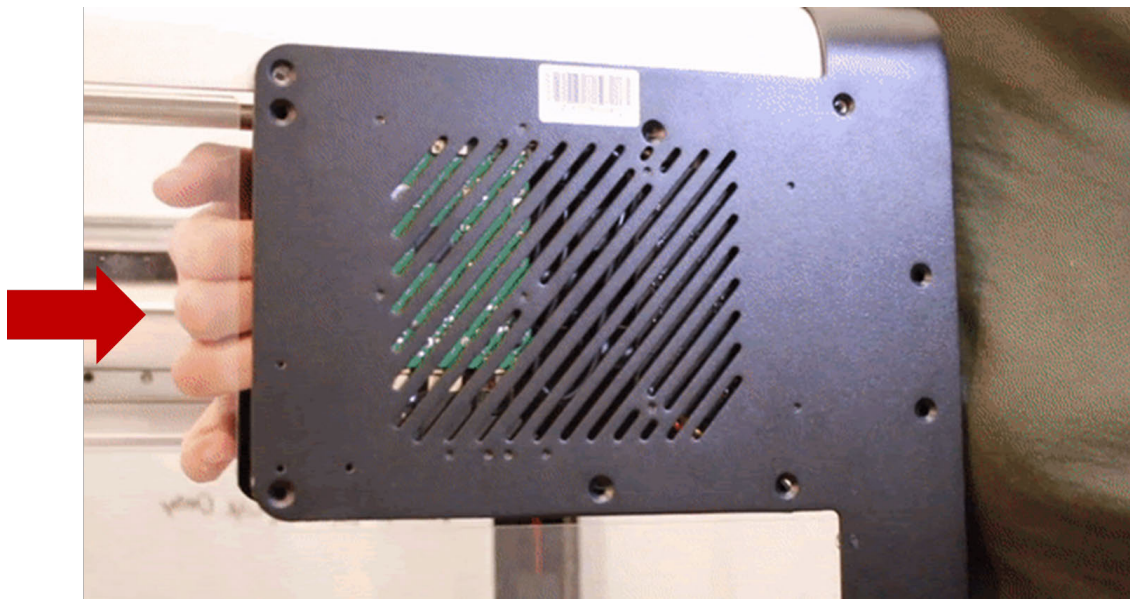
Hand here to remove casing.
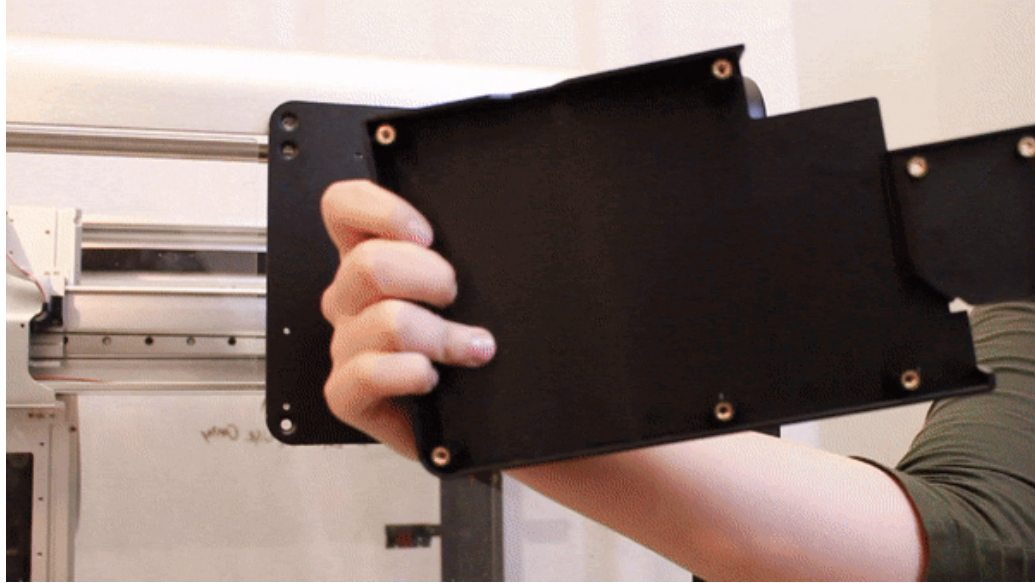
Casing removed.

5. With one hand, wedge the electronics enclosure open to access the Raspberry Pi (green circuit board on left) and notice the Micro SD card at the top.
6. With your other hand, take your tweezers and remove the Micro SD card.
   a. **Note**: Notice the orientation of the SD card for when you replace it to ensure it is properly inserted.

Orient tweezer.

MicroSD card out.

7. Below shows an inner view of removing the SD card:
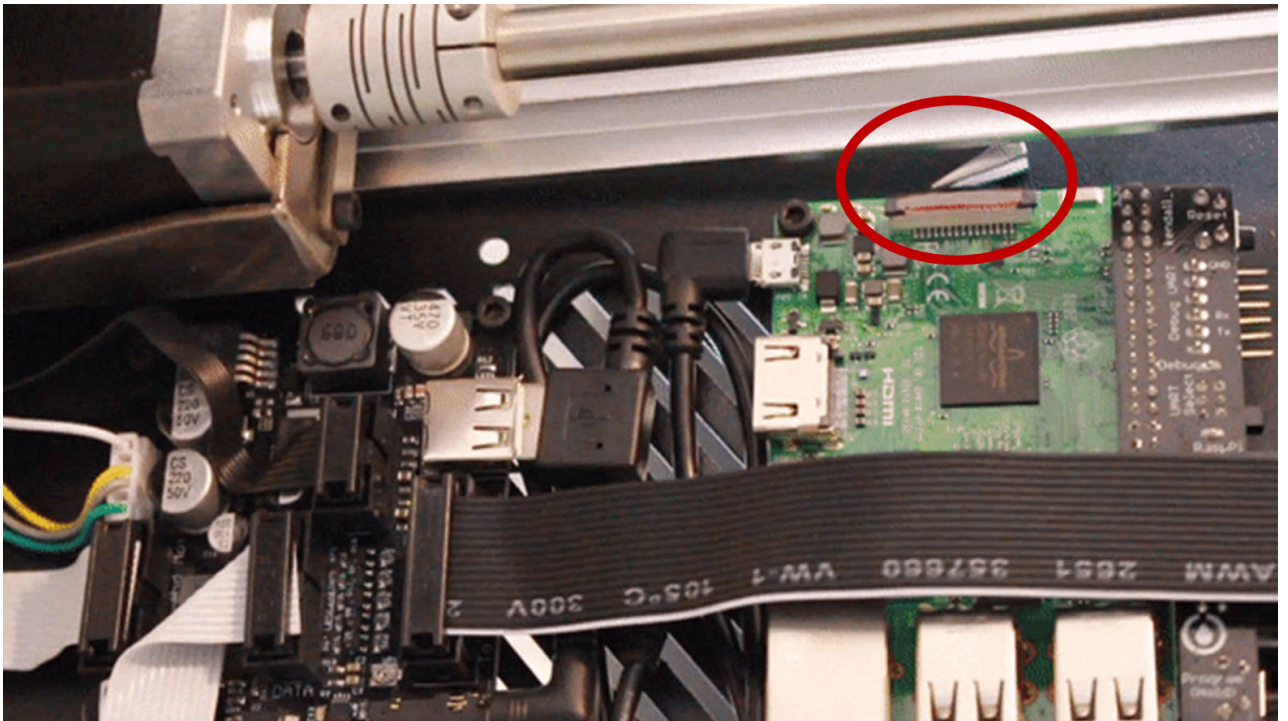
Tweezer reaching for MicroSD card.
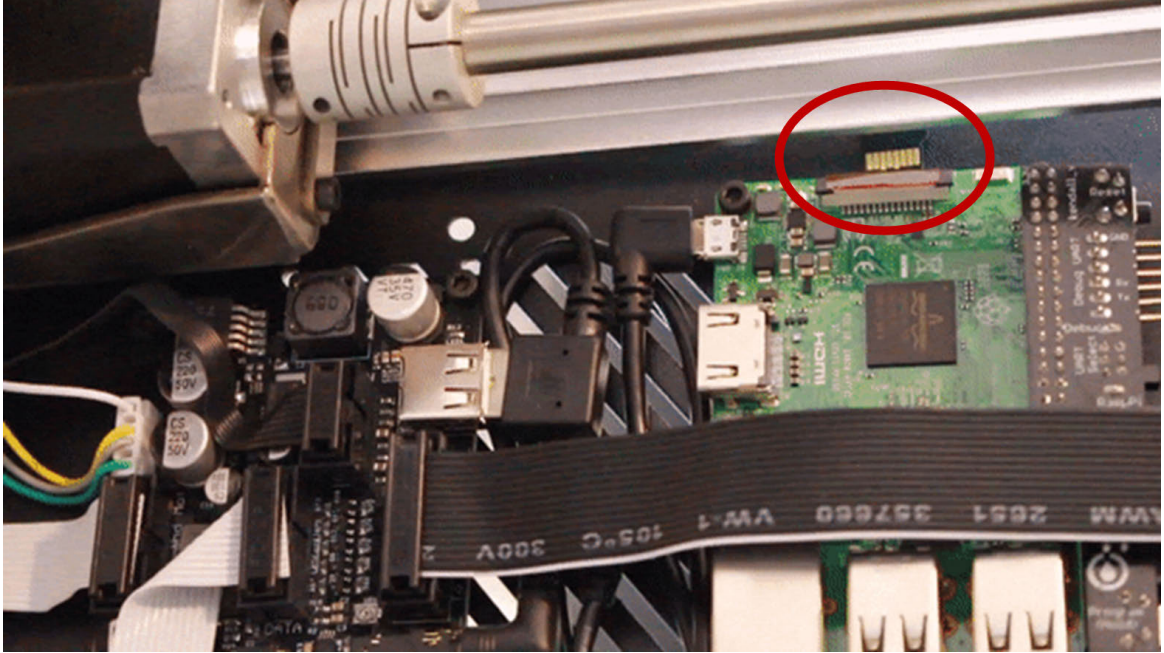
MicroSD card being pulled out.

8. Insert the new Micro SD card into the Raspberry Pi.
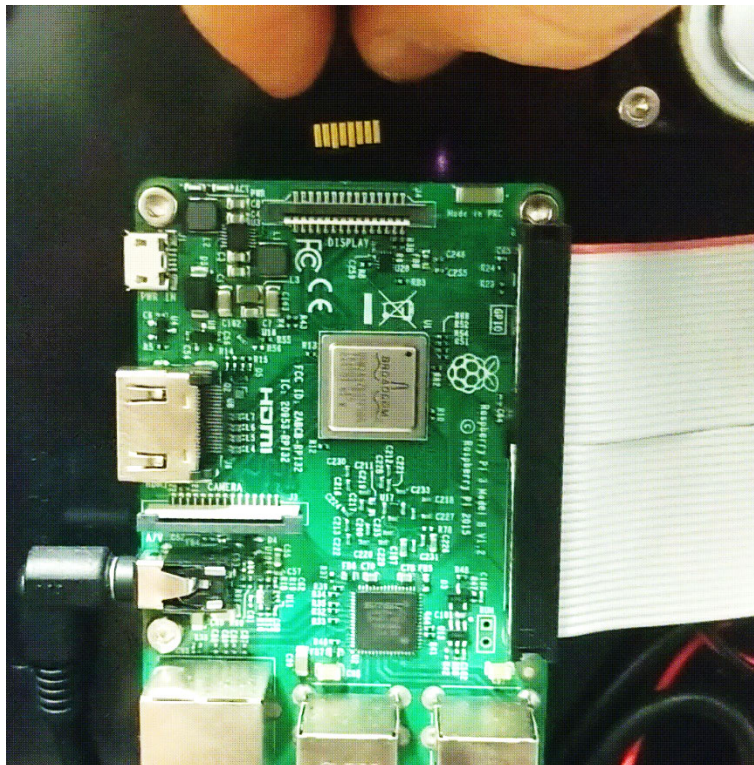
Re-fasten the electronics enclosure cover

Re-fasten the electronics enclosure cover and top acrylic. Power on the robot and it should automatically home. You will now need to [factory reset](#) your robot's deck.
Materials needed:

- T10 Torx (star) driver provided with your robot
- Tweezers
- New Micro SD card
- Step stool (if your robot is placed on a tall bench)

Re-fasten the electronics enclosure cover procedure

1. Power off your robot and disconnect from any cables.
2. Turn your robot around so that you can access the back electronics enclosure panel.
3. Now with your T10 Torx driver, remove the four corner screws shown below to open up the electronics enclosure cover.

1. With the panel removed, you can now access the Raspberry Pi (green circuit board on left). Notice where the Micro SD card goes: It slides into the top of the Raspberry Pi.
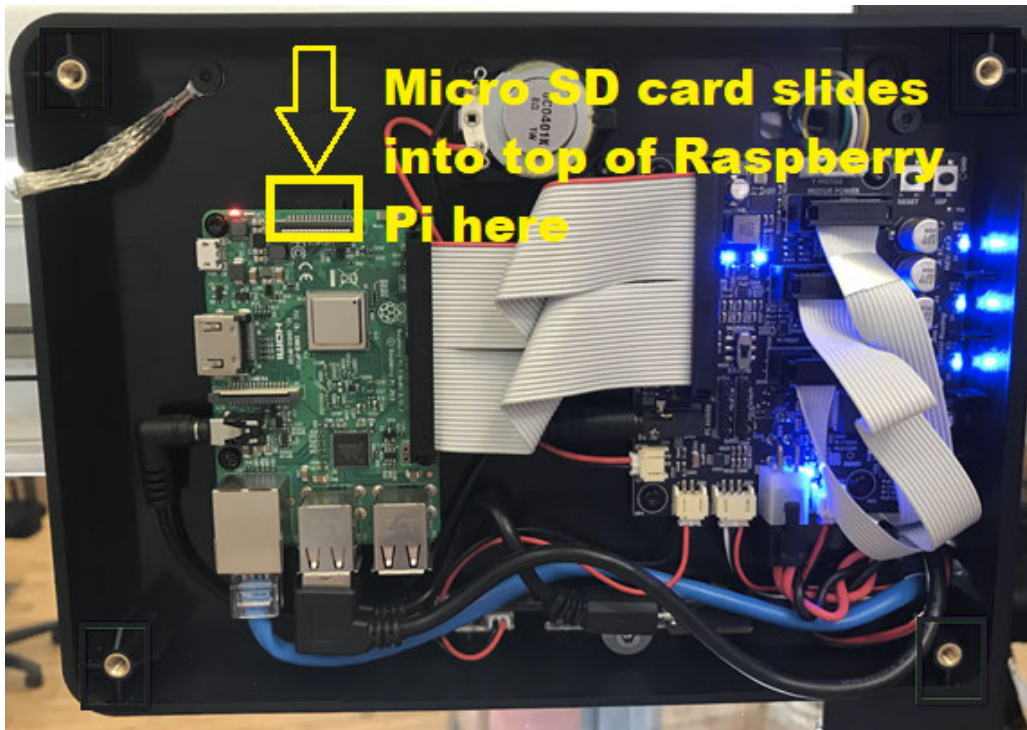
Update: The gray grounding strap is now located in the upper right (previously upper left as shown in this image).

Important: Before putting the metal panel back on, notice that the inside of the panel is partially unpainted. This is to establish a solid electrical connection from the panel to the gray grounding strap cable (located in the upper right within the enclosure). When putting the panel back on, it is important to **ensure that the gray grounding strap cable is sandwiched between two pieces of metal** when it is fastened.

Re-fasten the electronics enclosure while ensuring that the gray grounding strap cable is sandwiched between two pieces of metal. Power on the robot and it should automatically home. You will now need to factory reset your robot's deck.

9.18 Replace the OT-2 Tip Probe Switches

Overview:
The tip probe switch is an essential component to calibrate your robot, the following procedure will guide you through how to replace the electronic switch board if necessary.

