

# Before You Start

## Required Tools

---

- [Ruler](#): A precision steel ruler is recommended, in the units you intend to work in.
- [Small Electronics Tweezers](#): Any small tweezers will do. It's worth having a few sets because you will use these a lot.
- [Double Sided Tape](#): You'll use this to hold down your first tape strips and to provide a sticky surface on your PCB to place parts on.
- [Test PCB](#): This directory contains a test PCB you can have made, or any PCB you already have will work.
- [Part Tape Strips](#): Some parts in tape strips to test with. Use something cheap, like an 0805 resistor.

**Note:** Some of the links above are Amazon Affiliate links. If you buy something from one of the links it will help fund development of OpenPnP.

## Machine Working Outside OpenPnP

---

Before you start trying to make your machine work with OpenPnP you should first make sure all of its major functionality is working outside of OpenPnP.

You should be able to jog the machine in X and Y, rotate your nozzle, and move the head up and down in Z. Ideally you will also be able to test that any pumps or solenoids can be controlled.

If your machine uses Gcode, you can use a tool like [Printrun](#) to test the machine.

If you have not yet chosen a motion controller, see [Motion Controllers](#) for some helpful advice.

## Install OpenPnP

---

If you haven't already, [download and install OpenPnP](#) and start it up for the first time. Run through the [Quick Start](#) to get a basic understanding of the interface and the major controls.

## Configuration and Log Locations

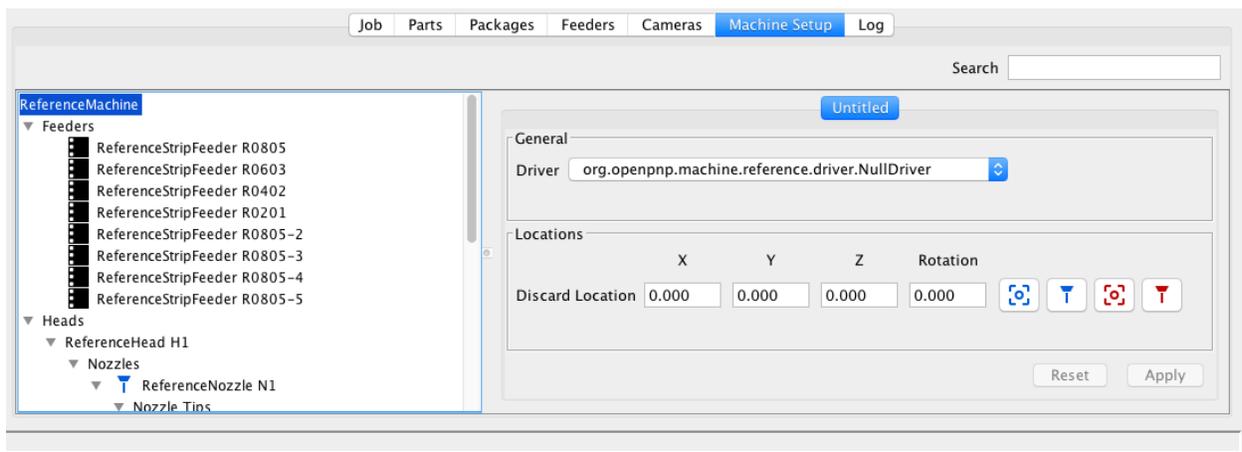
---

Get familiar with the locations of configuration and log files for OpenPnP. You will likely need to edit the `machine.xml` configuration file and you may need to look at the log files if things go wrong.

# Driver Setup

In OpenPnP, the Driver is the part of the software that interfaces between OpenPnP and a particular type of machine. Typically this is just a small piece of code that translates OpenPnP commands into commands for a particular motion controller such as Smoothie, TinyG, Marlin, etc.

## Choosing a Driver



1. Go to the Machine Setup tab and select the root node of the tree. On most setups it's called "ReferenceMachine". A setup panel will appear on the right.
2. In the setup panel, select the driver that most closely matches your motion controller or machine. Most machines should use the [GcodeDriver](#). Click apply.

