# **Klipper Tricks**

## **Super Slicer controlled temperatures**

[gcode\_macro PRINT\_START]

# Use PRINT\_START for the slicer starting script - please customise for your slicer of choice

default\_parameter\_BED\_TEMP: 101

default\_parameter\_EXTRUDER\_TEMP: 235

gcode:

M117 Heating...

M140 S{BED\_TEMP} ; set bed final temp

M104 S{EXTRUDER\_TEMP} ; set extruder final temp

M190 S{BED\_TEMP} ; wait for bed final temp

M109 S{EXTRUDER\_TEMP} ; wait for extruder final temp

M900

G21 ; set units to millimeters

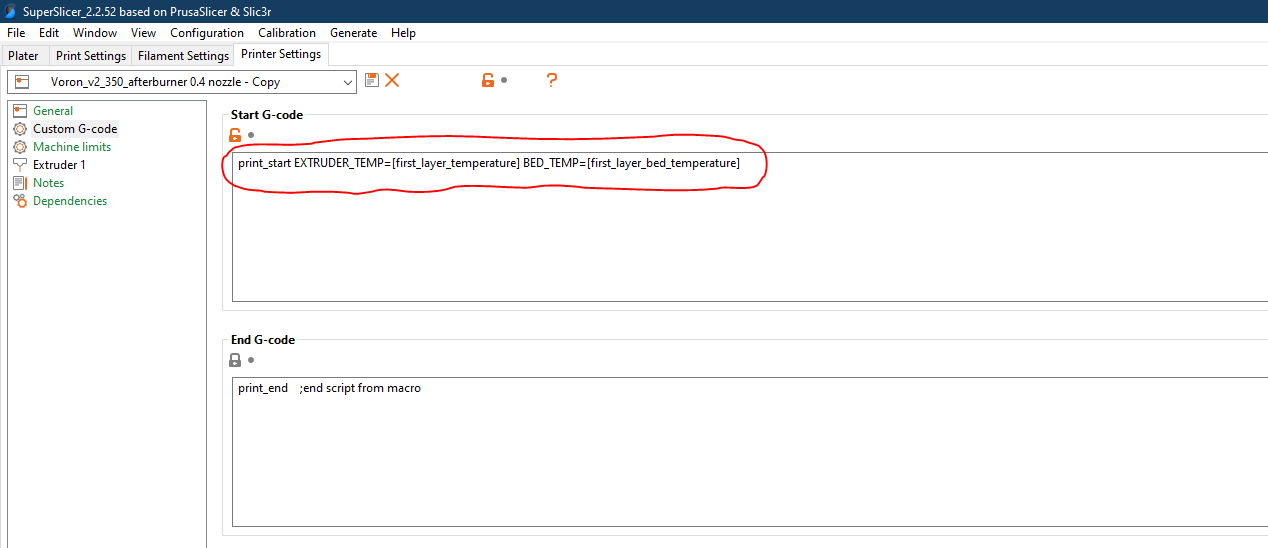
G90 ; use absolute coordinates

M83 ; use relative distances for extrusion

G32 ; home all axes

G1 Z20 F3000 ; move nozzle away from bed

This is the screenshot from within SS that needs to be done and the printer.cfg PRINT\_START macro(edited)



## **Heat Soak Trick**

#Pause/Resume Functionality

[pause\_resume]

#Wait for chamber temp, kick off wait loop if not already at temp

[gcode\_macro M191]

default\_parameter\_S: 0

variable\_chambertargettemp: 0

gcode:

SET\_GCODE\_VARIABLE MACRO=M191 VARIABLE=chambertargettemp VALUE={S} ; set target temp for reference outside of the macro (for the loop)

{% if not printer["temperature\_sensor chamber\_temp"].temperature >= S|int %} ; ##IF CHAMBER TEMP IS NOT ALREADY REACHED##

{ action\_respond\_info("Chamber not at temp yet, pausing...") }

{% if not printer.pause\_resume.is\_paused %}

PAUSE ; pause if not already paused

{% endif %}

UPDATE\_DELAYED\_GCODE ID=M191-Wait DURATION=5 ; start wait loop

{% else %}

{ action\_respond\_info("Chamber at or above temp, continuing...") } ; ##IF CHAMBER TEMP IS ALREADY REACHED##

UPDATE\_DELAYED\_GCODE ID=M191-Wait DURATION=0 ; break wait loop if it happens to be running already (shouldn't be)

{% if printer.pause\_resume.is\_paused %}

RESUME ; resume if paused (shouldn't be)

{% endif %}

{% endif %}

#This part will loop until the desired chamber temp is reached, then resume the print

[delayed\_gcode M191-Wait]

gcode:

{% if printer["temperature\_sensor chamber\_temp"].temperature >= printer["gcode\_macro M191"].chambertargettemp|int %} ; ##IF CHAMBER TEMP IS REACHED##

{ action\_respond\_info("Chamber at or above temp, continuing...") }

{% if printer.pause\_resume.is\_paused %}

RESUME ; break loop, resume print

{% endif %}

{% else %} ; ##IF CHAMBER TEMP IS NOT YET REACHED##

{ action\_respond\_info("Chamber not at temp yet, waiting...") }

UPDATE\_DELAYED\_GCODE ID=M191-Wait DURATION=5 ; continue waiting loop

{% endif %}

So M191 S45 will wait for chamber temp to reach 45 degrees then resume print, for example. M191 S0 will break out of the wait loop and continue.