Materials Needed:

1. T10 Torque driver provided with your robot
2. Replacement switch board

Tip probe replacement procedure:

1. Remove the cover which encases the switch by gently pulling the plastic housing.
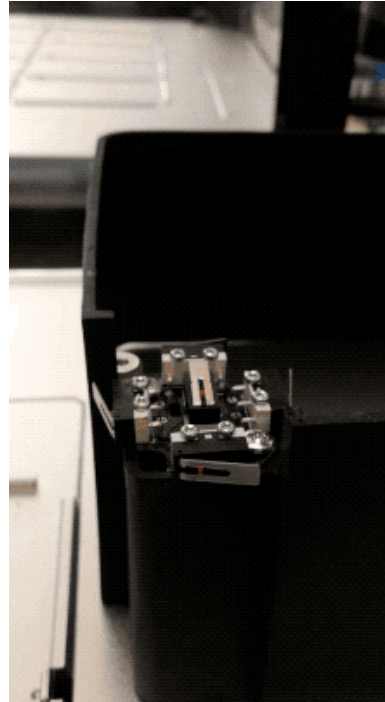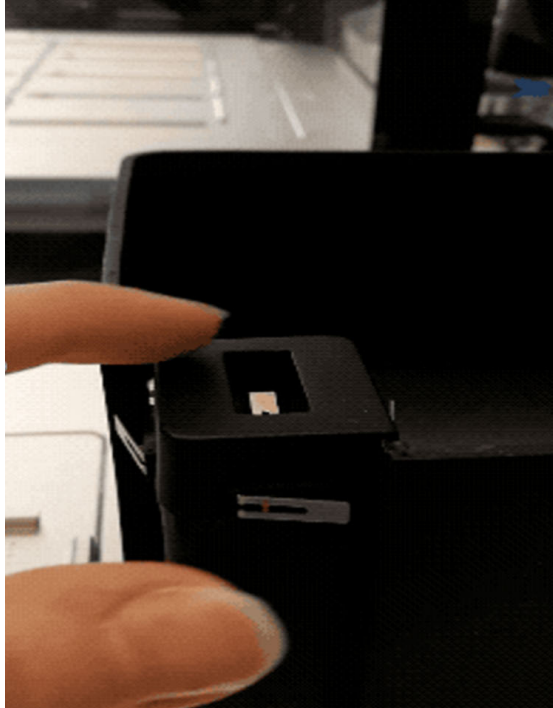
2. Once the housing is removed remove the 2x - T10 torque screws pictured below

T10 screws
to be
removed
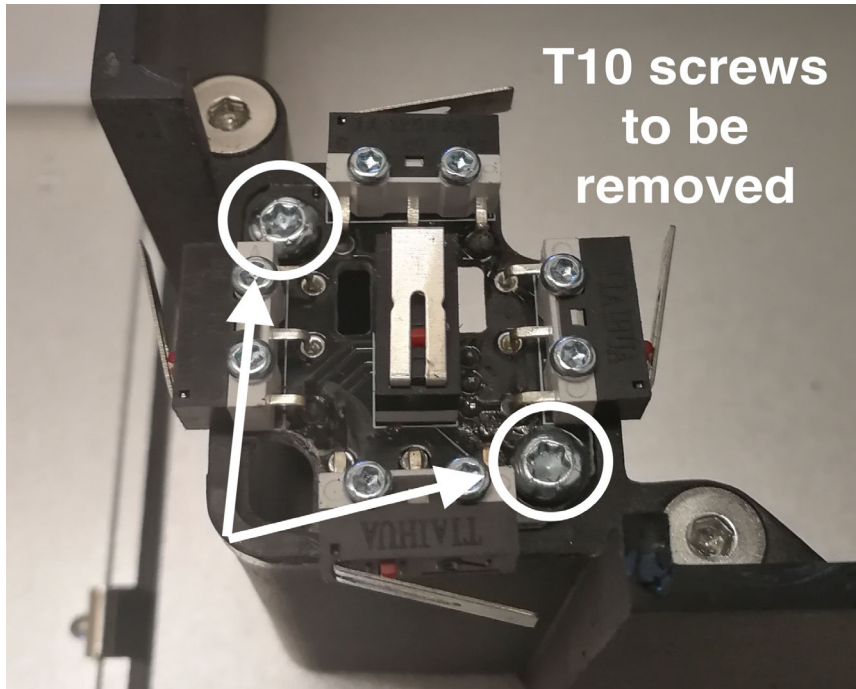
3. Carefully lift the switch plate
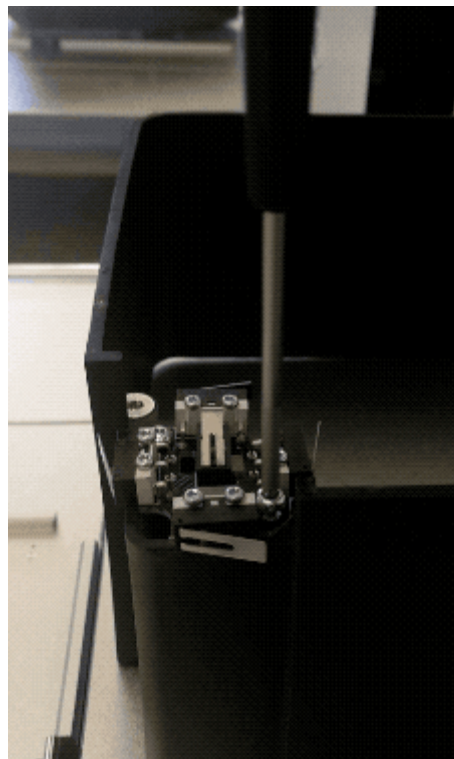
4. Unplug the switch plate wire from the connector and replace with the new switch board.
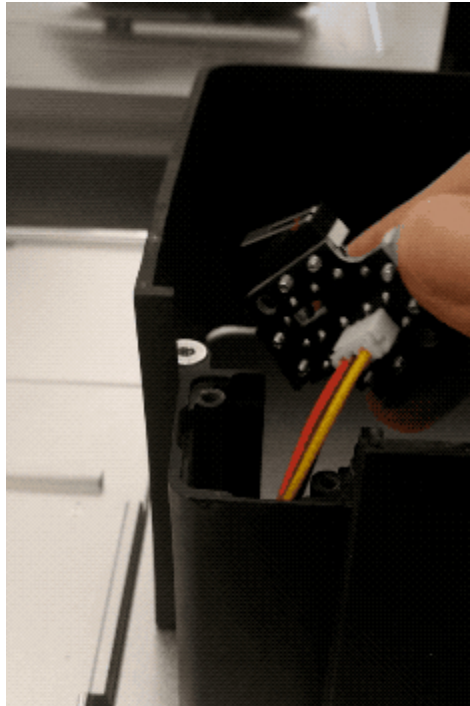
5. Fasten the new switch plate using the 2x - T10 torque screw removed previously.
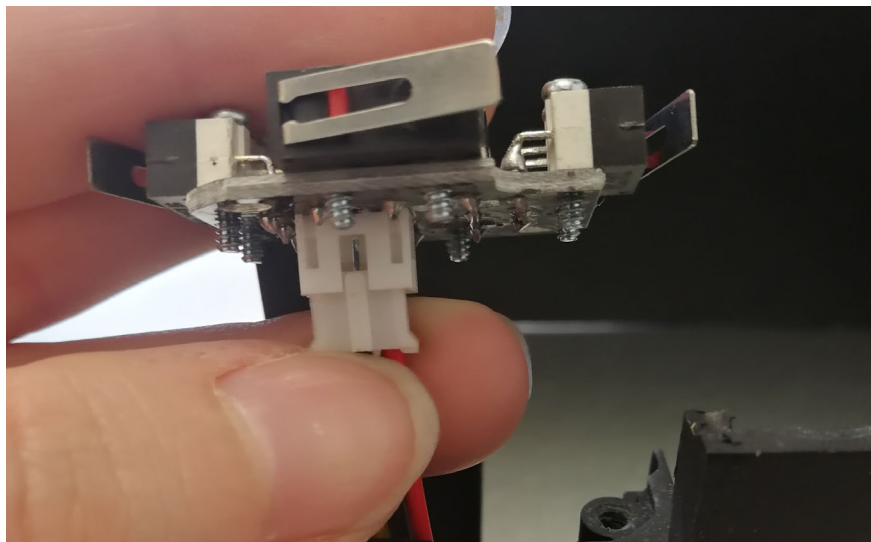
Make sure that the switches are installed in the orientation shown above. (The front switch should have the hinge on the left.) If the switches are installed backwards, pipette calibration will be unreliable.

9.19 SmoothieError: What is it and how do I fix it?

This article includes the following information about SmoothieErrors:

1. [Background](#)
2. [Common preliminary questions: Context of error](#)
3. [Types of error messages](#)

Background

A SmoothieError is how the OT-2 reports something that went wrong with a motor controller command. It is usually a symptom of an underlying mechanical or electrical problem. A variety of things can cause a SmoothieError.

SmoothieErrors indicate problems with the motor controller board and have a specific code ([G-code](#)) that dictates [how you should read them](#). A working knowledge of G-code often gives you more information for troubleshooting. For example, it can help you narrow down the cause of potential SmoothieErrors.

Common preliminary questions: Context of error

Listed below are questions you can use to start diagnosing and troubleshooting a SmoothieError. Opentrons Support may also ask you to provide some or all of this information if you encounter a SmoothieError.

1. What were you and the robot doing when this error popped up?
2. Take note of as many details as possible that span the set-up of the OT-2 for your protocol(s) up until the point the error occurred:
    - At what step did this problem arise during your use of the OT-2?
    - Was there a physical crash or any contact the pipette had with labware/OT-2 recently? Did this bring up the error?
    - Does this routinely happen in your protocol/calibration steps?
    - Has this happened before with this specific pipette in use?
3. Was there anything that you were doing prior to this issue occurring that might have caused it?
4. Examples of this might be:
    - Using non-Opentrons tips
    - Updating from a much older software version
    - Having previous issues with your OT-2
5. Take a photo of the robot when it caused this error as well as the error message or the exact text that pops up.
6. A video recording can also be extremely helpful but be sure to check that recreating the error state won't physically damage the pipette/OT-2 component.

7. How many pipettes are connected to the robot?
8. Were you able to successfully connect and use these pipettes in the past?
9. Can you**download the logs from the OT-2** **and send them over to the Support team for further investigation?**

 Types of smoothie error messages
There are two common SmoothieErrors: **Homing fail** and **hard limit**. See below for examples.

- Example **homing fail** error message:

1SmoothieError:
2M907 A0.1 B0.05 C1.0 X0.3 Y0.3 Z0.1 G4P0.005 G28.2C returned ALARM: Homing fail

- Example **hard limit** error message:

1SmoothieError:
2M907 A0.1 B0.5 C0.05 X0.3 Y0.3 Z0.1 G4P0.005 G0B3.5 returned ALARM: Hard limit -B

A SmoothieError can look like one of the two error messages shown below.

1. Example SmoothieError message that occurs during
protocol runs, calibrations, and homing:

⚠ Unexpected Error

*Exception <class 'opentrons.drivers.smoothie_drivers.driver_3_0.SmoothieError'> raised
by dispatch of token=UUID('e729ccee-fdd0-11ea-812f-b827eb5b1ec6') command=
<CalibrationCommand.move: 'move'> tipLength=None point=
<DeckCalibrationPoint.attachTip: 'attachTip'> axis=None direction=None step=None:
SmoothieError: M907 A0.1 B0.3 C0.05 X1.25 Y1.25 Z0.8 G4P0.005
G0X234.301Y121.9Z174.08 returned ALARM: Hard limit +Z*

An unexpected error has cleared your deck calibration progress, please try again.

If you keep getting this message, try restarting your app and/or robot. If this does
not resolve the issue please contact Opentrons Support.

CLOSE

2. Example SmoothieError message that occurs specifically during **Robot Calibration Check**:

If you receive a version of the first error message, proceed to the appropriate subpage to continue troubleshooting:

SmoothieError: Homing fail

SmoothieError: Hard limit

If you see the "**Robot Calibration Check**" error message, please review the logs you downloaded to identify the type of SmoothieError you are experiencing before proceeding to the appropriate subpage.

9.20 SmoothieError: Hard limit

A hard limit occurs when a limit switch was unexpectedly hit while the OT-2 was **not** homing. Essentially, it is when an endstop triggers when it is not supposed to (e.g., outside of a home).

The error message will mention the axis that failed, like "hard limit **+X**."
Note: Hard limits are normal after tip pickup. The Opentrons App won't show an error message in these cases. But the "hard limit" message will still show up in the logs, immediately after the tip pickup. Those specific "hard limit" log messages can be safely ignored.

Source for error trigger

There are several possibilities for the source of the hard limit:

1. Pipette plunger is stuck/not operating properly
2. Poor pipette attachment and/or electrical noise/static friction build up
3. Faulty pipette
4. Bent/broken limit switch
5. Electronic board malfunction/broken communication cable
   a. Typically a result of a faulty motor controller board, but could also be an error with the central routing board, Raspberry Pi, or any of the connecting cables.

Axes of errors

When examining the G-code of a hard limit, like this one:
1SmoothieError:
2M907 A0.1 B0.5 C0.05 X0.3 Y0.3 Z0.1 G4P0.005 G0B3.5 returned ALARM: Hard limit -B

The main piece of information you should focus on is the last part: -B. **This shows the axis/component at the time of the error and indicates + or - (unlike homing fails, which do not display + or -)** and, thus, you can start your troubleshooting there.

Read the G-code and double-check that the direction of the axis makes sense. For example, a "hard limit -**X**" should be impossible, because the X-axis limit switch is on the right, not the left. A direction that doesn't make sense indicates there is likely an issue that needs to be further investigated.
Table of axes

Axis Name	Description	Limit Switch Location
X	Gantry right (+X) and left (-X).	+X

Y	Gantry back, away from front window (+Y) and forward, towards front window (-Y).	+Y
Z	Left pipette mount up (+Z) and down (-Z).	+Z
A	Right pipette mount up (+A) and down (-A).	+A
B	Left pipette plunger up (+B) and down (-B).	+B
C	Right pipette plunger up (+C) and down (-C).	+C

General cases

Axes B/C - Pipette plunger

The robot is halting during calibration/running a protocol and throwing an error message like this:
1SmoothieError:
22M907 A0.1 B0.3 C0.3 X0.3 Y0.3 Z0.8 G4P0.005 G0B2.5 returned ALARM: Hard limit +B

In the above case, the **left pipette plunger** is throwing the error (**B-axis**).

1. Remove the pipette and inspect it to see if there is anything obviously physically wrong with it (any rattling noises inside if you gently shake it, for example).
2. Reattach to the left mount to see if the error persists
   a. If it works, most likely the cause was electronic noise/static build up that was dissipated when the pipette was removed.

3. If the error continues, try running the **unstick_axes.py protocol** (see instructions below) on the OT-2 and then home the OT-2 from the Opentrons App:
    a. Download and save the **unstick_axes.py protocol**.
    b. You will need to run the protocol on your OT-2 using the Opentrons App. To do this, open the Opentrons App. After connecting to your OT-2, go to the **Protocol** tab and upload the .py file you saved.
    c. Navigate to the **Run** tab and select **Start run**. You will be able to start the run without needing to calibrate as there is no labware involved.
        i. **Note**: You may need to run this protocol 2-3 times if the first attempt was unsuccessful.
    d. Afterwards, rerun your original protocol to be sure the pipette is functioning properly.
    e. If the original protocol is really long, you might want to consider making a quick test protocol in Protocol Designer that has the pipette pick up a tip and do a transfer with your labware.

Continuing on from the above example where the pipette was first mounted on the left when it was throwing the error...

If the error persists after running unstick_axes.py: Try attaching the pipette to the right mount and then home the robot from the Opentrons App. Additionally, you could go through the calibration steps or run a protocol after swapping which mount the pipette was originally on to test if the error is still occurring.

If a hard limit occurs on the right mount (would be on the C-axis in the error message): The pipette is most likely the source of the error, unless both mounts are experiencing a hardware error. The pipette may need to be replaced in this case as we typically don't want you to have to open the pipette.

If the pipette works perfectly on the right mount: Most likely the error is the left mount (faulty hardware) and will require hardware troubleshooting.

To be completely sure which case is occurring, you can also test another pipette on the left and right mount to see if it has the same behavior.

Axes Z/A - Pipette mount

The robot is halting during calibration/running a protocol and throwing an error message like this:

1SmoothieError:
2M907 A0.1 B0.3 C0.3 X0.3 Y0.3 Z0.8 G4P0.005 G0Z168.08 returned ALARM: Hard limit +Z

In this above case, the **left pipette mount** is throwing the error (**Z-axis**). The troubleshooting will be very similar to that mentioned for the pipette plunger, except that the investigation will involve the homing limit switches for the pipette mounts. You should follow the same steps from 1-4 above (**exclude step 3** as that is plunger specific).

If the error still occurs, you can try adjusting the Z/A motor currents on the OT-2. You can do this by running the **new_motor_currents.py protocol** (see instructions below) on the OT-2:

1. Download and save the **new_motor_currents.py protocol**.
2. You will need to run the protocol on your OT-2 using the Opentrons App. To do this, open the Opentrons App. After connecting to your OT-2, go to the **Protocol** tab and upload the .py file you saved.
3. Navigate to the **Run** tab and select **Start run**. You will be able to start the run without needing to calibrate as there is no labware involved.
4. Afterwards, rerun your original protocol to be sure the pipette is functioning properly.
5. If the original protocol is really long, you might want to consider making a quick test protocol in Protocol Designer that has the pipette pick up a tip and do a transfer with your labware.

If you're still seeing the error, proceed to hardware troubleshooting.

Axes X/Y - Gantry

The robot is halting during calibration/running a protocol and throwing an error message like this:
1SmoothieError:
2M907 A0.1 B0.3 C0.3 X0.3 Y0.3 Z0.8 G4P0.005 G0X168.08 returned ALARM: Hard limit +X

In this above case, the **gantry is hitting the X-axis limit switch incidentally**. The X and Y limit switches can be found on the back, right corner near the top of the Z-stage/gantry, each one labeled X and Y, respectively, on a white tab:

For these error messages:

1. Check that [all shipping brackets are removed](#).
2. Inspect the X/Y limit switches to see if they are broken/bent.
3. Inspect the electrical cables to see if they are properly connected or damaged in any way.
4. Press the X/Y limit switch a few times manually and try homing again. Sometimes the switch can get "stuck" and keep the motor controller board in a failure state, throwing this error.

5. If the error still presents, proceed to [hardware troubleshooting](#).

9.21 SmoothieError: Homing fail

A homing fail occurs when the limit switch was unexpectedly *not* hit while the OT-2 was homing. Homing fails usually happen if a) you're using different hardware with a different axis length or b) the axis is stuck or skipping *during the home*.

Possible situations - robot is homing

When in use, the robot goes through the homing process many times. This includes the gantry homing in the X and Y axes, the pipette mounts in the Z and A axes, and the pipette plungers in the B and C axes. You may have homing fails any time during gantry/pipette movement, including:

- Powering on the OT-2 (homing indicates it is powering on correctly)
- During calibration steps
- Start of a protocol or during protocol runs (after .drop_tip() or if specified in protocol using .home())
- [Robot Calibration Health Check](#)

Source for error trigger

There are several possibilities for the source of the homing fail:

1. Pipette plunger is stuck/not operating properly
2. Poor pipette attachment and/or electrical noise/static friction build up
3. Faulty pipette
4. Bent/broken limit switch
5. Electronic board malfunction/broken communication cable
   a. Typically a result of a faulty motor controller board, but could also be an error with the central routing board, Raspberry Pi, or any of the connecting cables.

Axes of errors

When examining the G-code of a homing fail, like this one:
1SmoothieError:
2M907 A0.1 B0.05 C1.0 X0.3 Y0.3 Z0.1 G4P0.005 G28.2C returned ALARM: Homing fail

The main piece of information you should focus on is the last part: G28.2C. **This shows the axis/component the OT-2 was trying to home at the time of the error, but does not indicate + or - (that's for [hard limits](#))** and, thus, you can start your troubleshooting there.