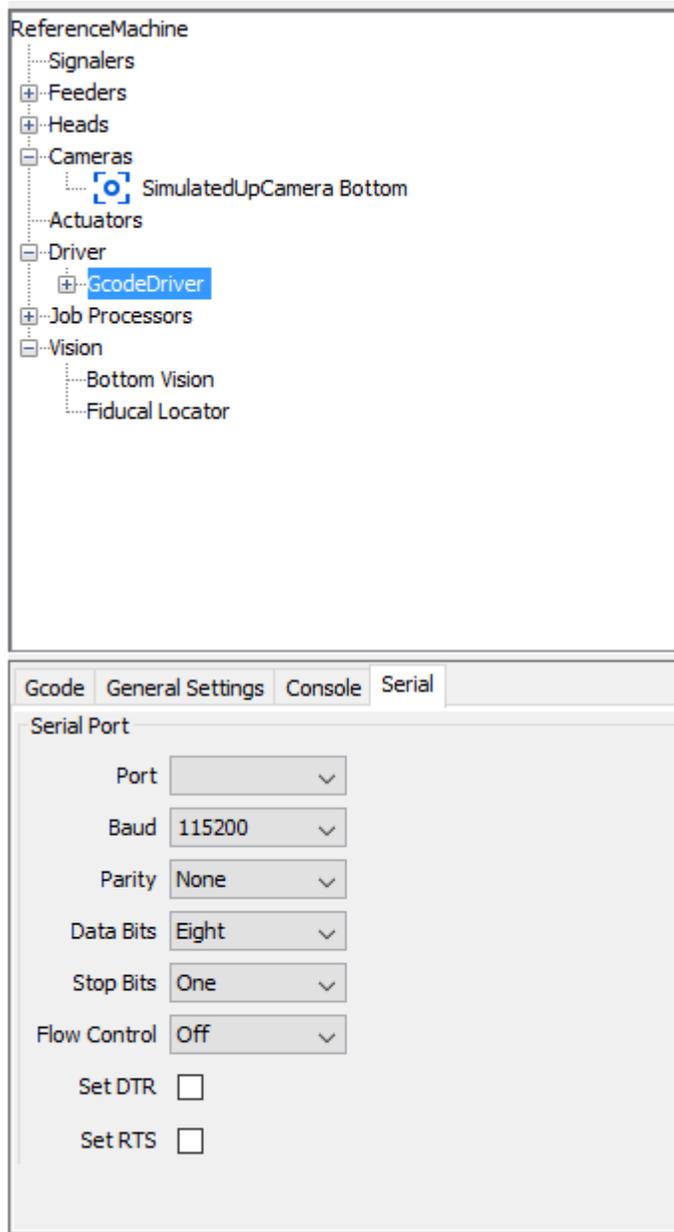
**\*\*Note:** GcodeDriver is a universal driver that works with many motion controllers. It can be used for TinyG, Smoothie, Marlin, Grbl, etc. You should use GcodeDriver unless you have very specific driver requirements.

3. OpenPnP will prompt you to restart the program, so do that.

## Set Serial Port and Baud rate

---

Most of the drivers in OpenPnP communicate using the serial port. Before you can connect, you need to set the serial port and baud rate for your controller.



1. After restarting OpenPnP go back to the Machine Setup tab and find the Driver you selected in the tree. It should be near the bottom, under the Driver branch. Select it and a setup panel will appear.
2. Select the serial port and baud rate for your controller and press the Apply button.

## Driver Specific Setup

---

See the pages below for additional information on setting up specific drivers within OpenPnP.

- [GcodeDriver](#): Recommended for all Gcode based controllers such as Smoothie, Grbl, TinyG, Marlin, etc.

## Connect

---

Now that the driver is configured, press the green power button to connect to your controller. If all goes well the button will turn red and the rest of the controls will become enabled. If there is an issue OpenPnP will give you an error message.

## Homing

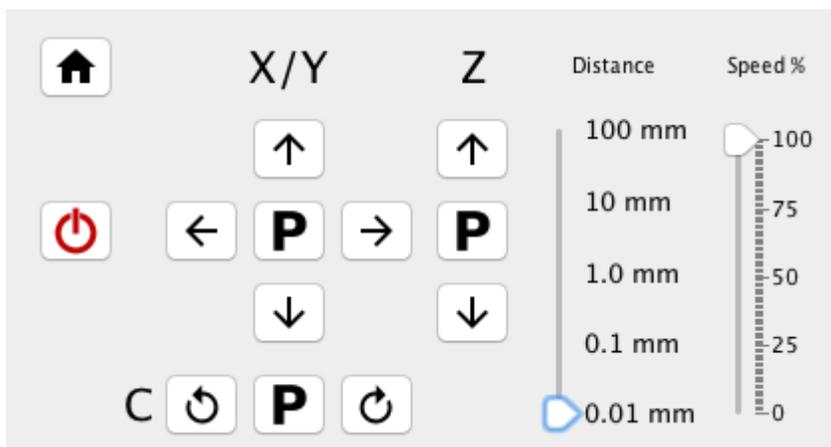
---

If your machine connected successfully it's time to home it. If you don't have home switches installed you can skip this step. To home the machine click the home

button .

## Jogging the Machine

---



With the machine homed, you can now try jogging the machine to make sure everything is working well. Set the Distance slider to 1mm and click the jog buttons to move the head around. Make sure that the machine moves in the directions specified by the buttons. If it doesn't, check your controller configuration.

# Top Camera Setup

---

The Top Camera is the camera that is attached to your machine's head. It's also sometimes referred to as the Downlooking Camera or Fiducial Camera. They all mean the same thing.

The Top Camera is used primarily when setting up feeders and jobs, and can be used to target any location on your machine. It's the thing you will look at most often when using OpenPnP.

## Add a Camera

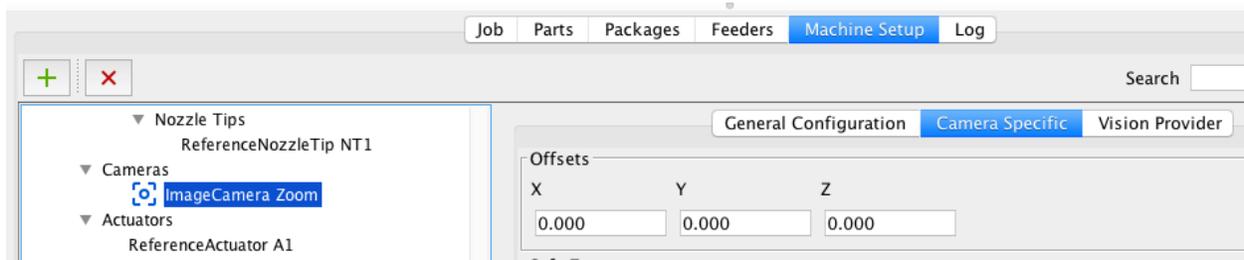
---

First, add a head camera by following the instructions in [General Camera Setup](#). Make sure to configure each section before returning to this page to continue.

