

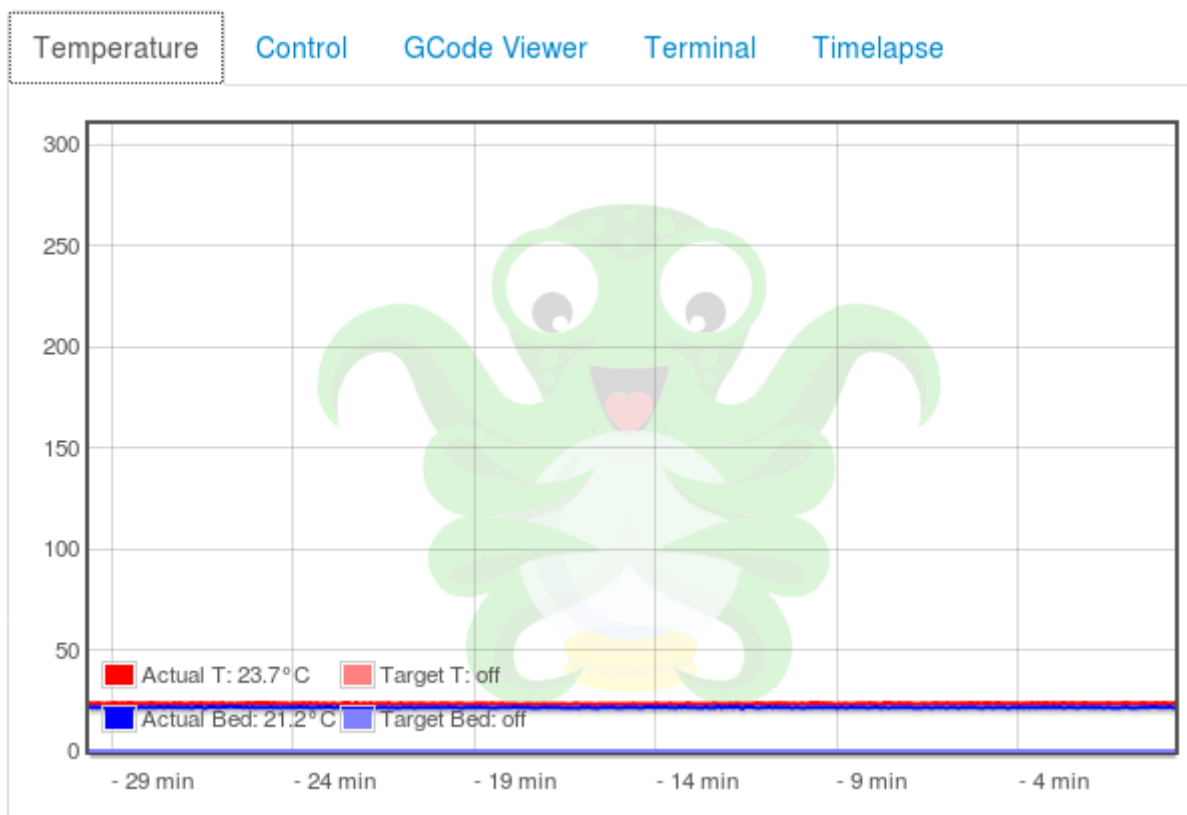
Configuration checks

This document provides a list of steps to help confirm the pin settings in the Klipper printer.cfg file. It is a good idea to run through these steps after following the steps in the [installation document](#).

During this guide, it may be necessary to make changes to the Klipper config file. Be sure to issue a RESTART command after every change to the config file to ensure that the change takes effect (type "restart" in the Octoprint terminal tab and then click "Send"). It's also a good idea to issue a STATUS command after every RESTART to verify that the config file is successfully loaded.

Verify temperature

Start by verifying that temperatures are being properly reported. Navigate to the Octoprint temperature tab.



Verify that the temperature of the nozzle and bed (if applicable) are present and not increasing. If it is increasing, remove power from the printer. If the temperatures are not accurate, review the "sensor_type" and "sensor_pin" settings for the nozzle and/or bed.

Verify M112

Navigate to the Octoprint terminal tab and issue an M112 command in the terminal box. This command requests Klipper to go into a "shutdown" state. It will cause Octoprint to disconnect from Klipper - navigate to the Connection area and click on "Connect" to cause Octoprint to reconnect. Then navigate to the Octoprint temperature tab and verify that temperatures continue to update and the temperatures are not increasing. If temperatures are increasing, remove power from the printer.

The M112 command causes Klipper to go into a "shutdown" state. To clear this state, issue a FIRMWARE_RESTART command in the Octoprint terminal tab.

Verify heaters

Navigate to the Octoprint temperature tab and type in 50 followed by enter in the "Tool" temperature box. The extruder temperature in the graph should start to increase (within about 30 seconds or so). Then go to the "Tool" temperature drop-down box and select "Off". After several minutes the temperature should start to return to its initial room temperature value. If the temperature does not increase then verify the "heater_pin" setting in the config.