Table of axes

Axis Name	Description	Limit Switch Location
X	Gantry right (+X) and left (-X).	+X
Y	Gantry back, away from front window (+Y) and forward, towards front window (-Y).	+Y
Z	Left pipette mount up (+Z) and down (-Z).	+Z
A	Right pipette mount up (+A) and down (-A).	+A
B	Left pipette plunger up (+B) and down (-B).	+B
C	Right pipette plunger up (+C) and down (-C).	+C

General cases

Axes B/C - Pipette plunger

The robot is halting during calibration/running a protocol and throwing an error message like this:
1SmoothieError:
22M907 A0.1 B0.05 C0.05 X0.3 Y1.25 Z0.1 G4P0.005 G28.2B returned ALARM: Homing fail

In the above case, the **left pipette plunger** is throwing the error (**B-axis**).

1. Remove the pipette and inspect it to see if there is anything obviously physically wrong with it (any rattling noises inside if you gently shake it, for example).
2. Reattach to the left mount to see if the error persists
   a. If it works, most likely the cause was electronic noise/static build up that was dissipated when the pipette was removed.
3. If the error continues, try running the **unstick_axes.py protocol** (see instructions below) on the OT-2 and then home the OT-2 from the Opentrons App.
   a. Download and save the **unstick_axes.py protocol**.
   b. You will need to run the protocol on your OT-2 using the Opentrons App. To do this, open the Opentrons App. After connecting to your OT-2, go to the **Protocol** tab and upload the .py file you saved.
   c. Navigate to the **Run** tab and select **Start run**. You will be able to start the run without needing to calibrate as there is no labware involved.
      i. **Note**: You may need to run this protocol 2-3 times if the first attempt was unsuccessful.
   d. Afterwards, rerun your original protocol to be sure the pipette is functioning properly.
   e. If the original protocol is really long, you might want to consider making a quick test protocol in Protocol Designer that has the pipette pick up a tip and do a transfer with your labware.

Continuing from the above example where the pipette was first mounted on the left when it was throwing the error...

4. **If the error persists after running unstick_axes.py**: Try attaching the pipette to the right mount and then home the robot from the Opentrons App. Additionally, you could go through the calibration steps or run a protocol after swapping which mount the pipette was originally on to test if the error is still occurring.

If a homing fail occurs on the right mount (would be on the C-axis in the error message): The pipette is most likely the source of the error, unless both mounts are experiencing a hardware error. The pipette may need to be replaced in this case as we typically don't want you to have to open the pipette.

If the pipette works perfectly on the right mount: Most likely the error is the left mount (faulty hardware) and will require hardware troubleshooting**.**

To be completely sure which case is occurring, you can also test another pipette on the left and right mount to see if it has the same behavior.

Axes Z/A - Pipette mount

The robot is halting during calibration/running a protocol and throwing an error message like this:
1SmoothieError:
2M907 A0.1 B0.05 C0.05 X0.3 Y1.25 Z0.1 G4P0.005 G28.2A returned ALARM: Homing fail

In this above case, the **right pipette mount** is throwing the error (**A-axis**). The troubleshooting will be very similar to that mentioned for the pipette plunger, except that the investigation will involve the homing limit switches for the pipette mounts. You should follow the same steps from 1-4 above (**exclude step 3** as that is plunger specific).

If the error still occurs, you can try adjusting the Z/A motor currents on the OT-2. You can do this by running the **new_motor_currents.py protocol** (see instructions below) on the OT-2:

1. Download and save the **new_motor_currents.py protocol**.
2. You will need to run the protocol on your OT-2 using the Opentrons App. To do this, open the Opentrons App. After connecting to your OT-2, go to the **Protocol** tab and upload the .py file you saved.
3. Navigate to the **Run** tab and select **Start run**. You will be able to start the run without needing to calibrate as there is no labware involved.
4. Afterwards, rerun your original protocol to be sure the pipette is functioning properly.
5. If the original protocol is really long, you might want to consider making a quick test protocol in Protocol Designer that has the pipette pick up a tip and do a transfer with your labware.

If you're still seeing the error, proceed to hardware troubleshooting.

Axes X/Y - Gantry

The robot is halting during calibration/running a protocol and throwing an error message like this:
1SmoothieError:
2M907 A0.1 B0.05 C0.05 X0.3 Y1.25 Z0.1 G4P0.005 G28.2X returned ALARM: Homing fail

In this above case, the **gantry is not hitting the X-axis limit switch**. The X and Y limit switches can be found on the back, right corner near the top of the Z-stage/gantry, each one labeled X and Y, respectively, on a white tab (see below, circled in red):

For these error messages:

1. Check that all shipping brackets are removed.
2. Inspect the X/Y limit switches to see if they are broken/bent.
3. Inspect the electrical cables to see if they are properly connected or damaged in any way.
4. Press the X/Y limit switch a few times manually and try homing again. Sometimes the switch can get "stuck" and keep the motor controller board in a failure state, throwing this error.
5. If the error still presents, proceed to hardware troubleshooting.

9.22 Tip pickup issues

These troubleshooting steps will help you in the case that you've been experiencing any of the following issues:

- Tips unexpectedly dropping
- Pipette not aspirating properly
- Uneven aspiration across channels of a multichannel

Step 1: Make sure multi-channel pipettes are mounted level
Multi-channel pipettes are especially sensitive to tilt. When attaching them, some extra care is needed to make sure they are perfectly straight.

See: Leveling Multi-Channel Pipettes.

Step 2: Ensure there is no issue with the pipette
The first step to take is to ensure there is no issue internally with the pipette. To do this, perform the following:

1. Upload a protocol that simply picks up tips and transfers a liquid
2. Pause the protocol directly after tip pick up
3. Use your hands to firmly secure the tip onto the pipette stem (do this for each stem if using a multichannel)
4. Resume the run
5. If this solves the problem, move onto the next step. If the issue persists, contact support with details of the issues and the steps you've taken.

Step 3: Increase pick up current (for GEN1 pipettes)
If you have a GEN1 pipette, you can adjust the pipette pick up current with the suggestions below:
By adjusting the **Pick Up Current** under the **Power/Force** section in the pipette settings you can increase the force applied when the pipette picks up tips. The default current is 0.5 amps - the best method is to test in increments of 0.1 until you find a current that works more reliably. This works best in unison with the method above.

Note: The max current that should be applied is 1.2 amps. The pipette lifetime has only been tested with the default current.

9.23 Factory Resetting the OT-2

Factory reset can be used to delete the deck calibration, pipette offset, tip length, and labware calibration data

⚠️ **Warning:** To re-calibrate any calibration data that you check off. The data will be erased by the factory reset along with other related calibration.

Performing a factory reset on your OT-2

Before getting started make sure that you have

- **An up-to-date Opentrons App and OT-2.** See Updating your robot's software.

In order to initiate factory reset, you need to find it under **Advanced Settings > Factory Reset**, select **Reset.**

You will then be able to choose which calibration data to reset

Robot Configuration Reset

Warning! Clicking **restart** will erase your selected configurations and **restart your robot**. This cannot be undone

Labware Calibration
Clear labware calibration ☐

Boot Scripts
Clear custom boot scripts ☐

Deck Calibration
Clear deck calibration (will also clear pipette offset) ☐

Pipette Offset Calibrations
Clear pipette offset calibrations ☐

Tip Length Calibrations
Clear tip length calibrations (will also clear pipette offset) ☐

[CLOSE]   [RESTART]

Factory reset mechanics

You can clear the saved calibration data of deck, pipette offset, tip length, labware calibration, and boot scripts by selecting them and selecting the **RESTART** button. For more information on what each of these calibration types mean and how they relate, please go to our article on How Positional Calibration Works on the OT-2. Below are general relationships that will help guide what to recalibrate.

General relationships

1. Labware calibration will get erased if you reset pipette offset, tip length, or deck calibration.
2. Pipette offset will get reset if you reset tip length or deck calibration.
3. Deck calibration and tip length don't influence each other when reset.

What are boot scripts?

This is a special case that happens when you load a custom boot script onto your OT-2. Please only do when directed by Opentrons Support.

Why am I doing a factory reset instead of just recalibrating?

The OT-2 has certain boundary conditions which can be violated by a sufficiently bad calibration data. These will prevent the OT-2 from re-calibrating because it will need to start the calibration at the **sufficiently bad calibration data point.** In order for it to calibrate, the bad calibration data point needs to be deleted and have it gone to its initial state.

9.24 Pipette leveling issues

You might have trouble leveling the pipette perfectly. You might not be able to wiggle it quite far enough to get all 8 channels to touch the surface at once. Or, even if you can, the pipette might spring back a little bit after you tighten the 3 screws.
If the difference between channels is less than 0.15 mm, it's level enough (0.15 mm is the thickness of between one and two sheets of typical printer paper). If it's more than 0.15 mm, and you're unable to correct it, your OT-2 might need more adjustment. Contact Opentrons Support for help.

9.25 Manual Pipette Swapping

Manual pipette swapping (done by hand) will not damage the pipette or the robot.

9.26 My OT-2 hasn't restarted for an extended period.

If you are waiting for 10 minutes or so, check to see if the blue ring light on the front of your robot is on and solid (not flashing). If the light is solid, this means the update was applied successfully and your computer probably just missed the "I'm done" signal from the robot. This can potentially take longer for bigger updates.

9.27 Correct issues when using multiple modules of the same type

Please carefully review this article to confirm you have correctly configured all the modules on your OT-2's deck so that they can be successfully detected by the Opentrons App and called in the order according to your Python API protocol.
Be sure to double check the following:

1. Ensure modules are plugged into the correct USB ports of the OT-2 based on the order in which they will be called in your Python API protocol.
2. The Run tab in the Opentrons App will now show which slot it thinks each module is in. This enables you to send a quick command to each module to confirm it is in a specific slot.
    - For example, if you're using two Temperature Modules you can send a cooling command to each module and verify that the active module is in the correct slot according to the Opentrons App.

    - **Note**: If you go to quickly check your multiple Magnetic Modules by sending a command and, for example, you leave the magnets engaged (e.g., up) in the

Magnetic Module(s), the magnets will stay up until otherwise reset. **Therefore, please ensure you deactivate any modules before starting a run.**

3. When troubleshooting problems associated with multiple modules, Opentrons recommends that you do **not** change your Python API protocol. It is **much** easier to troubleshoot the physical configuration of multiple modules rather than rewriting your code!

4. You can [download the logs from your OT-2](#) and send them to the Opentrons Support Team for further troubleshooting if needed.

Note: When in doubt, remember that the OT-2 maps out each module sequentially: in order of appearance to the corresponding physical robot USB ports, from lowest port number to highest port number. Typically, USB ports will be numerically ordered from left to right.

Chapter 10 | Contact Information

If you need help in any of the following areas, please contact Opentrons Support.

- Technical support
- Applications support
- Service
- Repair
- Parts, supplies, and accessories

10.1 Opentrons Support Contact

Live chat

Click the button in the lower right-hand corner of the Opentrons App or the Opentrons website to open a message that will connect you with our support team.

Email

Contact Opentrons Support at support@opentrons.com.

Hours of operation

- Opentrons is based in New York City, in Eastern Standard Time (EST).
- We have customer support members available **Monday through Friday, 9am to 6pm ET (13:00 GMT to 20:00 GMT)**.

Observed Holidays

- New Year's Day
- Martin Luther King Jr. Day
- President's Day
- Memorial Day
- Independence Day
- Labor Day
- Thanksgiving Day (Thursday and Friday)
- Christmas Day

10.2 Opentrons Sales Contact

- Sales support line: +1 (857) 577-7656

- Email: sales@opentrons.com

Chapter 11 | Appendix

11.1 Opentrons GitHub

[GitHub](#) serves a similar functionality to Google Drive. It:

1. Helps people collaborate from around the world on the same project
2. Can contain different file types which can link and work together
3. Is essentially an online filing system

The main features that distinguish GitHub are:

1. Branches: A GitHub branch is essentially a set of coding changes with a unique name. This is helpful for us to test out new software changes before it rolls out to Opentrons users (or for you to develop your own additions!) and keep records of changes we made. So, if I wanted to test if the OT-2 works well with a new function, I'd test it out on my own "New_Function" branch.
2. Pull Requests: [Pull Requests](#) asks the Opentrons developer team if the changes made from a branch like "New_Function" should be "Pulled" into the main branch. The main branch (Referred to as "Edge") is "live" with our most updated/safe code and is what we utilize to produce software updates.
3. Software Versioning: GitHub allows teams like Opentrons to make clear versioning updates and changes (in our case the [Changelog](#) keeps track from update to update).
4. Bug Reporting: If there is something that just isn't working the way you would like it to, you can submit a [bug request](#) that describes the behavior, how to reproduce it, and how you expected the software to perform. Note: **If you have a technical issue, please contact support@opentrons.com instead of submitting a bug report.**
5. Feature Requests: If there is a feature that you would like to see on the OT-2, unlike a normal folder system, you can add a [Feature Request](#) so that our team can review and measure whether one day we can have it on the system.

To learn more about its history and uses in the world refer to:

1. [Tech Crunch's What Exactly is GitHub Anyway](#)
2. If you want to dig into the real details you can review [Git and GitHub for Beginners - Crash Course](#).

How to search if your bug or feature request was ticketed on GitHub

We are constantly improving our hardware and software. If you would like to get involved and use a Bug Report of Feature Request (or are looking to see if there is a work around after reviewing our support documentation), it's best practices to check in our GitHub first.

You can go to org:Opentrons [Input search] and then look at the **Issues** section. View here for more details on searching and please ensure that your search bar doesn't accidentally contain org:Opentrons org:Opentrons.
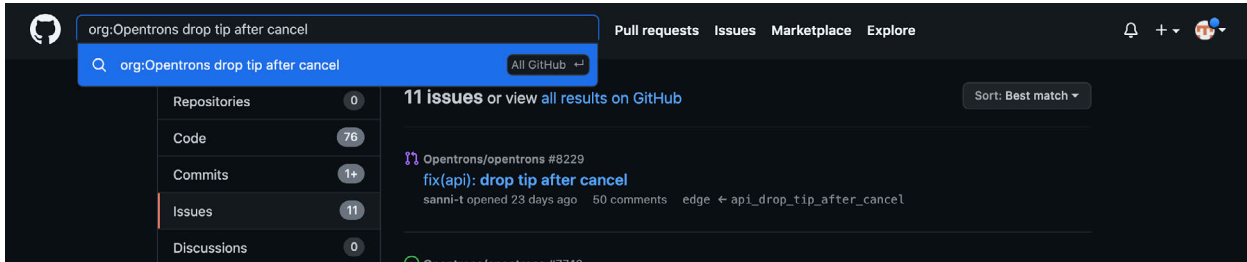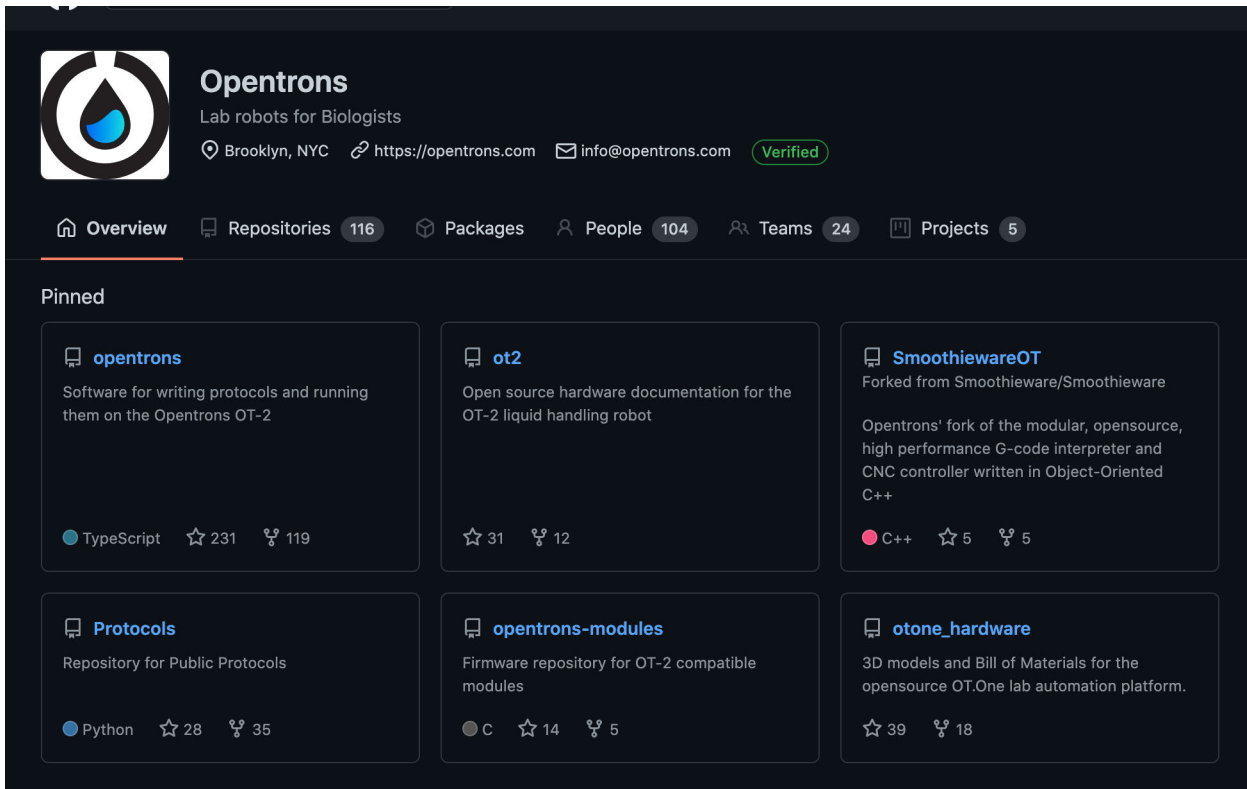
If you are on the correct page, then you can also use the interface and select what level you wish to search something on.
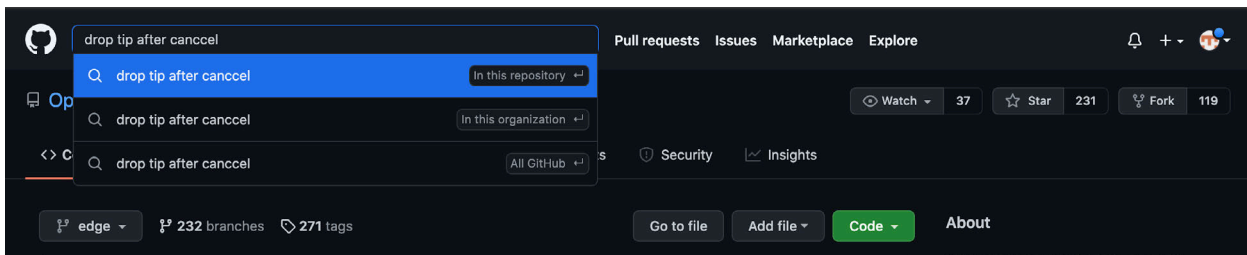
How to use Opentrons' GitHub

Reviewing the opening page on our GitHub can be intimidating. So, we are going to introduce each one with where it can be located and how to best utilize the information that is there. The repositories we will review are:

1. Opentrons/opentrons
2. Opentrons/Protocols
3. Opentrons/SmoothiewareOT
4. Opentrons/opentrons-modules

Opentrons/opentrons

The repository Opentrons/opentrons contains the source code for all of our user facing code. Each folder shows the "behind the scenes" of how anything from our Python API to our OT-2 connects with your computer on a network.