## Head Offsets

---

Camera Head Offsets tell OpenPnP where the camera is in relation to the other objects on the head, such as nozzles. It's best to use the primary camera as the origin for the head, so we set the offset to all zeros.



1. With the same camera still selected in the Machine Setup tab, look to the right for the configuration tabs.
2. Select the tab called "Camera Specific" and you should see a section called Offsets at the top.
3. Enter 0.0 for each of the X, Y and Z fields.
4. Press Apply.

## Camera Jogging

---

Now that the camera is configured, click and drag anywhere in the camera view. You'll see a white line follow the mouse cursor. When you release the mouse button the camera should move to the location you were pointing at. This is called Camera Jogging and is the easiest way to move the machine around when setting things up. You can also hold Shift and then Left Click within the camera view to do the same thing.

---

# Steps Per Mm

---

Now that you can move the machine around and you have a camera it's a good time to check that your controller is moving the right amount when you tell it to. This is often part of your "steps per mm" configuration in most controllers. The way that you set this will be dependent on your controller, and you should check the instructions for the controller for more information.

## Testing Steps Per Mm

---

To test that your steps per mm settings are correct:

1. Place a [ruler](#) on the bed of the machine, along the X axis, so that it is visible to the camera. The ruler should be at least 100mm in length. The closer it is to the length of your axes the better your accuracy will be.
  2. Jog the camera so that it is focused on the edge of the ruler.
  3. Jog the camera back and forth along the length of the ruler and adjust the ruler on the bed until the camera crosshair tracks the edge perfectly.
  4. Align the camera crosshair with one of the millimeter lines on the left of the ruler.
  5. Jog 100mm to the right.
  6. Check if the crosshairs now line up with the millimeter marking 100mm from where you started. If they do not, you will need to adjust your steps per mm setting. If the crosshairs went past 100mm then your steps per mm setting is too high. If they did not get all the way to the 100mm marking then the setting is too low.
  7. When your X axis is adjusted correctly, perform the same test with the Y axis by turning the ruler 90 degrees.
-

# Nozzle Setup

## Concepts

---

In OpenPnP we use a couple names for things that might not be familiar. These will become important down the road, so it's good to have an understanding of them now.

## Heads

In OpenPnP, the Head is the part of the machine that actually moves. The Head can have many things attached to it, and these may move independently, typically in Z. A common machine might have 1 Head, 1 Head mounted Camera and 2 Nozzles.

## Nozzles and Nozzle Tips

In OpenPnP, the thing that actually touches parts is called a Nozzle Tip. A Nozzle Tip is attached to a Nozzle and a Nozzle is attached to a Head. The reason for this distinction is that OpenPnP supports the ability to automatically change Nozzle Tips depending on the size of the parts you are picking up. The Nozzle stays the same, the Nozzle Tip is changed.

You may have any number of Nozzles on a Head, and any number of Nozzle Tips assigned to a Nozzle.

## Adding Nozzles

---

If you have more than one nozzle on your machine:

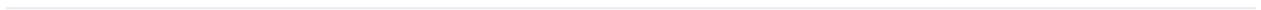
1. Open the Machine Setup Tab.
2. Find the head you want to add a nozzle to and look for the Nozzles item below it. Select it.
3. Add the nozzle by pressing the add nozzle button .
4. Select a nozzle type from the dialog. ReferenceNozzle is good for most setups.
5. Press Accept and the new nozzle will appear in the list. Select it to open it's properties.

## Adding Nozzle Tips

---

If you have more than one nozzle tip on your machine:

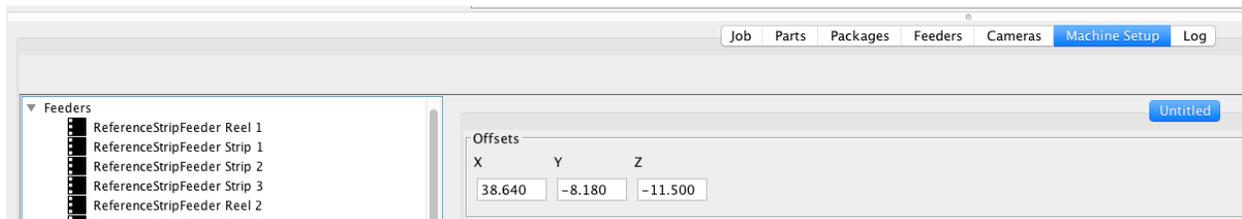
1. Open the Machine Setup Tab.
2. Find the nozzle you want to add a nozzle tip to and look for the Nozzle Tips item below it. Select it.
3. Add the nozzle tip by pressing the add button  .
4. Select the new nozzle tip to open it's properties.
5. To use the machine with the nozzle tip you just created you need to load it onto the nozzle. Press the load button  to do this.



## Head Offsets

---

Nozzle Head Offsets tell OpenPnP where your nozzles are in relation to the other objects on your head. In an earlier step we set the Camera Head Offsets to zero, making the top camera the origin of the head. Now we'll tell OpenPnP how far the nozzle is from the camera in real world units.