I am using a chamber thermistor but you should be able to sub in the hotend... just replace

printer["temperature\_sensor chamber\_temp"].temperature

with

printer.extruder.temperature

You can comment out the action\_respond\_info lines if you don't want it being so verbose in the terminal. **IMPORTANT CAVEAT: This does not like being run inside of other macros.** Klipper doesn't like pausing mid-macro. Put it as a separate line in your slicer start gcode, or put it as the very last line of a macro. I had to split my PRINT\_START into 2 parts (see below).(edited)

## **Bed Mesh Setup**

You'll have a bad time with bed mesh if you're missing this setting:

##FOLLOW THE FORUMULA FOR relative\_reference\_index, for example:

##[(7x7)-1] / 2 = 24

probe\_count: 7,7

relative\_reference\_index: 24

Here just use my config

[bed\_mesh]

mesh\_min: 30,50

mesh\_max: 320,300

speed: 300

horizontal\_move\_z: 6

probe\_count: 7,7

algorithm: bicubic

relative\_reference\_index: 24

#fade\_start: 1

# Default is 1.0.

#fade\_end: 10

# Default is 0.0, which disables fade.

#fade\_target: 0

# The z position in which fade should converge.

move\_check\_distance: 3

# The distance (in mm) along a move to check for split\_delta\_z.

# This is also the minimum length that a move can be split. Default

# is 5.0.

split\_delta\_z: 0.0125

# The amount of Z difference (in mm) along a move that will

# trigger a split. Default is .025.

mesh\_pps: 4,4

# A comma separated pair of integers (X,Y) defining the number of

# points per segment to interpolate in the mesh along each axis. A

# "segment" can be defined as the space between each probed

# point. The user may enter a single value which will be applied

# to both axes. Default is 2,2.

#bicubic\_tension: 0.2

# When using the bicubic algorithm the tension parameter above

# may be applied to change the amount of slope interpolated.

# Larger numbers will increase the amount of slope, which

# results in more curvature in the mesh. Default is .2.

## **Yet Another Heat Sock Script**

If anyone is interested, this is my heat\_soak script for frurunning before prints:

[gcode\_macro heat\_soak]

gcode:

{% if printer.toolhead.status == "Ready" %}

G28

G1 Z10

G1 X150 Y150 F6000

M140 S100

UPDATE\_DELAYED\_GCODE ID=heatsoakDelay DURATION=600

{% else %}

M117 "Load disabled while printing!"

{% endif %}

[delayed\_gcode heatsoakDelay]

gcode:

M117

SET\_PIN PIN=caselight VALUE=1

G4 P250

SET\_PIN PIN=caselight VALUE=0

G4 P250

SET\_PIN PIN=caselight VALUE=1

G4 P250

SET\_PIN PIN=caselight VALUE=0

It centers the nozzle turns on the bed and then after 10 minutes flashed the lights

hey @Saiyanslayer you define defaults for parameters before the gcode section, with the prefix default\_parameter\_ such as:

[gcode\_macro PARK\_MACRO]

default\_parameter\_X: 20

default\_parameter\_Y: 330

default\_parameter\_Z: 100

gcode:

SAVE\_GCODE\_STATE NAME=PARK\_MACRO\_state

G91 ; relative positioning

G1 E-2 F1000 ; retract filament

G1 Z10 ; lift z slightly

G90 ; absolute positioning

G1 X{X} Y{Y} Z{Z} F3000 ; park the head

RESTORE\_GCODE\_STATE name=PARK\_MACRO\_state

[gcode\_macro START\_PRINT]

default\_parameter\_BED\_TEMP: 100

default\_parameter\_EXTRUDER\_TEMP: 245

gcode:

{% if printer.heater\_bed.temperature <= (BED\_TEMP|int - 20) %}

M190 S{BED\_TEMP|int-20}

{%endif %}

M104 S{EXTRUDER\_TEMP|int-40}

M190 S{BED\_TEMP}

M109 S{EXTRUDER\_TEMP|int-40}

G32

G1 Z5 F5000

M109 S{EXTRUDER\_TEMP}

[12:03 PM]

and the start gcode in prusaslicer now looks like:

[12:04 PM]

START\_PRINT BED\_TEMP=[first\_layer\_bed\_temperature] EXTRUDER\_TEMP={first\_layer\_temperature[0]}

M109 S{first\_layer\_temperature[0]};

M190 S[first\_layer\_bed\_temperature];

[12:04 PM]

the m109 & m190 shouldn't be needed, except if the slicer doesn't see them, it assumes it should add its own at the beginning. I haven't figured out a way around that yet