Opentrons/Protocols

The repository Opentrons/Protocols is the source code for our Protocol Library. If you are excited to share your protocol with the world you can follow our documentation! Please be advised that you should be using the Protocol Library version of protocols and not the GitHub page of protocols. Those are formatted to be read by our Protocol Library interface and not be usable as stand alone code. You can however go to the Readme file on the protocol page (or the title of the protocol) and paste the title into the Protocol Library search bar in order to locate the file.

```python
# Start protocol
def run(ctx):
    # Setup for flashing lights notification to empty trash

    [num_samples, starting_vol, binding_buffer_vol, wash1_vol, wash2_vol,
     elution_vol, mix_reps, settling_time, park_tips, tip_track,
     flash] = get_values(  # noqa: F821
        'num_samples', 'starting_vol', 'binding_buffer_vol', 'wash1_vol',
        'wash2_vol', 'elution_vol', 'mix_reps', 'settling_time', 'park_tips',
        'tip_track', 'flash')
```

Opentrons/SmoothiewareOT

The Opentrons/SmoothiewareOT repository gets a bit more complicated. Before we were talking primarily about software that has different computers talking to each other, which is like two people communicating. Now we are moving into something called "firmware" which is where a computer tells hardware what to do. This is more similar to a person's brain communicating with muscles. Instead of electrical impulses sent from a brain, the OT-2/OT-2 use something called "G-code."

The OT-2/OT-2 uses a fork (modified version of the original) of an open source G-code (robot command) interpreter from Smoothie. If you are investigating how best to control the OT-2 gantry from a computer instead of using our software, this is an excellent resource to learn about how the commands are sent.

Opentrons/opentrons-modules

The Opentrons/opentrons-modules is a similar situation, but instead of working with the hardware to move an OT-2, it is instead focused on handling G-code within the context of modules.

It can cover:

1. Firmware for all Opentrons modules - Temperature Module (Tempdeck), Magnetic Module (Magdeck) and Thermocycler.

2. Libraries required to build any module firmware
3. Module board file definitions for Arduino (for the arduino-based modules)
4. Other files required for development & testing

It can cover:

1. Firmware for all Opentrons modules - Temperature Module (Tempdeck), Magnetic Module (Magdeck) and Thermocycler.
2. Libraries required to build any module firmware
3. Module board file definitions for Arduino (for the arduino-based modules)
4. Other files required for development & testing

11.2 Warranty

All hardware purchased from Opentrons is covered under a 1-year standard warranty. Read this section to learn more about the warranty terms.

OT-2 Warranty

The OT-2 has been rigorously stress-tested to last for more than 3 years of full-time operation. If anything on the robot, including pipettes & all Opentrons modules, breaks from normal wear and tear within 1 year of its arrival at the user's lab, the robot can be replaced for free. After 1 year, Opentrons will no longer replace hardware.

Exchanges Under Warranty

Opentrons will only replace items damaged during the shipping process or factory defective items.
To exchange an item for the same item, send an email to support@opentrons.com and the Opentrons support team will reach out with further instructions.

Warranty FAQ

Who is covered by the Warranty?

Opentrons warranty covers the original purchaser of Opentrons products. The warranty is non-transferable and only extends to the original purchaser.

What is covered by the Warranty?

Opentrons warranty applies to Opentrons hardware (OT-2, OT-2 Pipettes, OT-2 modules) sold by Opentrons Labworks, headquartered at 20 Jay St, Suite 528, Brooklyn, NY 11201 or by authorized resellers. Opentrons warrants to the end-user of the products that they will be free of manufacturing defects due to part quality issues or poor workmanship and also warrants that the products will materially conform to Opentrons' published specifications.

What is the period of coverage?
The period of coverage starts the day the products are delivered and lasts for 12 months. In the event that your product needs to be replaced, your warranty period will start from the date of delivery of your replacement and will last 12 months from that date.

What is not covered by the Warranty?
This Warranty does not cover damage as a result of accidents, misuse, abuse or neglect. The Warranty also does not cover 3rd party alterations and labware.

How will Opentrons remedy my claim?
During the warranty period, Opentrons, at its discretion, will either repair, replace or provide a refund for the defective or non-conforming parts. Opentrons will bear the cost of labor, parts and shipping.

What am I responsible for?
If you discover a defect or believe your product is not functioning to our published specifications during the period of coverage, please contact Opentrons Customer Support through our webpage (www.opentrons.com), our Run App software, or by emailing support@opentrons.com. Opentrons Customer Support team will work with the user and determine the best remedy for the user's situation. In some cases, Opentrons may determine that a return is necessary to provide a replacement or refund. In those cases, the user will be asked to print out a prepaid return label, prepare the product for return and ship the product to Opentrons.

How much does Opentrons warranty cost?
Opentrons warranty is complementary and included with the order of any Opentrons products covered by the warranty (see above "What is covered by the warranty?").

11.3 Extended Warranty

This section covers the Extended 12-month warranty offered to Opentrons users.
By purchasing the 12-month Extended Warranty, users get access to the following benefits for the Opentrons system:

- Extended warranty, maintenance and repairs on parts, modules and provisions for replacement of the entire OT-2 and modules.
- Semi-annual health check to ensure instrument performance.
- Free standard shipping (lead times vary per area)
- Faster parts and unit replacement without waiting for an onsite service engineer

Note: Opentrons is the only company that offers a unit replacement when our team has deemed it eligible for a replacement and if the hardware serial number is covered by our 12-month Extended Warranty.

Exclusions

This 12-month Extended Warranty does not include coverage for the following:

- Pipettes
- Repairs and replacement requests caused by accidents, misuse, abuse or neglect
- Integrated devices
- Protocols and Software
- Reagents and Consumables

Each of our 12-month Extended Warranty is associated with a hardware serial number. Opentrons does not cover Opentrons hardware with a serial number that is not associated with the 12-month Extended Warranty contract.

11.4 60-Day & 1st Year vs. the 12-Month Extended Warranty

The 12-month Extended Warranty starts at month 13 but needs to be purchased within 60 days of hardware purchase. There are no more warranties available for sale after the 24-month period.

11.5 Warranty Purchasing FAQs

Eligibility

The 12-month Extended Warranty is available to all direct customers purchasing Opentrons hardware. This warranty is currently not available to purchases made via a distributor nor is it available for second hand Opentrons hardware purchased outside of Opentrons.

How to purchase 12-month Extended Warranty

The 12-month Extended Warranty can be purchased by contacting territory sales representatives or by emailing sales@opentrons.com.

Price

The 12-month Extended Warranty is priced at 25% of the cost of the equipment the warranty was purchased for.

11.6 How to handle viscous liquids in Protocol Designer

The OT-2's default parameters have been optimized to work with aqueous reagents. Because of this, viscous liquids will require some adjustment of these parameters to improve liquid handling abilities. From internal testing, we know that the OT-2 with GEN2 pipettes can handle viscous liquids of varied viscosity with accuracy comparable to water handling capacity of OT-2 along with high precision. Here are some tips on how to handle these liquids to have a successful protocol run.

1. The first step is to aspirate/ dispense slowly so you have a clean aspiration/ dispensing cycle.
    a. Dead volume is generated if the excess liquid inside the tip is not removed after blow out.
    b. The lower flow rate allows all the liquid to be removed compared to higher flow rate where the liquid dispenses from the center but excess liquid sticks to the wall of the tip.
2. Add a delay after aspirating and dispensing liquid (and touch tip if applicable)

Recommended parameters for handling viscous liquids. Keep in mind that these are guidelines, so you may have to change parameters a bit to fit your needs:

Table 1. P20 optimized parameters

Pipette	Liquid	Aspiration Rate (uL/s)	Aspiration Delay (s)	Dispense Rate (uL/s)	Dispense Delay (s)	Touch tip
P20	Glycerol 10%	6.804	2	6.804	2	No
	Glycerol 90%	5.292	7	5.292	7	No
	Glycerol 99%	3.78	10	3.78	10	No
	PEG 8000 50% w/v	6.048	7	6.048	7	No
	Sanitizer 62% Alcohol	1	2	3.78	2	Yes
	Tween 20	5.292	7	3.024	7	Yes
	Engine oil	6.048	7	6.048	7	Yes

Table 2. P300 optimized parameters for different liquids

Pipette	Liquid	Aspiration Rate (uL/s)	Aspiration Delay (s)	Dispense Rate (uL/s)	Dispense Delay (s)	Touch tip
P300	Glycerol 10%	83.25	2	83.25	2	No
	Glycerol 90%	64.75	8	64.75	8	No
	Glycerol 99%	55.5	10	55.5	10	No
	PEG 8000 50% w/v	74	6	74	74	No
	Sanitizer 62% Alcohol	92.5	2	92.5	2	Yes
	Tween 20	13.9	10	13.9	11	Yes
	Engine oil	74	3	46.25	7	Yes

Table 3. P1000 optimized parameters for different liquids

Pipette	Liquid	Aspiration Rate (uL/s)	Aspiration Delay (s)	Dispense Rate (uL/s)	Dispense Delay (s)	Touch tip
P1000	Glycerol 10%	247.05	2	247.05	2	No
	Glycerol 50%	247.05	3	247.05	3	No
	Glycerol 90%	164.7	10	109.8	10	No
	Glycerol 90%	41.175	20	19.215	20	No

If you would like to learn more about the our viscous liquid testing, you can read about it here: https://opentrons-landing-img.s3.amazonaws.com/application+notes/Viscous+Liquids+App+Note.pdf

The OT-2's default parameters have been optimized to work with aqueous reagents. Because of this, viscous liquids will require some adjustment of these parameters to improve liquid handling abilities. From internal testing, we know that the OT-2 with GEN2 pipettes can handle viscous liquids of varied viscosity with accuracy comparable to water handling capacity of OT-2 along with high precision. Here are some tips on how to handle these liquids to have a successful protocol run.

1. The first step is to aspirate/ dispense slowly so you have a clean aspiration/ dispensing cycle.
   a. Dead volume is generated if the excess liquid inside the tip is not removed after blow out.

b. The lower flow rate allows all the liquid to be removed compared to higher flow rate where the liquid dispenses from the center but excess liquid sticks to the wall of the tip.
2. Normally after completing a dispensing step, using a slower flow rate is recommended for the blowout function.
3. However, if the blow out volume is too low or too high for the liquid dispensing:
    a. An air gap can be implemented before aspirating (currently only in the Python API) combined with a slower dispense rate
        i. The trade-off is that less liquid can be aspirated to prevent exceeding the maximum volume limits of the pipette.
    b. This is not recommended for GEN2 P20S pipettes
4. Something that is also recommended is a slower withdrawal speed after aspiration and dispensing.
    a. This prevents sticking of droplets on the tip and spilling of reagents in the labware.
5. Additionally, it is recommended that before aspirating or dispensing that you use move_to () to move the pipette tip close to the bottom of the labware at 10mm/s
    a. After aspirating / dispensing, you should use move_to() to move the pipette tip at 1-5mm/s to the top of the labware.

Recommended parameters for handling viscous liquids. Keep in mind that these are guidelines, so you may have to change parameters a bit to fit your needs:

Table 1. P20 optimized parameters

Pipette	Liquid	Aspiration Rate (uL/s)	Aspiration Delay (s)	Aspiration Withdrawal Rate (mm/s)	Dispense Rate (uL/s)	Dispense Delay (s)	Blowout Rate (uL/s)	Touch tip
P20	Glycerol 10%	6.804	2	5	6.804	2	0.5	No
	Glycerol 90%	5.292	7	2	5.292	7	0.5	No
	Glycerol 99%	3.78	10	2	3.78	10	0.5	No
	PEG 8000 50% w/v	6.048	7	5	6.048	7	0.5	No
	Sanitizer 62% Alcohol	1	2	20	3.78	2	0.5	Yes
	Tween 20	5.292	7	2	3.024	7	0.5	Yes
	Engine oil	6.048	7	1	6.048	7	0.5	Yes

Table 2. P300 optimized parameters for different liquids

Pipette	Liquid	Aspiration Rate (uL/s)	Aspiration Delay (s)	Aspiration Withdrawal Rate (mm/s)	Dispense Rate (uL/s)	Dispense Delay (s)	Blowout Rate (uL/s)	Touch tip
P300	Glycerol 10%	83.25	2	5	83.25	2	10	No
	Glycerol 90%	64.75	8	1	64.75	8	4	No
	Glycerol 99%	55.5	10	1	55.5	10	4	No
	PEG 8000 50% w/v	74	6	1	74	74	4	No
	Sanitizer 62% Alcohol	92.5	2	20	92.5	2	4	Yes
	Tween 20	13.9	10	1	13.9	11	7	Yes
	Engine oil	74	3	2	46.25	7	10	Yes

If you would like to learn more about the our viscous liquid testing, you can read about it here:
https://opentrons-landing-img.s3.amazonaws.com/application+notes/Viscous+Liquids+App+Note.pdf

11.7 What are dead volumes, and how do I reduce them?

What is dead volume?

Dead volume refers to the residual volume of liquid left in the well or lost to waste that cannot be used in the application due to the risk of aspirating air. This can be due to well geometry and a number of other factors. High dead volumes may be concerning when pipetting valuable liquids such as master mixes, enzymes, or antibody solutions but may be acceptable in the case of low-cost reagents like water or sodium solutions. Since we are no longer pipetting by hand, we need to also consider that we are only able to access the wells from the top down rather than having a full range of motion to maneuver and consolidate small droplets within the wells of labware, as is the case with most liquid handling platforms.

What factors contribute to dead volume?

The amount of dead volume depends on a few factors. These can be attributed to the labware type and geometry, the properties of the liquid used, the pipette being used, as well as your specific environmental conditions.

Labware

The geometry and material of your labware can greatly affect the dead volumes required. 96 well plates tend to have a conical shape, with a reduced surface area as you approach the bottom of the labware. This geometry is ideal for reducing dead volumes where needed- conical geometry with decreasing surface area toward the bottom will allow for smaller dead volumes. However, the opposite is true for things like large reagent reservoirs. The large surface area of the wells in this type of labware lend themselves to the requirement of a larger dead volume. When choosing labware in the interest of reducing dead volumes, look for items with conical or V shaped bottoms. If you are not worried about high dead volumes, flat bottom plates are perfectly acceptable!
Material also plays a role in dead volumes. Most scientific labware is manufactured from materials that are by default hydrophobic and tend to bead liquids, reducing liquid adhesion to the walls of the well and reducing the dead volumes by wicking the liquids to the bottom of the well where it can be easily accessed. Some hydrophilic materials and coatings do exist however, usually in the case of plates intended for cell culturing and protein binding. Both of these plate material options are perfectly acceptable for use on the OT-2, but bear in mind the effects on dead volume and the goals of your workflow when selecting labware.

Liquid type

Liquids have many properties that may affect required dead volumes. The largest of these are threefold: viscosity, surface tension, and pipette tip submerge depth.
Viscous liquids will need to be aspirated and dispensed at a lower flow rate to ensure accurate volumes and a full dispense of the liquid. Quickly aspirating viscous liquids can lead to inaccurate pipetting volumes and you may run out of liquid before expected due to this.
Surface tension refers to the property of a liquid that allows it to resist some external force, like a pipette tip plunging through the surface. Liquids with high surface tension may tend to require deeper submersion of the pipette tip in order to properly aspirate without introducing air into the

tip. This may lead to an increase in dead volumes due to the tendency of liquids to adhere to the outer wall of pipette tips.

When working with flat bottom labware, it is important to note the behavior of the liquid in a low volume state. Some liquids may tend to wick to the perimeter of the wells when the reservoir is nearing depletion, whereas some may form beads near the center.

The depth at which a pipette tip is submerged into a liquid also plays a large role in determining dead volumes. The deeper a tip enters into a liquid, the more contact that liquid has with the outer surface of the pipette tip. As the tip-liquid contact surface area increases, the resulting dead volume increases. To avoid this, it's best to aspirate your liquids from just below the surface, accounting for the change in liquid volume resulting from the pipetting action.

Pipette type and technique

The type and number of channels on your pipette also play a role in dead volumes. Multichannel pipettes of high maximum volume will tend to result in the highest dead volumes, whereas a single channel low maximum volume will result in the lowest dead volumes. This is due to the surface area the liquid is exposed to. The more surface area the liquid is in contact with, the more likely it is that some amount of liquid will adhere to the surface of the tips. Using a low maximum volume pipette not only allows for less surface area contact, but also a finer positioning of the tip within the well.

Some pipetting techniques can be implemented into protocols to reduce the dead volume of a protocol. Actions like blow outs, touching the tip along the destination or source wells walls, and slower aspirate and dispense speeds can help to reduce dead volumes.

Environmental conditions

Lastly, environmental conditions can impact the dead volume, these factors include air temperature and humidity.

Air temperature and the resulting surface tension have an inverse relationship: as the air temperature increases, the surface tension decreases. Similarly, relative humidity and surface tension also have an inverse relationship. As relative humidity increases, the surface tension of the liquid decreases.

11.8 How to play sounds out of the OT-2

One really cool feature of the OT-2 is that it has a speaker that you can play sounds out of! 🔈 🎵 This article will go over the basics of how you can get your OT-2 to play sounds out of the speaker!