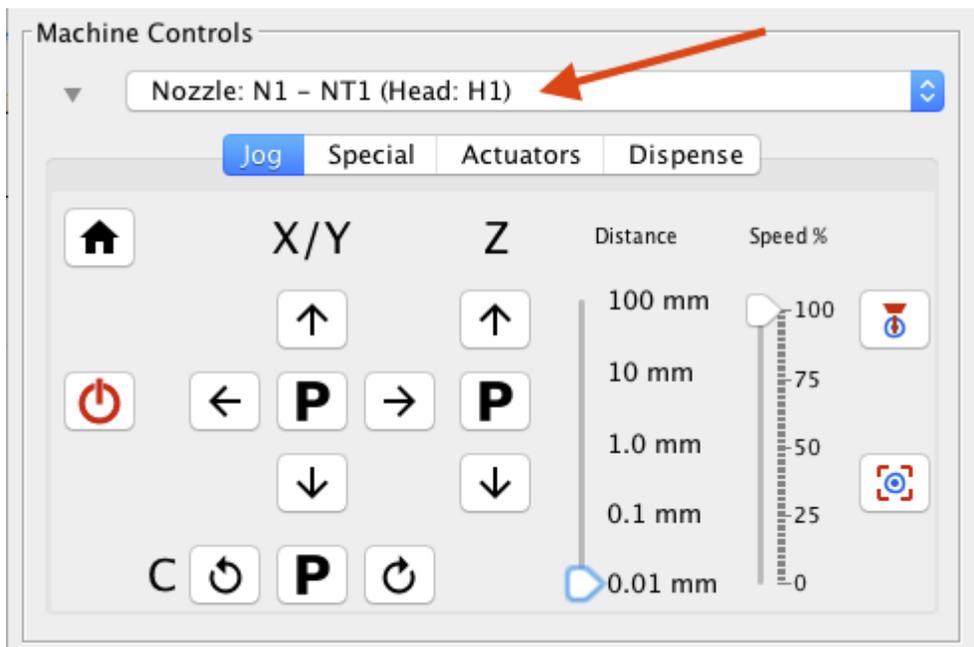


1. Place something on the bed of the machine that can be marked by the nozzle. A piece of a small, flattened blob of putty or clay will work. [Silly Putty](#) works perfectly. This is our target.
2. Jog the machine so that the primary Nozzle is over the target and then lower the nozzle until it makes a clear mark on the target.
3. Click the X, Y and Z, DRO one time each. They will turn blue and show 0.000. They are now in relative coordinate mode and will show the distance you have moved since clicking them until you click them again.
4. Jog the machine so that the top camera is over the mark on the target, perfectly centered and in focus.
5. Find the Nozzle in the Machine Setup tab and find the Offsets fields in the panel on the right. It's on the second tab titled "Untitled".
6. Set the offsets to the X, Y and Z shown in the DROs. Press Apply. Note: For many machines it's not necessary to set the Z offset. It's only used in more complex setups.
7. For each additional Nozzle you need to setup, follow this same process.

## Selecting a Nozzle

---

If you have multiple nozzles on your machine, you can select the one you want to work with in OpenPnP by selecting it from the tools dropdown in the Machine Controls panel. The nozzle that is selected here is the one that will be used whenever you tell the machine to move a nozzle.



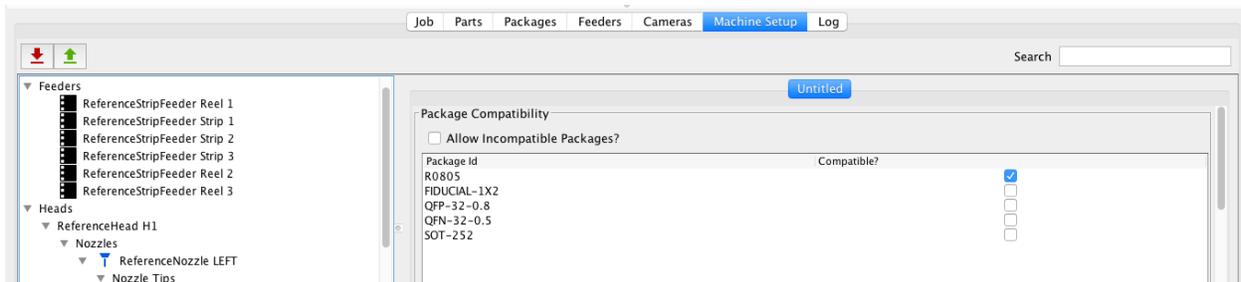
## Advanced

---

On more advanced setups, especially those with multiple nozzles and nozzle tips, you may need to setup package compatibility and the nozzle tip changer.

## Package Compatibility

Package Compatibility lets OpenPnP know which Nozzle Tips you want to use for each package that you'll place. If you want to use a Nozzle Tip for every type of package you can just select "Allow Incompatible Packages?" and OpenPnP will consider that nozzle tip to be available for all parts. If you want to limit the parts that a particular nozzle tip works with, just uncheck the "Allow Incompatible Packages?" checkbox and check the packages you want to enable in the table below. Don't forget to hit Apply.



## Nozzle Tip Changer

The Nozzle Tip Changer allows OpenPnP to automatically choose the best nozzle tip for a given part. Currently the only supported style of Nozzle Tip Changer is the so called "Four Position" changer. You can configure four motions that insert a nozzle tip. To return a nozzle the four position moves are executed in reverse before picking up a new one. Note that the *First Location* is always approached from Z0 above and the last move is always returning from *Last Location* to Z0 so the four moves are actually six moves.