If the printer has a heated bed then perform the above test again with the bed.

Verify stepper motor enable pin

Verify that all of the printer axes can manually move freely (the stepper motors are disabled). If not, issue an M84 command to disable the motors. If any of the axes still can not move freely, then verify the stepper "enable_pin" configuration for the given axis. On most commodity stepper motor drivers, the motor enable pin is "active low" and therefore the enable pin should have a "!" before the pin (for example, "enable_pin: !ar38").

Verify endstops

Manually move all the printer axes so that none of them are in contact with an endstop. Send a QUERY_ENDSTOPS command via the Octoprint terminal tab. It should respond with the current state of all of the configured endstops and they should all report a state of "open". For each of the endstops, rerun the QUERY_ENDSTOPS command while manually triggering the endstop. The QUERY_ENDSTOPS command should report the endstop as "TRIGGERED".

If the endstop appears inverted (it reports "open" when triggered and vice-versa) then add a "!" to the pin definition (for example, "endstop_pin: ^!ar3"), or remove the "!" if there is already one present.

If the endstop does not change at all then it generally indicates that the endstop is connected to a different pin. However, it may also require a change to the pullup setting of the pin (the '^' at the start of the endstop_pin name - most printers will use a pullup resistor and the '^' should be present).

Verify stepper motors

Use the STEPPER_BUZZ command to verify the connectivity of each stepper motor. Start by manually positioning the given axis to a midway point and then run `STEPPER_BUZZ STEPPER=stepper_x`. The STEPPER_BUZZ command will cause the given stepper to move one millimeter in a positive direction and then it will return to its starting position. (If the endstop is defined at position_endstop=0 then at the start of each movement the stepper will move away from the endstop.) It will perform this oscillation ten times.

If the stepper does not move at all, then verify the "enable_pin" and "step_pin" settings for the stepper. If the stepper motor moves but does not return to its original position then verify the "dir_pin" setting. If the stepper motor oscillates in an incorrect direction, then it generally indicates that the "dir_pin" for the axis needs to be inverted. This is done by adding a "!" to the "dir_pin" in the printer config file (or removing it if

one is already there). If the motor moves significantly more or significantly less than one millimeter then verify the "rotation_distance" setting.

Run the above test for each stepper motor defined in the config file. (Set the STEPPER parameter of the STEPPER_BUZZ command to the name of the config section that is to be tested.) If there is no filament in the extruder then one can use STEPPER_BUZZ to verify the extruder motor connectivity (use STEPPER=extruder). Otherwise, it's best to test the extruder motor separately (see the next section).

After verifying all endstops and verifying all stepper motors the homing mechanism should be tested. Issue a G28 command to home all axes. Remove power from the printer if it does not home properly. Rerun the endstop and stepper motor verification steps if necessary.

Verify extruder motor

To test the extruder motor it will be necessary to heat the extruder to a printing temperature. Navigate to the Octoprint temperature tab and select a target temperature from the temperature drop-down box (or manually enter an appropriate temperature). Wait for the printer to reach the desired temperature. Then navigate to the Octoprint control tab and click the "Extrude" button. Verify that the extruder motor turns in the correct direction. If it does not, see the troubleshooting tips in the previous section to confirm the "enable_pin", "step_pin", and "dir_pin" settings for the extruder.

Calibrate PID settings

Klipper supports [PID control](#) for the extruder and bed heaters. In order to use this control mechanism, it is necessary to calibrate the PID settings on each printer (PID settings found in other firmwares or in the example configuration files often work poorly).

To calibrate the extruder, navigate to the OctoPrint terminal tab and run the PID_CALIBRATE command. For example: `PID_CALIBRATE HEATER=extruder TARGET=170`

At the completion of the tuning test run `SAVE_CONFIG` to update the printer.cfg file the new PID settings.

If the printer has a heated bed and it supports being driven by PWM (Pulse Width Modulation) then it is recommended to use PID control for the bed. (When the bed heater is controlled using the PID algorithm it may turn on and off ten times a second, which may not be suitable for heaters using a mechanical switch.)

A typical bed PID calibration command is: `PID_CALIBRATE HEATER=heater_bed TARGET=60`

Next steps

This guide is intended to help with basic verification of pin settings in the Klipper configuration file. Be sure to read the [bed leveling](#) guide. Also see the [Slicers](#) document for information on configuring a slicer with Klipper.

After one has verified that basic printing works, it is a good idea to consider calibrating [pressure advance](#).

It may be necessary to perform other types of detailed printer calibration - a number of guides are available online to help with this (for example, do a web search for "3d printer calibration"). As an example, if you experience the effect called ringing, you may try following [resonance compensation](#) tuning guide.