Where is the speaker located?

The OT-2 and OT-2 have an 8 Ohm and 2 Watt speaker on the Raspberry Pi. This speaker can play mp3/mp4 files by utilizing a software with Python called mpg123. The OT-2/OT-2 comes preloaded with /etc/audio/speaker-test.mp3 which is what is used in our example file path AUDIO_FILE_PATH.
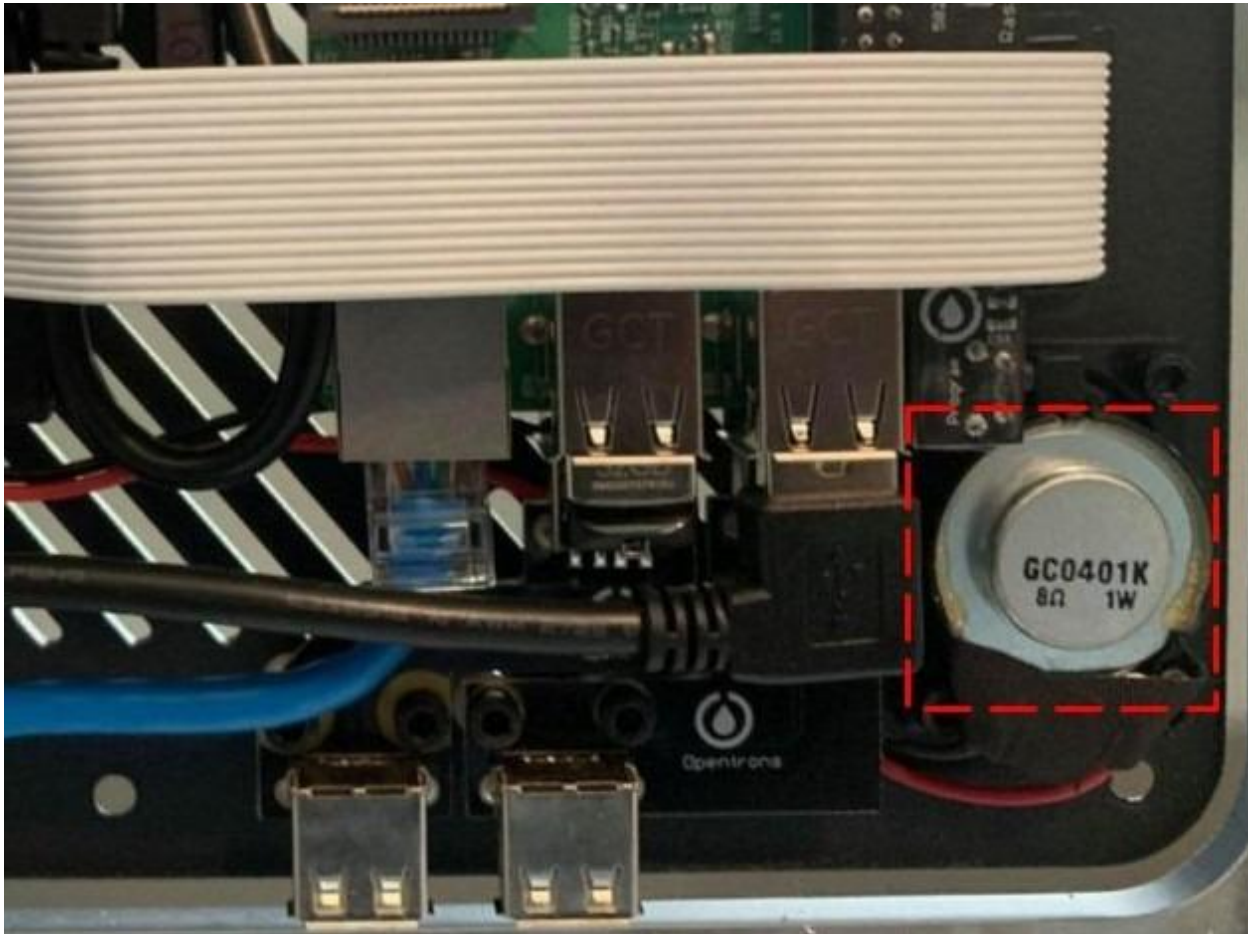
How do I use it?

This is only supported with our Python API. You can call the speaker function using <u>test_speaker</u>. An example of how to use it is shown below.

```
1 import subprocess
2 from opentrons import protocol_api, types
3 metadata = {'apiLevel': '2.9'}
4 AUDIO_FILE_PATH = '/etc/audio/speaker-test.mp3'
5 def run_quiet_process(command):
6     subprocess.check_output('{} &> /dev/null'.format(command), shell=True)
7 def test_speaker():
8     print('Speaker')
9     print('Next\t--> CTRL-C')
10    try:
11        run_quiet_process('mpg123 {}'.format(AUDIO_FILE_PATH))
```

```
12    except KeyboardInterrupt:
13        pass
14        print()
15
16
17def run(protocol: protocol_api.ProtocolContext):
18    tr2 = protocol.load_labware('opentrons_96_tiprack_20ul', '1')
19    p20 = protocol.load_instrument('p20_single_gen2', 'left', tip_racks=[tr2])
20    p20.pick_up_tip()
21    p20.drop_tip()
22    test_speaker()
```

11.9 Uploading files through Jupyter Notebook

Jupyter Notebook can be used to quickly upload, download, and access files on the OT-2.

Launching Jupyter Notebook

Each robot comes pre-installed with Jupyter Notebook, which can be accessed in your preferred web browser.

To launch the Jupyter Notebook server, select **OPEN** beneath Advanced Settings in the Opentrons App:
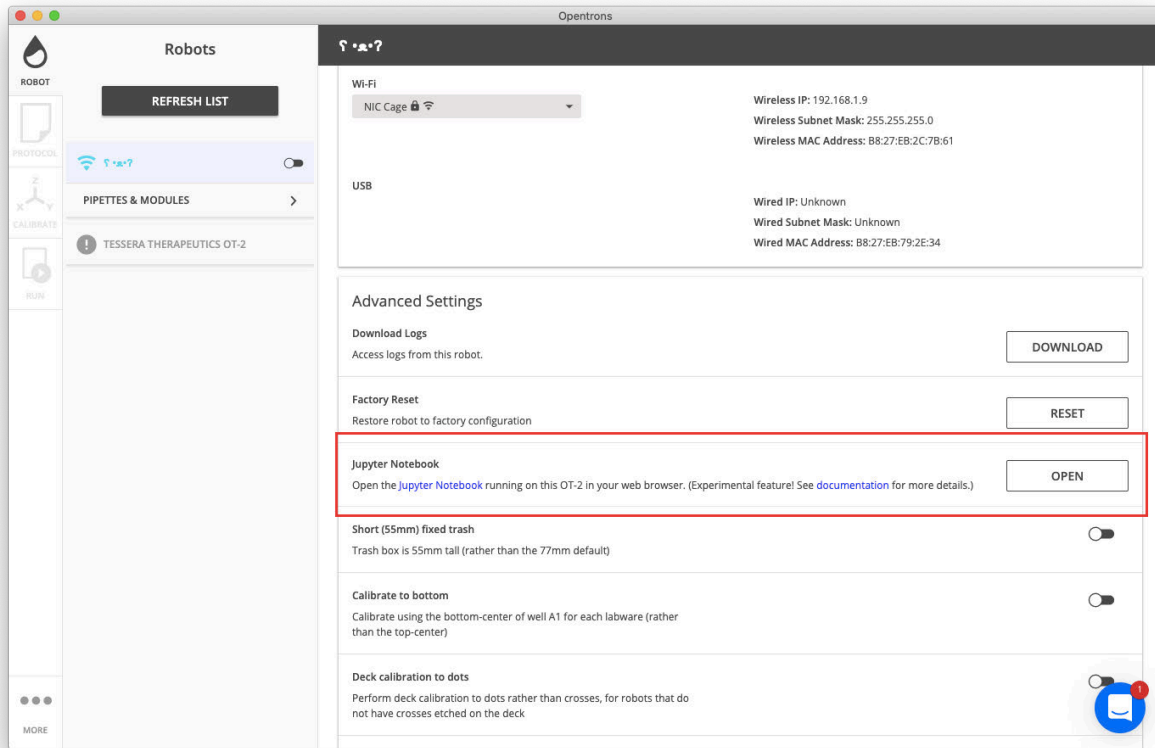
For more details, see [Running the robot using Jupyter Notebook](#).

Uploading a file

The home page will look like the screenshot below. Here, you can select **Upload** in the upper right:
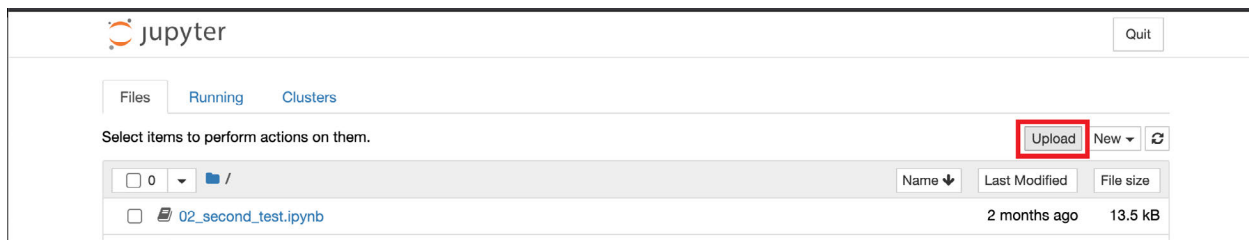
Then select the relevant file:
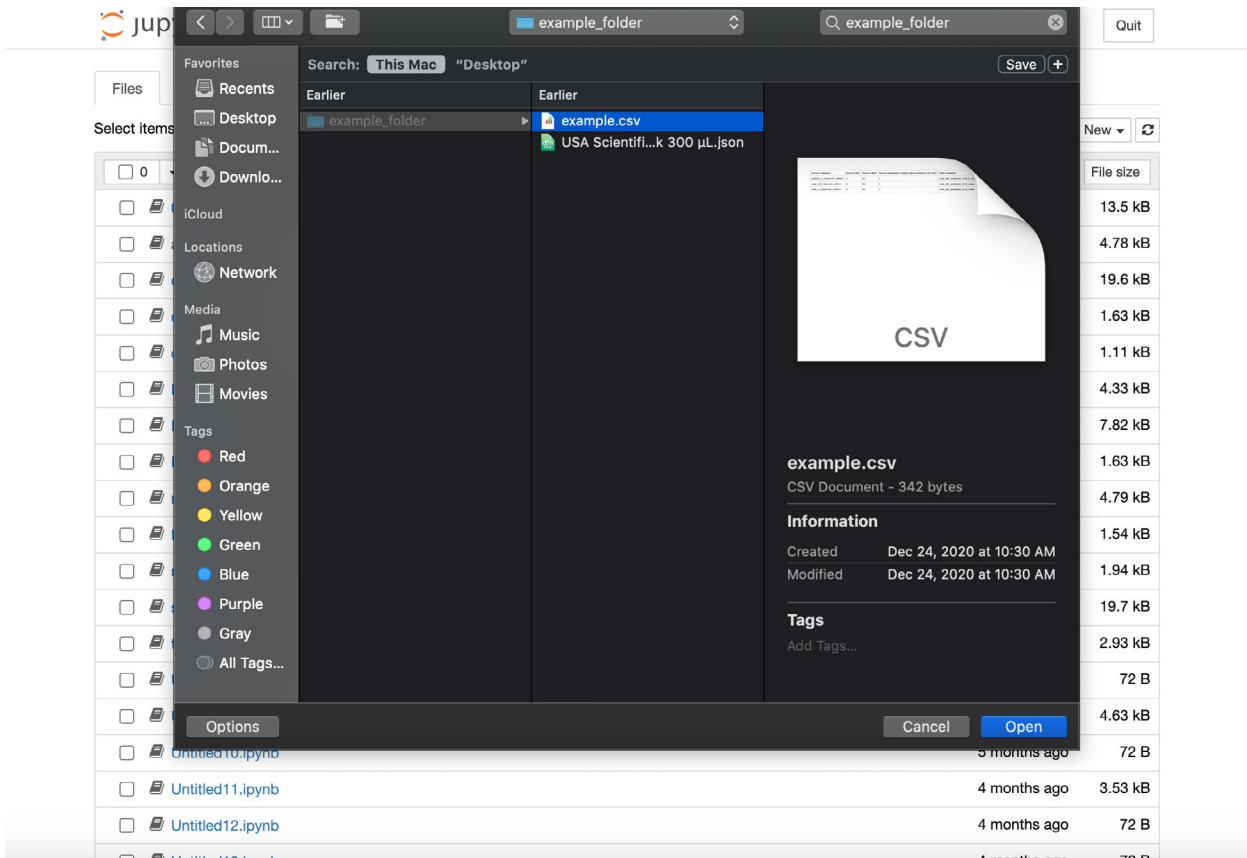
This file will be uploaded to the **/var/lib/jupyter/notebooks** directory:

Alternatives to using Jupyter Notebook

- For file transfers, it is also possible to use the Secure Copy SCP. This is helpful if you want to upload files to different locations on the OT-2 other than **/var/lib/jupyter/notebooks**
- If you are using CSV files, it is usually easier to copy and paste the CSV file to the Python file as described in our support article on handling CSV files

11.10 Setting up SSH access to your OT-2

SSH is a way to connect to and control your OT-2 over the command line. It's required for some troubleshooting and advanced tasks.
This guide will walk you through setting up SSH for the first time. There are two parts to the process: installing SSH on your computer and configuring your OT-2 to accept SSH connections from your computer.
You will create a special pair of files called an *SSH key pair.* The key pair comprises a *private key,* which remains on your computer, and a *public key,* which you will copy to your OT-2.
Setup will require a USB connection to your OT-2. After setup is complete, you will also be able to connect to your OT-2 with SSH over Wi-Fi.
This guide applies to robot server versions 3.16.0 and later.

Install prerequisites

You may need to install two small third-party tools:

- **OpenSSH**, the SSH implementation that we recommend.
- **cURL**, a tool to send simple network requests, which we need for setup.

On macOS

OpenSSH and cURL are both already installed by default on macOS.

On Linux

OpenSSH and cURL might be installed by default, depending on your distribution. If they're not, use your distribution's package manager to install them (for instance, sudo apt install ssh curl).

On Windows 10

Microsoft provides OpenSSH as a separately installable Windows component.
Open **Settings.** Go to **Apps** > **Apps and Features** > **Manage Optional Features**. Scan the list to see if **OpenSSH Client** is already there. If it's missing, at the top of the page, select **Add a feature**, locate **OpenSSH Client**, and install it.
You don't need to install cURL on Windows; a different tool will replace it.

11.11 Connect to your OT-2 with SSH

Introduction

SSH is a way to connect to and control your OT-2 over the command line. It is required for some troubleshooting and advanced tasks.

Before starting

SSH requires some first-time setup. Before following these instructions, make sure you've completed the setup guide:

Setting up SSH access to your OT-2

Once that's done, you're ready to connect.

1. Find your OT-2's IP address

In the Opentrons App, find the OT-2 that you want to connect to. Note its **Wired IP** or **Wireless IP**.
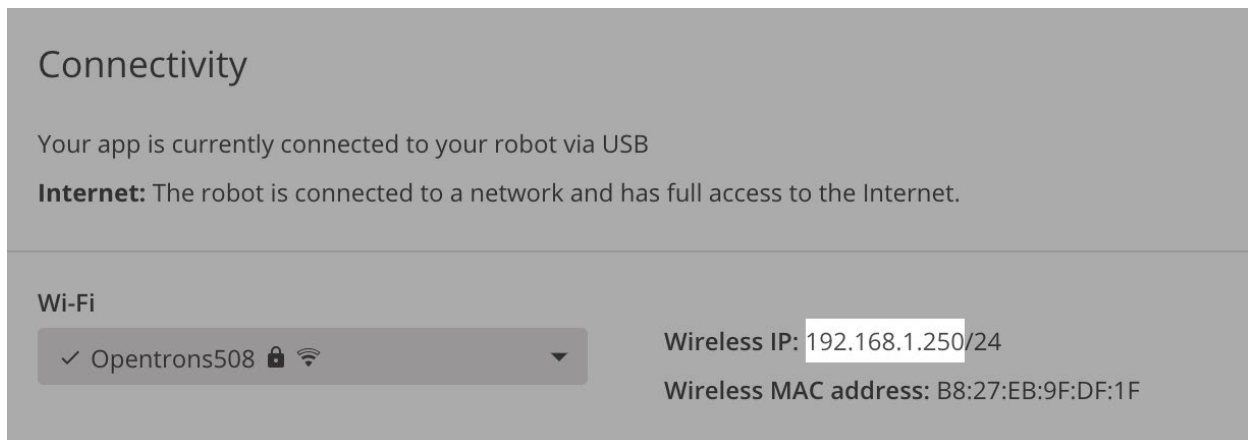
- If you use the **Wired IP,** the SSH connection will be made over USB.
- If you use the **Wireless IP,** the SSH connection will be made over Wi-Fi.

We recommend connecting over USB when possible, because the connection will be more reliable.

These IP addresses will change periodically, especially after the OT-2 is restarted or reconnected. You should double-check the IP address each time you want to make an SSH connection.

2. Connect

Open a **PowerShell** window if you're on Windows 10, or a **Terminal** window if you're on Linux or macOS.

Enter the following command. Replace ROBOT_IP with the IP address you found above.

1ssh -i ot2_ssh_key root@ROBOT_IP

Connection warnings

You may see a short warning saying, "Permanently added *[IP address]* to the list of known hosts." You can safely ignore it.

3. After establishing a connection.
If the connection worked, you'll be greeted by the Opentrons logo. ✨
Try running some commands on your OT-2. For example, enter the command:
1whoami
You should see the command output root .

11.12 Protocol Timer - Duration estimator

In consideration for scientists who want to be able to time protocols we have created a duration estimator starting in software version 4.6!