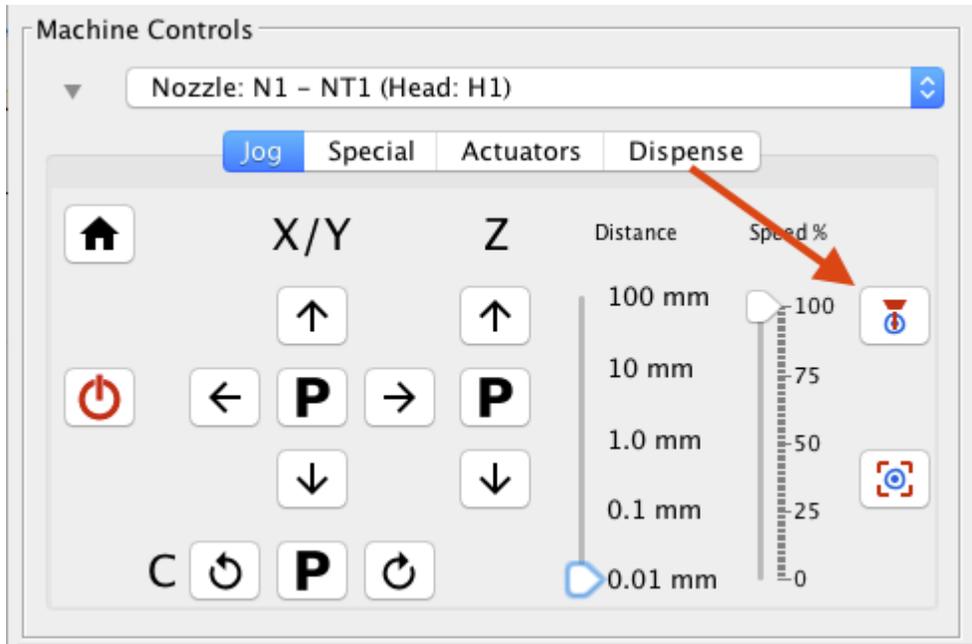
TODO: This is an advanced topic and further documentation is needed.



## Nozzle Jogging

---

Now that your nozzle is configured, let's move it around and make sure it's working.



1. Make sure you've selected the nozzle as discussed in Selecting a Nozzle above.
2. Move the camera to an identifiable spot on the machine bed.
3. Click the position tool button  in the Machine Controls panel.
4. The nozzle should move to where the camera was previously looking. You can click the position camera button  below to move the Camera back.

The Position Tool and Position Camera buttons can be used to switch back and forth between what the camera is pointing at and what the nozzle is pointing at.

---

# Actuators

---

Actuators are generic devices you can attach to your machine to do additional tasks not covered by Nozzles and Cameras. For instance, you can use a solenoid with a pin attached to the head to drag a tape forward to feed it. You can also use Actuators to read values from sensors attached to the machine.

Some of the things OpenPnP uses Actuators for are:

- [Auto feeder triggers](#).
- [Nozzle vacuum pressure sensing](#).
- Drag feeder solenoid pins.
- [Camera lighting control](#).

Using the [Scripting](#) system you can use Actuators to extend OpenPnP to control a wide array of devices and sensors.

## Adding Actuators

---

1. Open the Machine Setup tab.
  2.
    - If you are adding a head mounted actuator, find the head in the tree on the left. Under the head look for Actuators and select it. Head mounted actuators are often attached to devices such as drag feed solenoids and nozzle change tools.
    - If you are adding a machine mounted actuator, find Actuators under the root of the tree and select it. Machine mounted actuators can be used for things like conveyors and lighting.
  3. Add an actuator by pressing the green plus button .
  4. Select a actuator driver from the provided list and press the "Accept" button. The newly added actuator will show up in the actuators list.
  5. Click on the name of the new actuator to open it's properties.
-

# GcodeDriver

---

See [actuate-boolean-command](#)

See [actuate-double-command](#)

See [actuator-read-command](#)

See [actuator-read-regex](#)

## Head Offsets

---

See [Setting Head Offsets](#) for the general process. It is basically the same for Actuators.

---

# Bottom Camera Setup

The Bottom Camera is a upward looking camera fixed to the machine, and not the head. The Bottom Camera is used during Bottom Vision processing to more accurately place parts. It's not required to have a Bottom Camera, but you will generally get better results if you do have one.

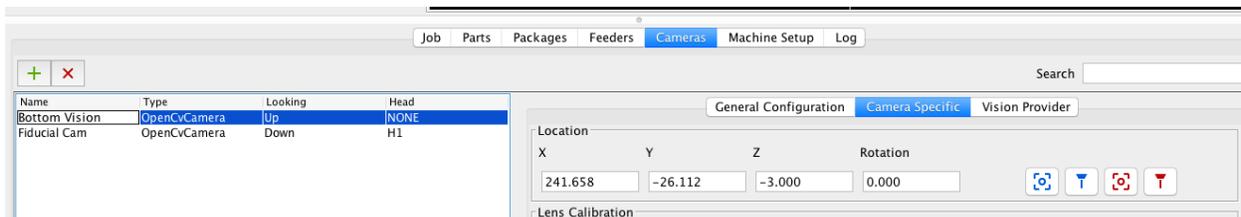
## Add a Camera

First, add a camera by following the instructions in [General Camera Setup](#). Make sure to configure each section before returning to this page to continue.

Many Bottom Cameras tend to use much shorter fisheye lenses, so pay careful attention to the Lens Calibration section in the document above. It's likely to be more important for the Bottom Camera than the Top Camera.

## Set Camera Location

OpenPnP needs to know where on the machine the bottom camera is.



1. Open the Cameras tab and select the Bottom Camera.
2. Select the Camera Specific configuration tab on the right.
3. Scroll down to the Location group.
4. Jog the primary nozzle so that it is perfectly centered and in focus over the camera.
5. Click the capture tool location button **T** to grab the location of the nozzle, thereby setting the camera location.
6. Press Apply to save your changes.

## Move Nozzle to Camera

---

Now that OpenPnP knows where the camera is, let's move the nozzle around and make sure everything looks right.

1. Hold Shift and then Left Click the mouse in the camera view for the bottom camera.
  2. The nozzle will move to the camera and should land where you clicked.
  3. Using the jog controls, move the nozzle forward, backward, left and right while looking at the camera view.
  4. Make sure that when you click arrows to move nozzle, in the camera view the nozzle moves in the same direction. For example if you click the up arrow nozzle should move vertical on your screen. If it doesn't, you may need to adjust the [rotation and transforms](#) of the camera.
-

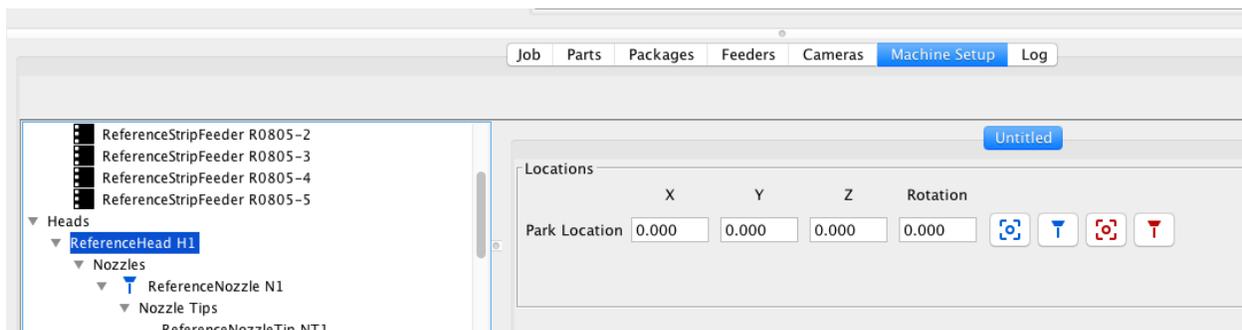
# Park Location

---

The Park Location is the location that your head will move to when commanded to park, and at the end of a job. This should be a safe spot that is out of the way so that when a job is complete you can access the machine to remove your finished board.

## Set Park Location

---



1. Open the Machine Setup tab and find your Head in the tree. It should be called ReferenceHead.
2. Select the head and look for the settings panel on the right.
3. Jog the nozzle to the location you want the machine to park when it's finished a job.
4. Click the capture nozzle location button **T** to capture the location.
5. Press Apply to save your changes.

## Test Park Location

---

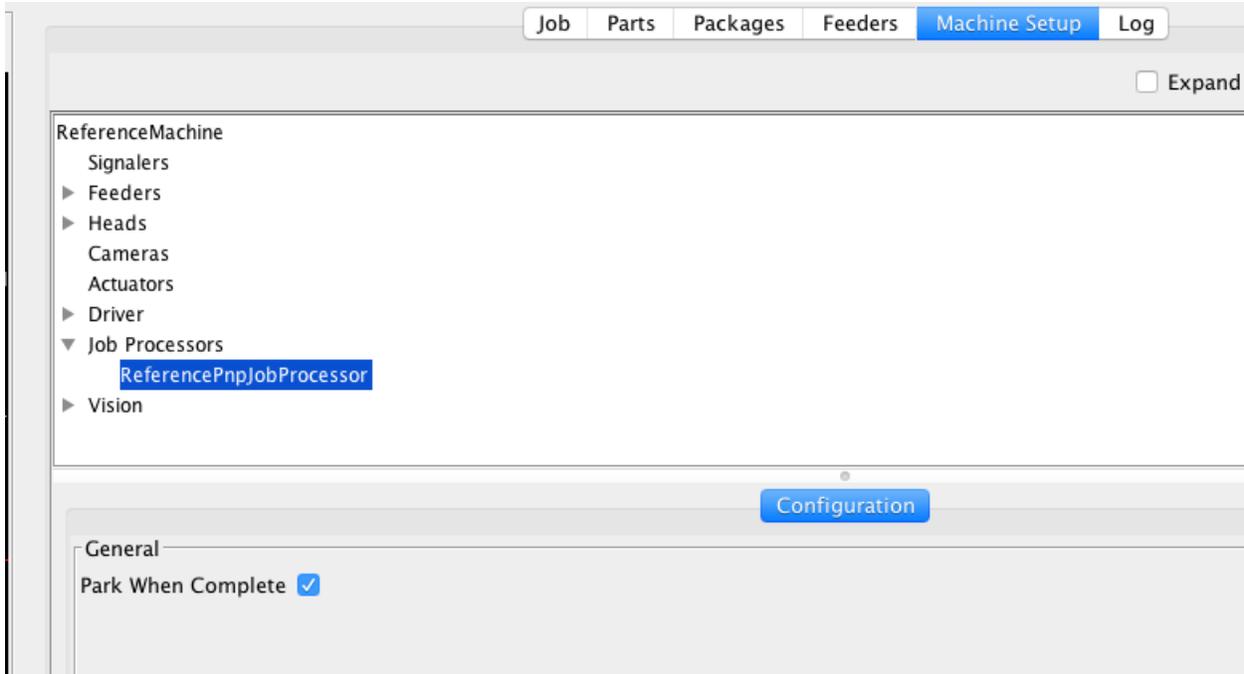
1. Jog the nozzle away from the location it was left at in the previous step.
2. Click the X/Y Park button **P** in the Jog Controls and see that the nozzle moves to the Park location.