What does this timer do?
The timer takes how long each individual task is and compounds them into final results. So, from the picking up and dropping of tips, to modules, and even the flow rates of the pipettes all of those durations are compiled into the final result.
How do I use the timer?
Since it is still in Beta-phase it isn't integrated into the app, it instead can be accessed through the opentrons_simulate function

1. If you are more Python savvy than you can use our Python API and use the "-e" flag like so
   a. opentrons_simulate -e /path/to/protocol
   b. You might need to use pip install --upgrade opentrons if it doesn't work to ensure it is updated
2. If you are less experienced with Python please follow below:

For those that are having difficulty with the Python setup:

1. Open Replit
2. Select the Python template by clicking on it
3. Click "create repl"
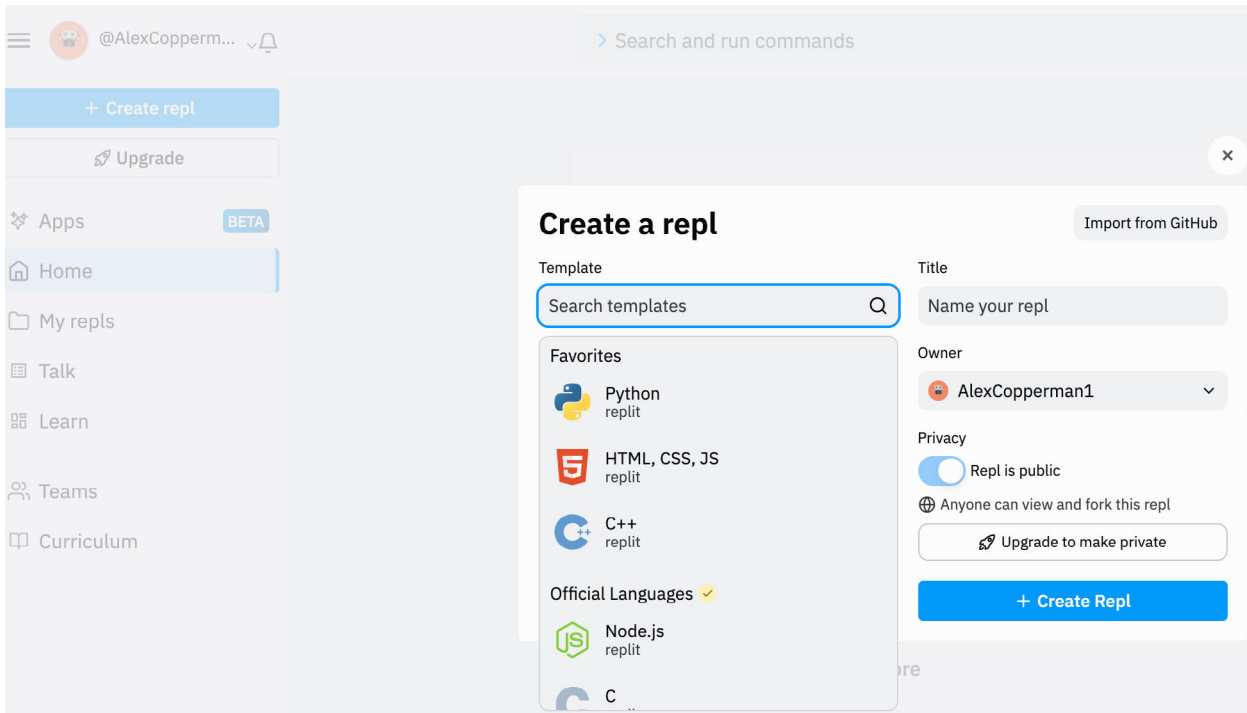
Move from console to shell
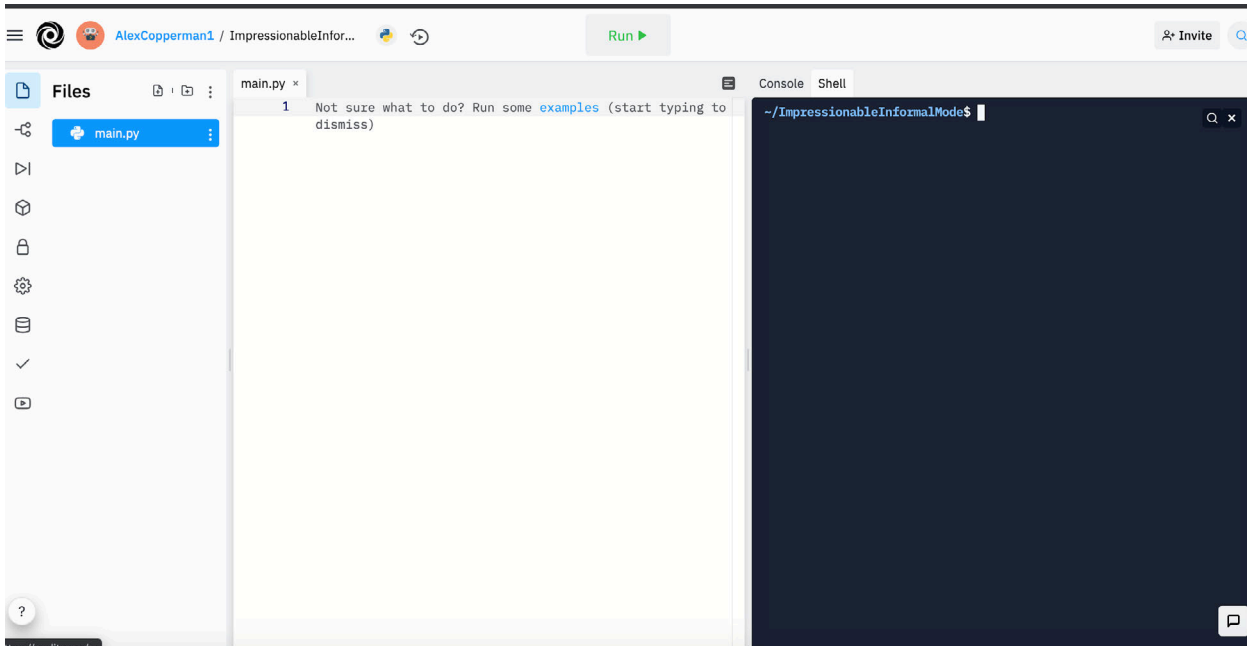
Type in from opentrons import protocol_api and then click the green Run button
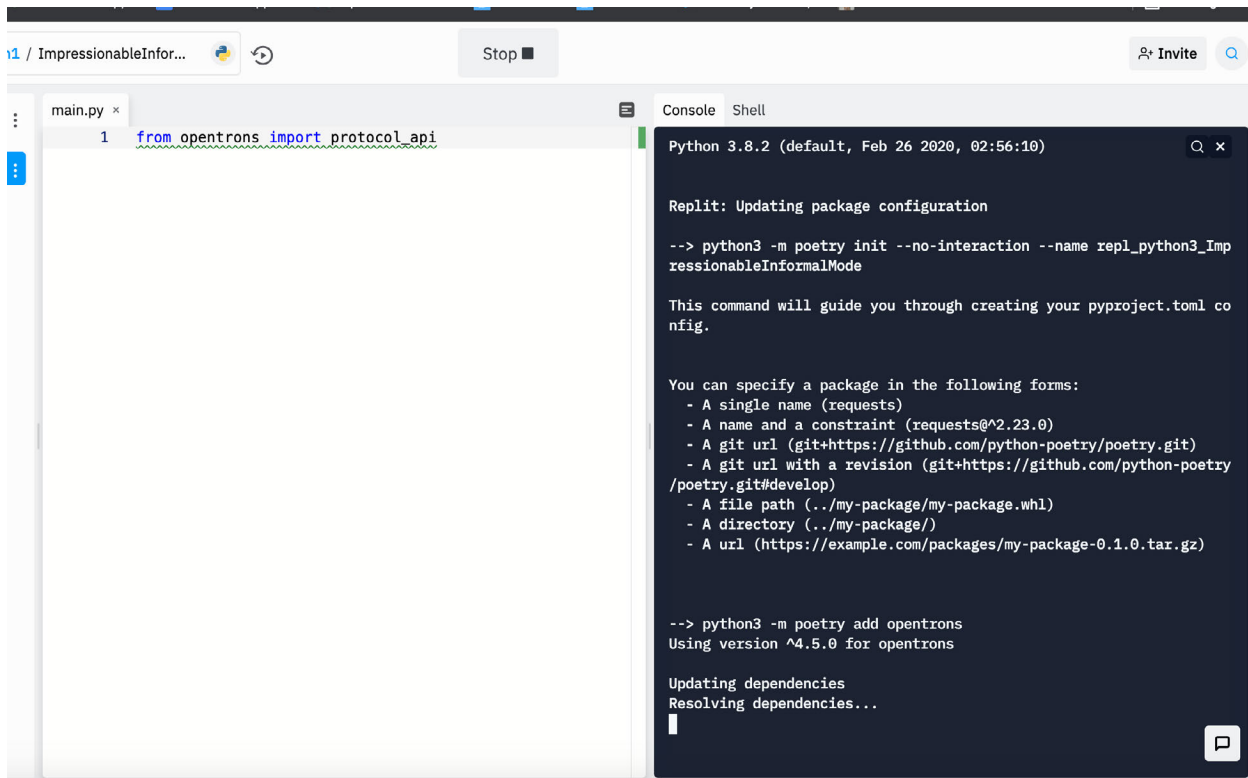
Click the three dots across from Files and then upload your file
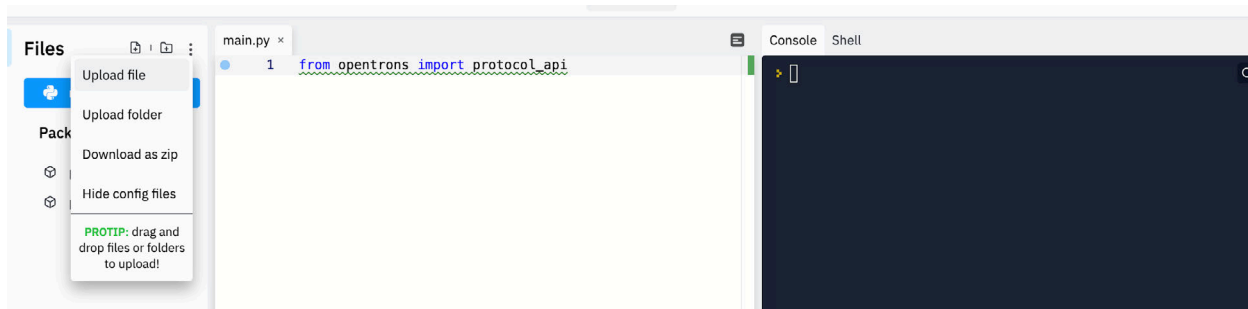
Enter ls to find the file that you are using and then copy and paste it after opentrons_simulate -e 'insert_protocol_name_here.json'
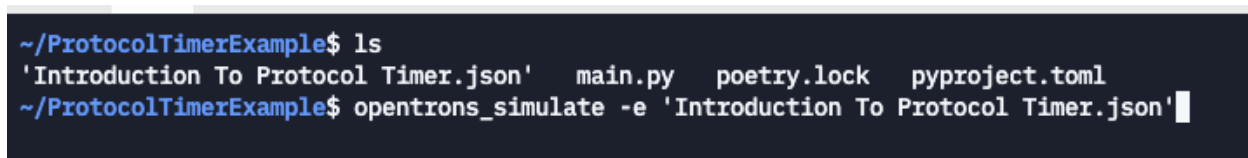
Both Protocol Designer protocol files and Python API protocol files are accepted by the Protocol Timer and will simulate using the methods above. The output will look like so:

The data supporting the timer

First and foremost, this is an experimental feature. There is an expected percent error for all protocols. This value is most correlated with the length of protocol, but we have seen even 4-hour protocols remain at most 10% error. We have listed out the tested protocols and provided the accuracy results of our protocol duration estimations in the **Appendix**.

Giving feedback

We would really appreciate your feedback to improve this product. Highly recommended to fill out this form if you have any suggestions or protocols that you have timed that have a higher percent error than 15% and want to send over.

Protocol Library protocols*

- BP Genomics Extraction
- Thermofisher Magmax Viral/Pathogen Kit
- Omega Biotek Magbind Viral DNA/RNA kit
- Zymo Extraction
- MGI Extraction
- Beckman RNAdvance Viral XP

*Each of these protocols were tested for 3 different sample totals: 8, 24, and 48

Custom testing protocol procedure

- Edit the sample number to the number of samples per trial.
- Load protocol onto the OT-2
- Report final run time

Accuracy data for each protocol with varying number of samples

BP Genomics Extraction

# of samples	Actual Time (minutes)	Expected time (minutes)	Percent Error
8	60	58	3
24	78	76	3
48	112	104	7

Thermofisher Magmax Viral/Pathogen Kit

# of samples	Actual Time (minutes)	Expected time (minutes)	Percent Error
8	52	54	4
24	76	74	3
48	95	104	9

Omega Bio-tek Magbind Viral DNA/RNA kit

# of samples	Actual Time (minutes)	Expected time (minutes)	Percent Error
8	46	45	2
24	62	60	3
48	85	82	4

Zymo Extraction

# of samples	Actual Time (minutes)	Expected time (minutes)	Percent Error
8	45	43	4
24	74	69	7
48	114	108	5

MGI Extraction

# of samples	Actual Time (minutes)	Expected time (minutes)	Percent Error
8	42	41	2
24	64	61	5
48	101	91	10

Beckman RNAdvance Viral XP

# of samples	Actual Time (minutes)	Expected time (minutes)	Percent Error
8	30	29	3
24	40	40	0
48	60	58	3

Custom testing protocol

# of samples	Actual Time (minutes)	Expected time (minutes)	Percent Error
48	246	222	10

11.13 Copying files to and from your OT-2 with SCP

Introduction
SCP is a command line tool for copying files between devices. You can use it:

- **To upload files from your computer to your OT-2.** This is useful for things like CSV input files.
- **To download files from your OT-2 to your computer.** This is useful for things like pictures taken with the OT-2's camera.

Using SCP to transfer files is an advanced task. To do it, you should be comfortable with the command line and Linux filesystems.

Before starting
SCP is installed alongside SSH, and it shares its first-time setup to configure the OT-2 to accept connections. Before following these instructions, make sure you've completed the SSH setup guide:
Setting up SSH access to your OT-2
1. Find your OT-2's IP address
In the Opentrons App, find the OT-2 that you want to transfer files to or from. Note the **Wired IP** or **Wireless IP.**

- If you use the **Wired IP,** the files will be transferred over the USB connection.
- If you use the **Wireless IP,** the files will be transferred over the Wi-Fi connection.

We recommend connecting over USB, when possible, because the connection will be more reliable.

These IP addresses will change periodically, especially after the OT-2 is restarted or reconnected. You should double-check the IP address each time you want to make an SSH connection.
2. Use scp to transfer files
Note: these commands must be run on your local liquid handler, not the robot.
Download a file from your OT-2 to your computer
1scp -i ot2_ssh_key root@ROBOT_IP:/path/on/OT-2 /path/on/computer

- ROBOT_IP should be replaced with your robot's actual IP address, as found in step 1.
- /path/on/OT-2 is where to find the file on your OT-2.
- /path/on/computer is where on your computer you want to copy the file to.

Upload a file from your computer to your OT-2
1scp -i ot2_ssh_key /path/on/computer root@ROBOT_IP:/path/on/OT-2

- ROBOT_IP should be replaced with your robot's actual IP address, as found in step 1.
- /path/on/computer is where to find the file on your computer.
- /path/on/OT-2 is where on your OT-2 you want to copy the file to.

Where on the OT-2's filesystem can I keep my files?

Most of the OT-2's filesystem is read-only. However, two places you can store files in are:

- The **/data/user_storage** directory, which is good for general-purpose storage.
- The **/var/lib/jupyter/notebooks** directory, which is also be accessible through your OT-2's Jupyter Notebook.

Files placed in either of these directories will persist across reboots and software updates.

11.14 Integrating the OT-2 with other lab equipment

Robot System Overview
The core of the OT-2 is a Raspberry Pi 3+ running a custom Linux distribution.
It connects to your computer either via a Wi-Fi network, or via a USB-to-Ethernet adapter. Either way, it communicates with your computer using networking technology, and therefore can be communicated with and controlled using tools designed to talk to Linux computers via the network, like SSH.
Most of the filesystem is read-only. You should only keep your data in **/data** or **/mnt/usbdrive.**
The OT-2 also runs a web server. This web server is what the Opentrons App talks to and is how protocols are run.
An important thing to note about the Python code on GitHub is that **it will not control the robot from your computer**. The only way to control the robot from your computer is through the web server. The Python code on GitHub is the code that runs **on the robot itself**, and the reason we refer customers to it is either:

1. to use as a reference when writing code that interacts with it on the robot, or

2. to **simulate** protocols on their computer.

Summary: The robot runs a Linux system that executes Python code to control its hardware and provide a web server. The only way to control the robot is by running Python code on the robot. Data can be persistently saved and loaded in **/data** or **/mnt/usbdrive**.

OT-2 Integration Case Study Example

In the following sections, we'll consider a case study of integration: a protocol that requires talking to a balance that communicates via serial, some other instrument that communicates over the network using VISA, and an OT-2.

Integrating By Putting The OT-2 In Control

Since the robot runs Linux and is connected to the network, and has USB ports available on the Raspberry Pi, the code that controls the lab equipment could run on the OT2 itself. This is the simplest-to-implement method of integration.

In our case study, you could therefore:

- Connect the balance via a USB/serial adapter (FTDI) to a free USB port on the OT-2's Raspberry PI
- Connect the OT-2 and the VISA instrument to the same network
- Using pip, install pyvisa on the OT-2
- Write Python code that runs the protocol and runs on the OT-2

The Python code would, in addition to containing the normal protocol commands, connect to the balance via serial and the other instrument using pyVISA. You would run this code by transferring it to the robot using scp. You'll need to [set up SSH access to your robot](#) first.

1scp -i ot2_ssh_key /path/on/computer root@ROBOT_IP:/path/to/protocol.py

and then use ssh to remote into the robot and start the protocol:

1ssh -i ot2_ssh_key root@ROBOT_IP "nohup opentrons_execute path/to/protocol.py"

The nohup command means the command will continue even if the SSH session goes away. It makes sure the ssh command exits immediately and leaves the protocol running on its own. Without the nohup command, the protocol will stop if interrupted on your computer.