## Enable Park When Job Completes

---

If you'd like the machine to park itself when a job completes, you have to turn that option on.

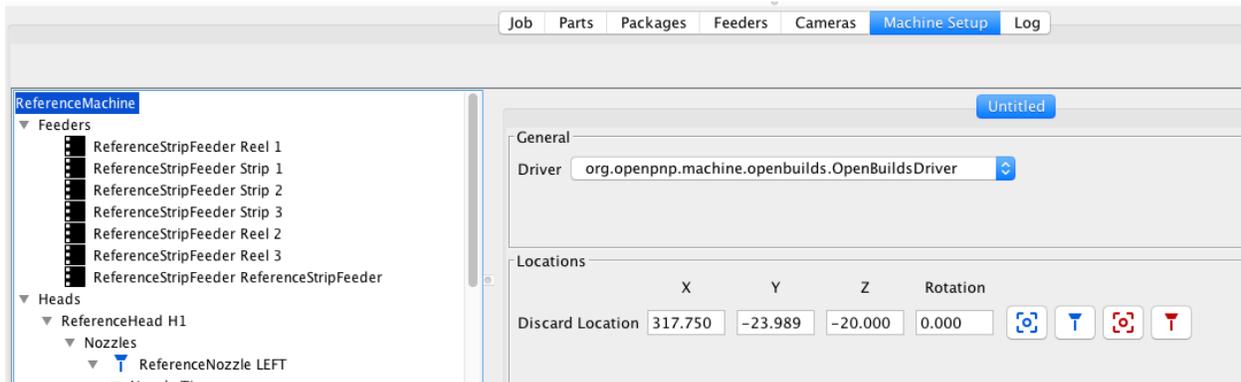
1. Open Machine Setup -> JobProcessors -> ReferencePnpJobProcessor.
2. Toggle the Park When Complete checkbox.
3. Press Apply.



# Discard Location

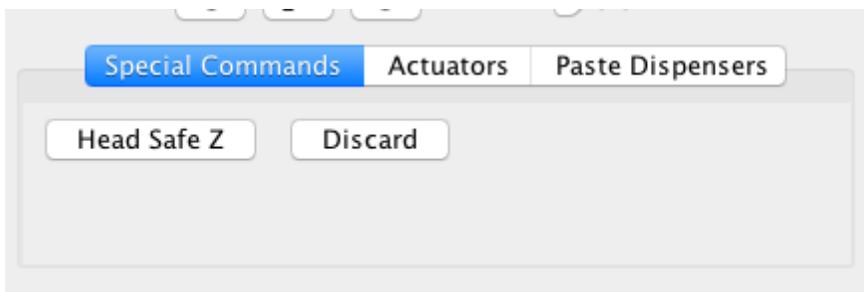
The Discard Location is where OpenPnP will go to drop parts that it fails to place. This can be as simple as a cup taped to your machine. The machine will discard parts when commanded to, and when a pick or bottom vision fails.

## Set Discard Location



1. Open the Machine Setup tab and select the root node. This is usually called ReferenceMachine.
2. Find the Locations section in the setup screen on the right and locate the Discard Location fields.
3. Jog the nozzle so that it is centered over the area you want to discard parts to.
4. Lower the nozzle down as far as you'd like it to go before dropping a part. You can leave this at zero if your discard bin is high enough, or lower it to make sure the parts will definitely fall in the bin.
5. Click the capture nozzle location button  to set the X, Y, Z and Rotation fields.
6. Click Apply to save your changes.

## Test Discard Location



With the discard location set, we can test that it works properly. We haven't setup a feeder yet, so we won't actually discard a part but by clicking the Discard button in the Special Commands tab we can make sure the nozzle moves to the right place. You should also hear your air solenoids click, if you have them.

---

# Feeders

---

Feeders are sources of parts for OpenPnP to place. A feeder can be as simple as a strip of parts taped to the machine or as complex as a fully automated motorized feeder.

There are several different types of feeders available in OpenPnP: Strip Feeder, Drag Feeder, Tray Feeder, Tube Feeder, Auto Feeder and Slot Auto Feeder. The differences between each type will be covered below.

## Concepts

---

### Pick Location

Every feeder has a pick location, which is where the nozzle will go when it needs to pick a part from that feeder. Some feeders change their pick location over time; for instance, a Strip Feeder will tell OpenPnP that the pick location has moved down the tape after each pick.

### Setting Z

It's important to set the right Z height for the feeder so that the nozzle just barely touches the part before it picks. Each feeder has a Z setting in its configuration panel, although some of them look different than others. In all cases, just look for the Z field in the settings and make sure it's set to where the nozzle needs to go to pick the part.

### Choosing a Feeder

---

Here is a short description of the feeder types supported by OpenPnP:

- **ReferenceStripFeeder:** A feeder that allows the user to place a cut piece of SMT tape on the machine. The feeder will advance along the tape and pick parts from it. Cover film must be removed manually. Supports vision for setup and part centering by referencing the holes in the tape.

See [ReferenceStripFeeder](#)'s page for more information and help.