We're transferring the protocol to the robot and executing it directly rather than using the Opentrons app to avoid having to add a bunch of checks if the robot is simulating.

The protocol could save data to files in *data*, which you can retrieve after the protocol finishes using scp. It could also, since it is network-connected, transfer the files directly via the network to your computer.

Summary: Connect everything to the OT-2 and write a protocol that controls everything, saving and loading data either to the /data directory on the OT-2 or, via the network, on your computer.

Integrating While Keeping Your Computer In Control

Another option is to write a Python protocol that connects back to your computer. In our case study, you would:

- Connect the balance to your computer
- Have the code that interacts with the VISA instrument on your computer
- On your computer, run code that listens for the OT-2 protocol to reach out and talk to it
- Write a Python protocol that is uploaded through the Opentrons app and reaches out and talks to your server

This can be complex, but it does mean that your computer stays in control and you have minimal custom code to write. Consider writing a simple Python server using Flask (📖Welcome to Flask — Flask Documentation (2.0.x)), Bottle (Bottle: Python Web Framework — Bottle 0.13-dev documentation), or AIOHTTP (Welcome to AIOHTTP — aiohttp 3.8.1 documentation) (it's your choice - it will be on your computer. We like AIOHTTP, though).

You set up a route called /update, which something can POST to with a step name, and will return either "wait" or "proceed":

On your computer: This is written with AIOHTTP in mind

```
1 from aiohttp import web # You can install aiohttp with pip
2 import json
3
4 async def update(request):
5     """
6     This function serves POST /update.
7
8     The request should have a json body with a "step" key that at some point
9     has the value "done-aspirating".
10
11    It will return a json message with appropriate HTTP status.
12    """
13    try:
14        body = await request.json()
15    except json.JSONDecodeError:
16        text = await body.text()
17        print(f"Request was not json: {text}")
18        return web.json_response(status=400, # Bad Request
19                    data={'error': 'bad-request'})
20    if 'step' not in body:
21        print(f"Body did not have a 'step' key")
22        return web.json_response(status=400, # Bad Request
23                    data={'error': 'no-step'})
24    if body['step'] == 'done-aspirating':
25        # Here you might for instance check a balance
```

```
26      # attached to the computer to validate apsiration
27      print("Robot is done aspirating")
28      return web.json_response(status=200, # OK
29                    data={'done': True})
30
31# Create and run the actual server application
32app = web.Application()
33# Install the update function to serve the /update endpoint for POST
34app.router.add_post('/update', update)
35# Run the application
36web.run_app(app, # our application
37         host='0.0.0.0', # listen on all network interfaces
38                  # (change to 127.0.0.1 to listen only to
39                  # requests from this computer for testing)
40         port=80)      # the standard http port; may need to
41                  # change to something else if another
42                  # server is running
```

The Python protocol gets to a certain step, hits that endpoint, and either waits or proceeds:

```
1import requests # you can install this with pip
2
3def run(protocol):
4  tiprack = protocol.load_labware('opentrons_tiprack_300ul', '1')
5  trough = protocol.load_labware('nest_12_reservoir_15ml', '2')
6  pipette = protocol.load_instrument('p300_single', 'right', tip_racks=[tiprack])
7
8  pipette.pick_up_tip()
9  pipette.aspirate(10, trough.wells()['A1'])
10
11  done = False   # Poll to see if the server wants you to proceed
12  while not done:
13    r = requests.post('http://127.0.0.1/update', json={'step': 'done-aspirating'})
14    done = r.json()['done']
15
16  pipette.dispense()
```

This protocol can be simulated with opentrons_simulate while the server runs in another terminal. You will see the server print "Robot is done aspirating".

When this protocol is run through the opentrons run app, it will be simulated before it actually runs. That means that the protocol will run without the robot moving, including making the POST request to your computer. To avoid this, you can only do the request if protocol.is_simulating() is False .

If you want to avoid simulation entirely, you can transfer the protocol to the robot with scp and run it directly there, as above.

When the protocol is run on the robot, you must also change 127.0.0.1 to the IP address of your computer.

Summary: Write a protocol that connects to a webserver on your computer. Your computer is connected to the other equipment and waits for the protocol to connect and say it is ready.

Conclusion

One of these two methods should help out most integration tasks. If they don't, keep in mind that fundamentally the OT-2 is just a computer that runs Linux that happens to be connected to some hardware, and has a Python module for controlling that hardware already installed.
Opentrons has plans to make this sort of task easier in the future; but right now, this is what we've got.

11.15 Using the OT-2's camera

The Opentrons OT-2 has a camera built into its chassis, positioned to get a view of the deck. It has a resolution of 640x480px for still images and 320x240px for video.
The camera can be used in two ways: taking a picture (can only be a still image) through the HTTP API of the robot and saving images or video from the robot's terminal.

Advanced Usage Warning

While the OT-2 does have a built-in camera, the software functionality behind this feature is still in development. The existing iteration of the camera functionality is considered an advanced usage of the OT-2 features. We will support you in all efforts to be able to use this feature, but there are limitations to what we can support and what the camera can do.

Camera Limitations

1. If you take photos while the OT-2 is in motion it can render the OT-2 unresponsive.

Where the camera is located on the OT-2

Taking images with HTTP

You can send an HTTP request to the OT-2 to have it take a picture.

Host	Robot's IP address. See below.
Port	31950
HTTP method	POST
HTTP path	/camera/picture

Special HTTP headers	Opentrons-Version: 2
	(header required since OT-2 software version 4.0.0)

The OT-2 will respond with the picture in the response body, with Content-Type: image/jpeg.
 For instance, you can find your robot's IP address from the Opentrons App:

Connectivity

Your app is currently connected to your robot via Wi-Fi

Internet: The robot is connected to a network and has full access to the Internet.

Wi-Fi

✓ Opentrons 🔒 📶 ▼

Wireless IP: 10.10.3.187/22

Wireless MAC address: B8:27:EB:C0:96:36

USB

Wired IP: Unknown

Wired MAC address: B8:27:EB:95:C3:63
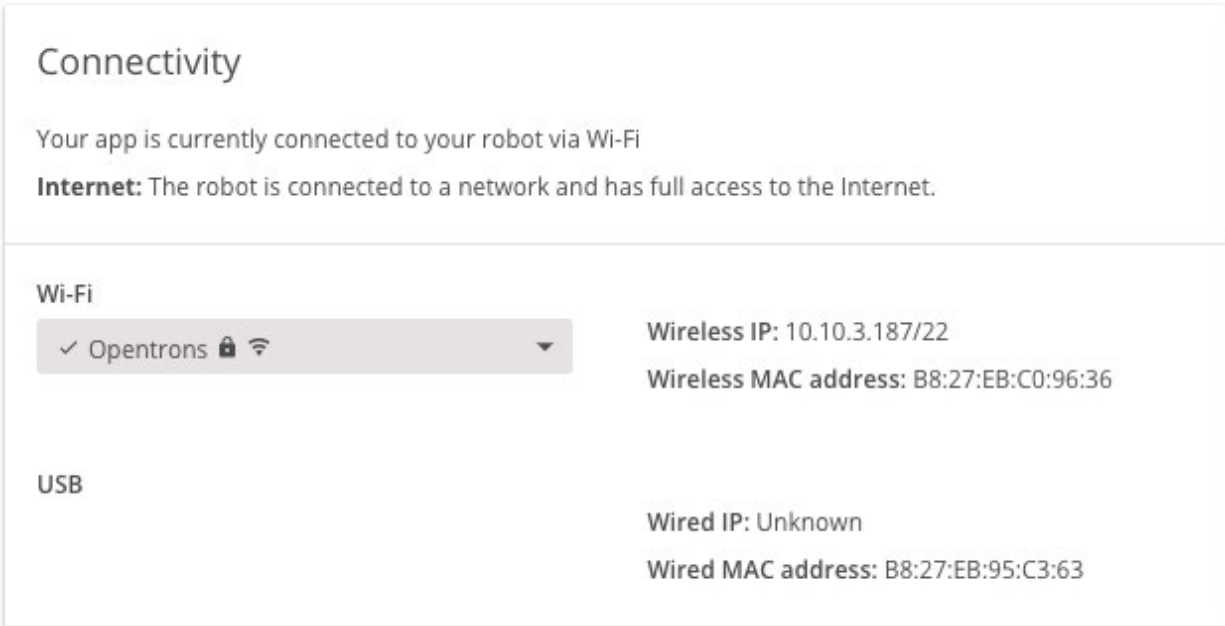
In this image, the robot's IP address is 10.10.3.187
You can then use an application like Postman, wget, or curl to send a POST request to 10.10.3.187:31950/camera/picture.
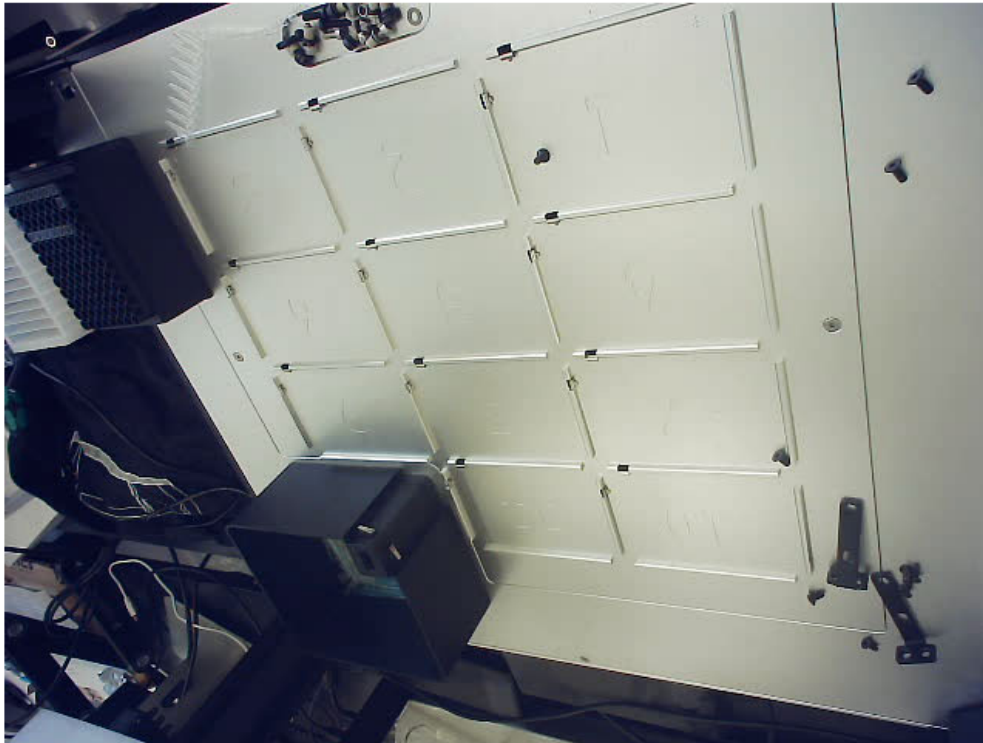
Taking images and video from the robot's command line

Sometimes, taking a still image is not enough, or you want to handle the images from the robot itself. In this case, you can SSH into the robot and use the ffmpeg utility to interact with the camera directly.

1. Connect to your OT-2 with SSH

See Connecting to Your OT-2 With SSH.

2. Invoke ffmpeg

Once you can run commands on the robot, you can use the pre-installed ffmpeg utility to save pictures and video. Full documentation on this interface can be found online.

Example: save an image

ffmpeg -y -f video4linux2 -s 640x480 -i /dev/video0 -ss 0:0:1 -frames 1 /data/user_storage/image.jpg

This will save a 640x480 (-s 640x480) JPEG image (-ss 0:0:1 -frames 1) from the camera (-i /dev/video0) to a file in the /data/user_storage directory named image.jpg .

Example: save a video

ffmpeg -y -video_size 320x240 -i /dev/video0 -t 00:00:01 /data/user_storage/video.mp4
This will save 1 second (-t 00:00:01) of 320x240 video (-video_size 320x240) from the camera (-i /dev/video0) to a file in the /data/user_storage directory named video.mp4 .

Taking images and video from a Python protocol

To integrate the camera into your protocol, use the Python subprocess module to invoke the ffmpeg commands above. For instance, to save an image from the camera from a protocol you might do this:

```
1import subprocess
2
3def run(protocol):
4  if not protocol.is_simulating():
5
6    subprocess.check_call(['ffmpeg', '-y', '-f', 'video4linux2', '-s',
7             '640x480', '-i', '/dev/video0', '-ss', '0:0:1',
8             '-frames', '1', '/data/user_storage/image.jpg'])
9
10    contents = open('/data/user_storage/image.jpg', 'rb').read()
11    protocol.comment('The OT-2 Just took a picture!')
```

 And the image file would be in the variable contents.

Note that the code that takes the picture is wrapped in if not protocol.is_simulating() to avoid taking pictures during protocol simulation.

Downloading image and video files from the robot to your computer

You can use SCP to download the files you just created. See: [Copying Files To and From Your OT-2 With SCP](#).

You can also save your files under the **/var/lib/jupyter/notebooks** directory, which will allow them to be accessed through your robot's Jupyter Notebook.

11.16 Running the robot using Jupyter Notebook

1. [Launching Jupyter Notebook](#)
2. [Creating a Python 3 notebook](#)
3. [Writing and running your protocol](#)
4. [❗Positional calibration in Jupyter Notebook](#)

Launching Jupyter Notebook

Each robot comes pre-installed with Jupyter Notebook, which can be accessed in your preferred web browser.

To launch the Jupyter Notebook server, select **Open** beneath advanced settings in the Opentrons App:
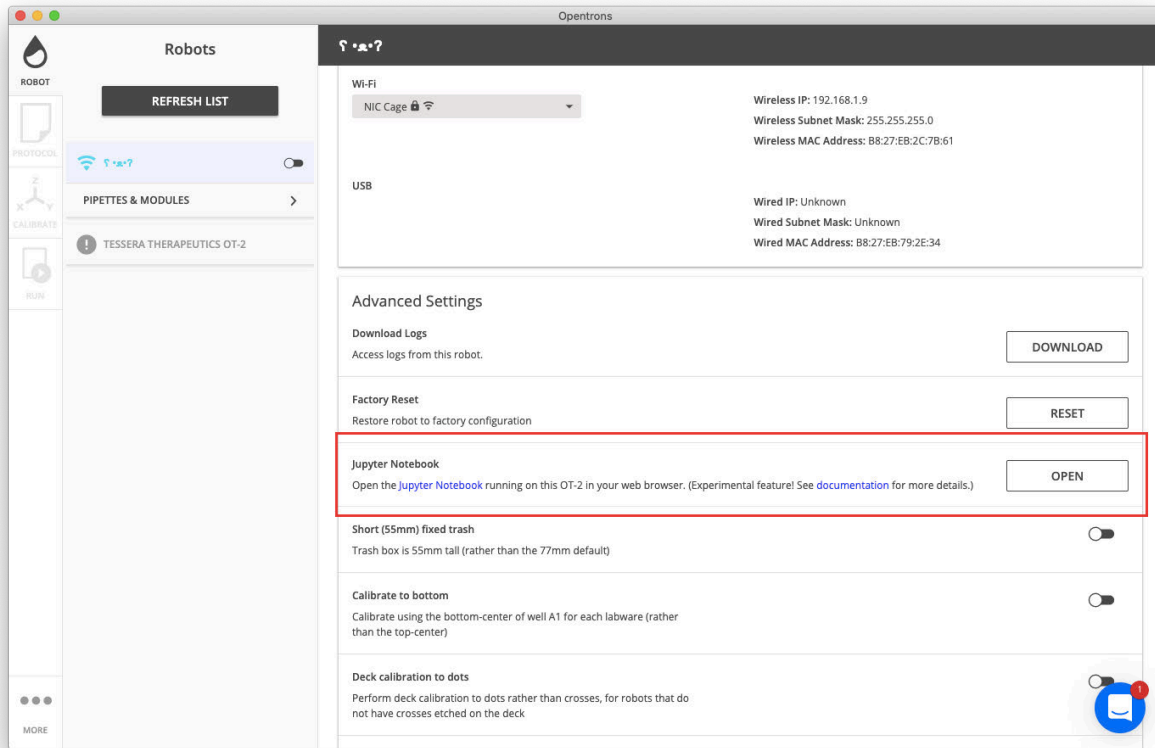
The home page will look like this:
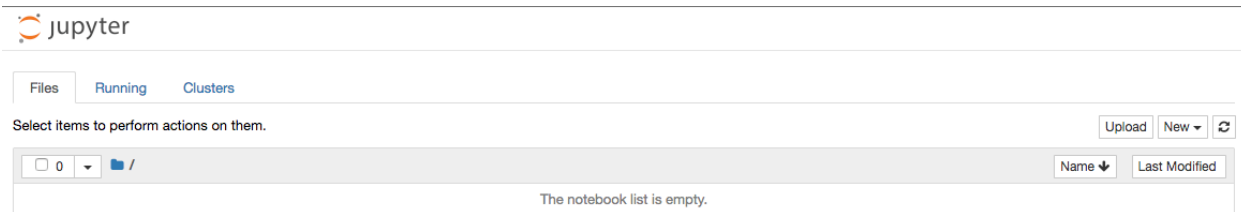
11.17 Create a Python 3 notebook

Once you're inside Jupyter Notebook, you'll need to create a new notebook by selecting **New > Python 3**:
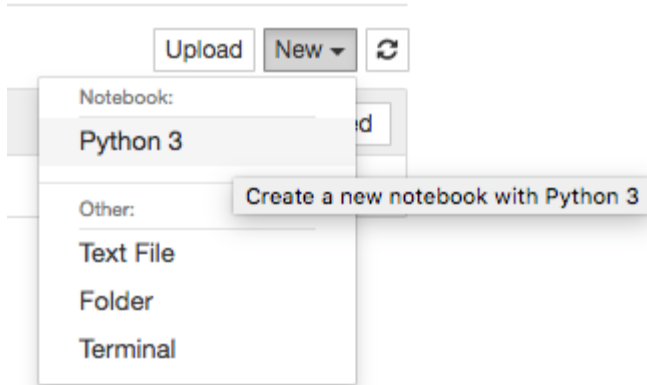
Writing and running your protocol

Once you've created a new notebook, you can enter code into the cells and run them. However, there are some special considerations you'll need to keep in mind when writing code and running the OT-2 via Jupyter Notebook. See our API documentation for more information and examples of how to structure your code to run on Jupyter.