- **ReferenceDragFeeder:** A "drag" feeder which allows the use of an Actuator, typically a solenoid with a pin, to advance the tape by dragging it. Basic vision for part center detection is also supported.
- **ReferenceTrayFeeder:** Supports 2D arrays of parts in trays. Currently limited to trays that are aligned at 90 degrees in X and Y. Simple incremental pick, no vision.
- **ReferenceTubeFeeder:** The simplest feeder which picks from the same location every time. Intended to be used with a vibratory tube feeder that presents a part at the same location repeatedly.
- **ReferenceAutoFeeder:** A basic auto feeder controller for feeders that feed on their own using hardware. Uses a number of Actuators to perform feed operations.

See [ReferenceAutoFeeder](#)'s page for more information and help.

- **ReferenceSlotAutoFeeder:** A feeder slot system masquerading as a regular feeder. Using ReferenceSlotAutoFeeder you can configure slots on advanced feeder systems so that you can easily move feeders from slot to slot. Similar to the ReferenceAutoFeeder, it uses Actuators to trigger feeders.

See [ReferenceSlotAutoFeeder](#)'s page for more information and help.

## Adding a Feeder

---

Let's add a Strip Feeder to the machine so that we can test the feeder system.

1. Use your [Double Sided Tape](#) to tape a small [Part Tape Strip](#) to the bed of the machine. You can cut off a small section of 10 or so parts. These parts will be wasted, so don't use anything important. Chip resistors are good for this.
2. Once securely taped down, remove the cover film from the cut strip using [Tweezers](#).
3. Open the Feeders tab and click the green plus button  to open the add feeder dialog.
4. Select ReferenceStripFeeder from the dialog and press Accept.
5. The new feeder is added to the table below and selected automatically. Double click the Name field to give it a descriptive name. Let's call this one "Test Feeder".
6. On the right side of the screen, in the feeder settings, find the Part dropdown and select the "R0805-1K" part. It's okay if this doesn't match the part you are using for now.
7. In the Tape Settings section below, fill in the Tape Width and Part Pitch fields. If you are using chip resistors larger than 0402 you can use the default settings of 8mm tape width and 4mm part pitch.

8. Click the Auto Setup button and follow the on screen instructions to finish setting up the feeder. There is more information on the [ReferenceStripFeeder](#) page about this process, and it's worth taking a look at it to help you.
9. When the Auto Setup process finishes the camera should be centered perfectly over the first part in the tape. If it's not, try the Auto Setup process again. Sometimes lighting conditions can cause it to make mistakes.
10. Jog the nozzle so that it is just touching the first part in the tape.
11. Fill in the value shown in the Z DRO in the Z fields in the strip feeder setup. This will set the height of the feeder so that OpenPnP knows where to pick parts.

## Common Settings

---

### Part Selection

Every feeder has a Part field that you must fill in. By telling OpenPnP which part a feeder is holding OpenPnP can make intelligent decisions about which feeders to use for a job. You can even have multiple feeders for the same part, in case it's a part you use many of.

### Locate a part

---

To test your settings for a feeder you can make the camera or the nozzle go to the location of the part that's going to be fed next. To position the camera on the part, click the  button. To position the nozzle, click the  button.

### Pick a Part

---

Now that you've configured a feeder, click the Feed and Pick button  to perform a pick operation. If all goes well the machine should feed a part and pick it up with the nozzle. You can use the Discard button we set up previously to drop the part.

---

# Bottom Vision

---

Bottom Vision allows OpenPnP to more precisely place parts by checking their offset and rotation on the nozzle after picking. It requires a Bottom Camera to be configured and there are some extra steps for making sure you get the best results.

Bottom Vision is a large and complex topic, and has it's own section of the Wiki. See [Bottom Vision](#) for more details.

You can use OpenPnP's [Actuator](#) system along with [Scripting Events](#) to turn lights on and off in response to camera capture events.

## Set Up a Lighting Actuator

---

1. Add an actuator for the lights you want to control. Select Machine Setup -> ReferenceMachine -> Actuators and then click the green + button to add a new Actuator.
2. Expand the Actuators list and select the new Actuator.
3. Change the Actuator's name to "UpCamLights" and press Apply.
4. Go to Machine Setup -> ReferenceMachine -> Driver -> GcodeDriver -> Gcode.
5. In the Head Mountable dropdown, select UpCamLights.
6. In the Setting dropdown select ACTUATE\_BOOLEAN\_COMMAND.
7. In the text area, set the Gcode to the commands that will turn your lights on and off. For example, to send M800 for on and M801 for off, use: {True:M800}{False:M801}. You should change the two commands to whatever works for your machine.
8. Press the Apply button.
9. Restart OpenPnP.

## Test The Lights

---

1. Go to Main Window -> Machine Controls -> Actuators and click the UpCamLights button. Your lights should come on. Click it again and they should turn off. If this doesn't work, adjust your Gcode settings until it does, or ask for help.

## Add Lighting Control Scripts

---

1. Download the [zipfile containing the scripts](#) to your desktop or somewhere else you can easily find it.
2. Expand the zipfile. There should be two scripts inside.
3. Go to Main Menu -> Scripts -> Open Scripts Directory.
4. In the folder that opens, open the Events directory.
5. Copy the two scripts into the Events directory.

Now, whenever OpenPnP needs to capture an image from your Up looking camera it will run the Camera.BeforeCapture.js script, capture an image and then run the

Camera.AfterCapture.js script. The first one will turn the lights on and the second will turn the lights off.

You can customize the way this works by modifying the scripts or the Gcode that controls the lights.

If you want to control the Down camera lights, just add another Actuator called DownCamLights. The script is already set up to use it.