Below you will find example code and commands for writing a test protocol. We recommend first simulating the protocol to make sure the steps are what you expect them to be.

```
1# Import opentrons.execute
2
3import opentrons.execute
4
5
6# This is where you establish the API version
7
8protocol = opentrons.execute.get_protocol_api('2.11')
9
10
11# Must home before running
12
13protocol.home()
14
15
16# Labware
17
18plate =protocol.load_labware('nest_96_wellplate_200ul_flat', '2')
19
20tiprack_1 = protocol.load_labware('opentrons_96_tiprack_1000ul',location = '1')
21
22
23# Pipette
24
```

```
25p1000 = protocol.load_instrument('p1000_single_gen2','right', tip_racks=[tiprack_1])
26
27
28# Protocol steps
29
30p1000.transfer(100, plate['A1'], plate['A2'])
31
32
33# Output steps in protocol
34
35for line in protocol.commands():
36
37    print(line)
```

11.18 Positional calibration in Jupyter Notebook

❗Important: You can't currently execute positional calibration through Jupyter notebook, only through the Opentrons App. Calibration on the OT-2 is generally saved until you explicitly overwrite it. When you move to Jupyter Notebook, it will use the existing calibration from the Opentrons App.

To properly calibrate your protocol in the Run App before running it in Jupyter Notebook, you can create a dummy protocol. This consists of all the pipettes and labware you will be using in your protocol and can be made in the Python API or Protocol Designer. For it to upload properly, make sure to include liquid handling steps in the protocol.

This is an example of a simple dummy protocol in the Python API using a GEN2 single-channel P1000 pipette, 1000 uL tip rack, and 96 well plate:

```
1from opentrons import protocol_api
2
3metadata = {'apiLevel': '2.11'}
4
5
6def run(protocol: protocol_api.ProtocolContext):
7
8    # Labware
9
10    plate = protocol.load_labware('nest_96_wellplate_200ul_flat', '2')
11
12    tiprack_1 = protocol.load_labware('opentrons_96_tiprack_1000ul',location = '1')
13
14
15    # Pipette
16
17    p1000 = protocol.load_instrument('p1000_single_gen2','right', tip_racks=[tiprack_1])
```

```
18
19
20   # Protocol steps
21
22   p1000.transfer(100, plate['A1'], plate['A2'])
```

Screenshot below to reference the proper spacing:

```python
from opentrons import protocol_api

metadata = {'apiLevel': '2.11'}

def run(protocol: protocol_api.ProtocolContext):
    # Labware

    plate =protocol.load_labware('nest_96_wellplate_200ul_flat', '2')

    tiprack_1 = protocol.load_labware('opentrons_96_tiprack_1000ul',location = '1')

    # Pipette

    p1000 = protocol.load_instrument('p1000_single_gen2','right', tip_racks=[tiprack_1])

    # Protocol steps

    p1000.transfer(100, plate['A1'], plate['A2'])
```

11.19 Using fewer than 8 channels of a multichannel pipette

8-Channel pipettes have the ability to use less than all eight channels during liquid handling steps. You can instruct the OT-2 to perform transfers and other liquid handling actions using anywhere from one to all eight channels.

An 8-Channel pipette using this method can remove the need to occupy one of the pipette mounts with a Single Channel pipette that may only be needed for a few actions.

This method is intended to be employed by advanced users with some precautions. Note: This method is not officially supported and to be used at your own risk

1. Due to the fact that the channels of the 8-Channel pipette are mechanically linked, we cannot control the aspiration and dispense actions of individual channels. When aspirating and dispensing, all channels will be performing the same action at the same time.

2. The 8-Channel P300 uses 0.1 Amps per channel, but the 8-Channel P20 would use 0.075.

3. This method makes use of accessing the hardware controller directly, which can prove to be unstable and may not continue to be supported with later robot software updates.

4. When arranging the labware on the deck with the intention of using this method, you'll need to make sure that there are no labware objects taller than the targeted tip rack in the slot behind the tip rack. For example, if you plan to have a multichannel act as a single channel and assign it to use a tip rack in slot 5, labware taller than the tip rack in slot 5 should not be placed above it in slot 8. This is because the 8-Channel pipette is physically wider than a single channel, and some overhang of the pipette body may impact tall labware on tip pickup.

5. Slots 1-3 cannot be used to house a tip rack intended for access by an 8-Channel pipette using this method: the pipette could crash into the front of the OT-2

6. A certain force is applied to the pipette in order to pick up tips. An 8-Channel pipette normally requires 8x the force of a Single-Channel to pick up all eight tips appropriately. In the case of using an 8-Channel to pick up fewer than eight tips, the force value will need to be modified accordingly by adjusting the number_of_tips = n value. Within the code this tip number value will be multiplied by the per_tip_pickup_current = .1 value to produce the correct tip pick up force

Example code for 8-Channel P300 (GEN2):
Change the per_tip_pickup_current from 0.1 to 0.075 if an 8-Channel P20 (GEN2) is being used

```
1
2 def run(ctx):
3
4    tipracks = [ctx.load_labware('opentrons_96_tiprack_300ul', '4')]
5
6    pip = ctx.load_instrument('p300_multi_gen2', 'right')
7
8    num_channels_per_pickup = 1
```

```
9   # (only pickup tips on front-most channel)
10
11   per_tip_pickup_current = .1
12   # (current required for picking up one tip, do not modify unless
13   # you are using a GEN2 P20 8-Channel in which case change it to
14   # 0.075)
15
16   pick_up_current = num_channels_per_pickup*per_tip_pickup_current
17   ctx._implementation._hw_manager.hardware._attached_instruments[
18   pip._implementation.get_mount()
19   ].update_config_item('pick_up_current', pick_up_current)
20   tips_ordered = [
21   tip for rack in tipracks
22      for row in rack.rows()[
23      len(rack.rows())
24      -num_channels_per_pickup::-1*num_channels_per_pickup]
25      for tip in row]
26   tip_count = 0
27   def pick_up(pip):
28      nonlocal tip_count
29      pip.pick_up_tip(tips_ordered[tip_count])
30      tip_count += 1
31   for i in range(len(tips_ordered)):
32      pick_up(pip)
33       # perform some step
34      pip.drop_tip()
```

11.20 Using Python packages in Python API protocols

If you're writing protocols in our Python API, and you want to take advantage of the full power of Python, you might want to import some code from a package outside Python's standard library.

The package could be one of the great packages already made by the Python community, or one you wrote yourself.

Currently, this is an advanced task. You should be comfortable with the command line to do it.

Installing a package from the Python community

1. SSH into your OT-2.
2. Use pip to install the package from PyPI.
3. Restart your OT-2.

Then, you can upload protocols that import and use the package.

Installing a self-written Python package

If you have a set of functions or variables that you use often, you might want to consider creating your own Python package. This will save you the need to rewrite them. All you need to do is import the module into a protocol similar to how you would for a third-party Python package.

We advise using a text editor to create the module on your computer. The OT-2 is limited to command-line editors which makes it difficult to write complex or long modules.

Once the package is written, you will then need to set up the Python module on the OT-2. There are two ways to do this:

1. Build it into a [module](#) and then install it onto the OT-2 the same way you would an external package.
   a. **Note**: You will need to restart your OT-2 after installing a package.
2. Transfer the raw files onto the OT-2 by using [SCP](#) to a specific path and add that path to your import paths in your protocol.

Can I upload protocols comprising multiple Python files through the Opentrons App?

Not currently.

Instead, write your protocol so that it's a single self-contained Python file.

Or, as described above, collect your additional files into a package, and use pip to install it.

Troubleshooting Package Installation

Some of the more sophisticated Python packages, like [NumPy](#), aren't written purely in Python. Parts of them are written in other languages, like C.
You won't be able to install those packages on the OT-2. They require a [compiler](#), which the OT-2 doesn't have. If you try, you'll see an error like this:
1compilation terminated
2error: command 'get' failed with exit status 1
If one of these packages is critical to your workflow, you can submit a request to us to include it on all OT-2s by default, starting with the next software update.

First, please search our [GitHub repository](#) issues to see if someone else has already requested that Python package. If they have, please thumbs-up their issue instead of making a new one.

Otherwise, you can make a new issue to request the package. In the issue, please tell us a little bit about how you plan on using the package and why it's necessary. This will help our software engineers fulfill the request if possible—or suggest an alternative, if there is one.

Be advised that we do not guarantee any lead time on how long it will take to add packages to the OT-2 or that we will be able to include any requested packages in future software releases.

11.21 ModuleNotFoundError

If your Python API protocol uses a third-party Python package, and you see an error like:

1ModuleNotFoundError [line x]: No module named 'example'

It's probably because you're trying to use a package that's not yet installed on your OT-2. Follow the steps above to install it.

11.22 Is the robot capable of loading plates into an imaging system?

No, our current focus is creating high quality, affordable, efficient liquid handling robots. However, we are open source so if you wanted to make such a modification you could, but we would not be able to do it for you.

11.23 How many computers do I need?

It is possible to use one computer to connect to multiple OT-2s. This can be done by disconnecting from the OT-2 via the app and then connecting to a different OT-2 in the app. You can expect a delay of a few minutes when switching connections to a different OT-2.
We highly recommend one computer per OT-2 for the following reasons:

1. It could slow down the overall process of running/testing the samples.
2. It could introduce confusion about which laptop contains the required protocol, labware, and/or other experimentation data files
3. It would limit our ability to troubleshoot more advanced issues. The OT-2 requires a generated key to SSH into and keeping track of the keys would become very confusing.
4. Having one dedicated computer per OT-2 simplifies the connection process and mitigates the risk of potential connection problems.
5. If using a wired connection, you would need to move the computer around the room, switch cables, and connect to each OT-2 separately.

The app is optimized to work with one OT-2 at one time, it is clunky, and there are generally unknown bugs that can come up:

1. The OT-2s could lose its place in the calibration flow when switching between the different OT-2s.
2. Even if your computer has multiple USB ports, there can be weird routing issues that can lead to the inability to connect to the different OT-2s.
3. If you need to use one computer for multiple OT-2s, make sure to complete calibrations, run the protocol, and properly disconnect through the app before switching to a different OT-2. Once you are connected to the different OT-2 you can flick the deck lights to verify you are connected to the correct OT-2.

We do understand you might have limited computer resources. If you do want to operate multiple OT-2s with one computer, you can:

1. Connect your computer to OT-2 #1
2. Upload your protocol
3. Calibrate required pipette and labware
4. Run the protocol
5. Disconnect from OT-2 #1
6. Connect computer to OT-2 #2, flick the deck lights, and repeat steps 2-4
7. Repeat steps 5 and 6 for remaining OT-2s

11.24 Understanding the X/Y/Z coordinates of the OT-2

The following sections are covered in this article:

1. [Working area diagrams](#)
2. [Math for maximum labware heights based on pipette](#)
3. [How to use the X/Y/Z coordinates in a Python protocol](#)

Working area diagrams

Click below for downloadable .pdfs of drawings that lay out the working area of the OT-2 as well as the max height distance a piece of labware can be depending on the pipette.

- **[Opentrons Pipettes - Highest Bounds](#)**
- **[Opentrons Pipettes - XY Bounds](#)**
- **[Opentrons Pipettes - Lowest Bounds](#)**

Math for maximum labware heights based on pipette

- **P1000, single-channel**: 223.46 - 78.3 = **150.46 mm**
- **P300, single-channel**: 202.77 - 51 = **151.77 mm**
- **P20, single-channel**: 183.77 - 31.1 = **152.77 mm**
- **P300, 8-channel**: 192.44 - 51 = **141.44 mm**
- **P20, 8-channel**: 176 - 31.1 = **144.9 mm**

How to use the X/Y/Z coordinates in a Python protocol

1. You can use the code snippet below as a template to evaluate the absolute X/Y/Z coordinates of the OT-2.
   a. To do this, you can either 1) download the .py file **[here](#)** and proceed to step b or 2) copy and paste the following script into a [source-code editor](#) of your choice.

```
1  from opentrons.types import Location, Point
2
3
4  def run(ctx):
5      ''' input x, y, z values here (in mm).
6      # The coordinates are absolute
7      # with reference to the bottom left corner of slot 1 as origin.
8      # x, y, and z can be a float or integer
9      loc = Location(Point(x, y, z), None)
10
```

```
11  #pipette and labware
12  tiprack = ctx.load_labware('opentrons_96_tiprack_20ul', '11')
13  pip = ctx.load_instrument('p20_single_gen2', 'right', tip_racks=[tiprack])
14
15  #commands
16  pip.pick_up_tip()
17  pip.move_to(loc)
```

      b. ❗**Important: Before you can use the script, you will need to modify it by using a source-code editor.**

      c. You will need to include specific X/Y/Z values in addition to the pipette and labware in use where the script's instructions indicate. Note that pip.pick_up_tip() and pip.move_to(loc) are used for moving the instrument and picking up tips. When you are done, save it as a .py file.

      d. Finally, you need to run the script on your OT-2 using the Opentrons App. To do this, open the Opentrons App. After connecting to your OT-2, go to the **Protocol** tab and upload the .py file you modified and saved. Navigate to the **Run** tab and select **Start run**.

11.25 Completed Certifications

ISO 9001, CE, FCC, NRTL & CSA

11.26 Determine the difference between a GEN1 and GEN 2 Pipette
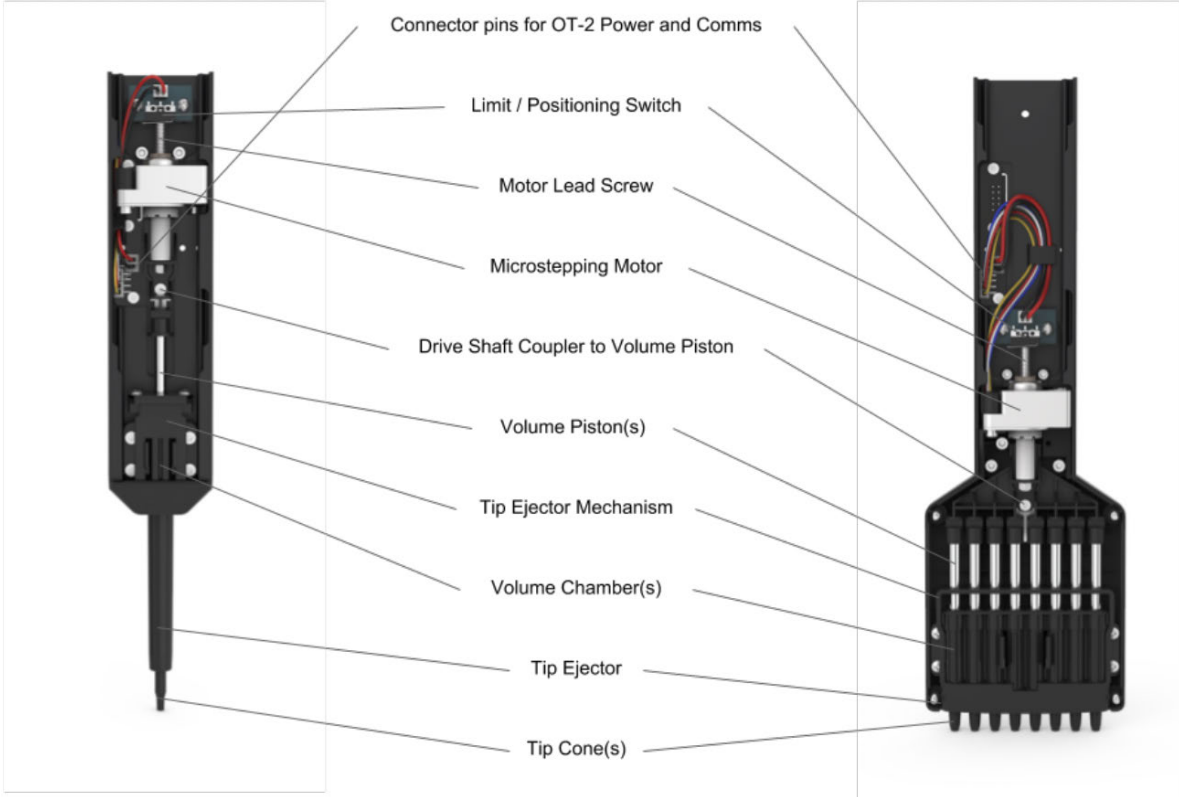
GEN1 pipettes

Single-Channel Electronic GEN1 Pipette   **8-Channel Electronic GEN1 Pipette**

Opentrons GEN1 electronic pipette use a robust stepper motor attached to a precision-ground linear shaft. The shaft creates a seal with an o-ring to create a piston (or volume chamber) which accurately controls the volume of liquid aspirated and dispensed inside the disposable tips, see diagram below. The tip cones are made from a rigid polypropylene plastic designed to pick up and eject 100,000 tips over the lifetime of the pipette. On the body of the GEN1 pipette there is no marking that says "GEN." The GEN1 pipette is shorter than GEN2 pipettes.

Connector pins for OT-2 Power and Comms

Limit / Positioning Switch

Motor Lead Screw

Microstepping Motor

Drive Shaft Coupler to Volume Piston

Volume Piston(s)

Tip Ejector Mechanism

Volume Chamber(s)

Tip Ejector

Tip Cone(s)

GEN1 Pipette design cutaway diagram with parts labeled.

GEN2 pipettes

Single-Channel Electronic GEN2 Pipette   **8-Channel Electronic GEN2 Pipette**

The Opentrons Single-Channel GEN2 pipettes reliably conduct liquid transfers from 1 μL -1,000 μl, while the Opentrons 8-Channel GEN2 pipettes reliably conduct liquid transfers from 1 μL -300 μl. Both have tip pickup and drop +/- 1mm of a successful seal when using Opentrons tips. Opentrons second generation pipette will always be marked with the "GEN2." The GEN2 pipettes are longer than the GEN1 pipettes, see dimensions of pipette below.