

Install DGUS Reloaded on Ender 5+

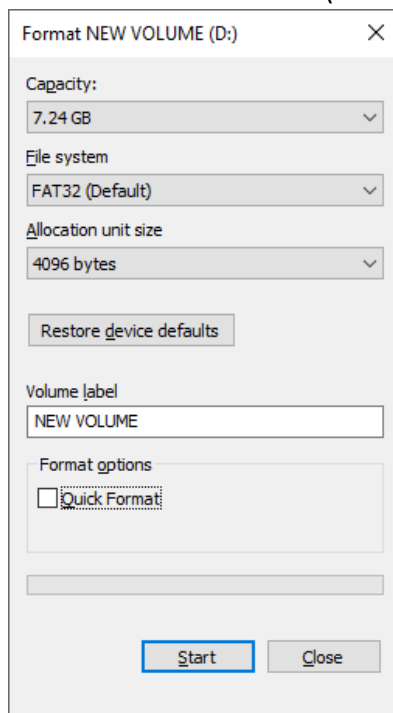
Version 1.0 20230916 Jacques De Jonghe

1. Assumptions

- 1.1 An Ender 5+ with a V2.x board and the original screen.
- 1.2 A Raspberry PI (RPI) connected to the Ender 5+ via USB.
- 1.3 The RPI can be a 3, 4 or zero 2.
- 1.4 The RPI has a stable power supply. Better, use the power supply of the Ender 5+ via a 5V buck converter.

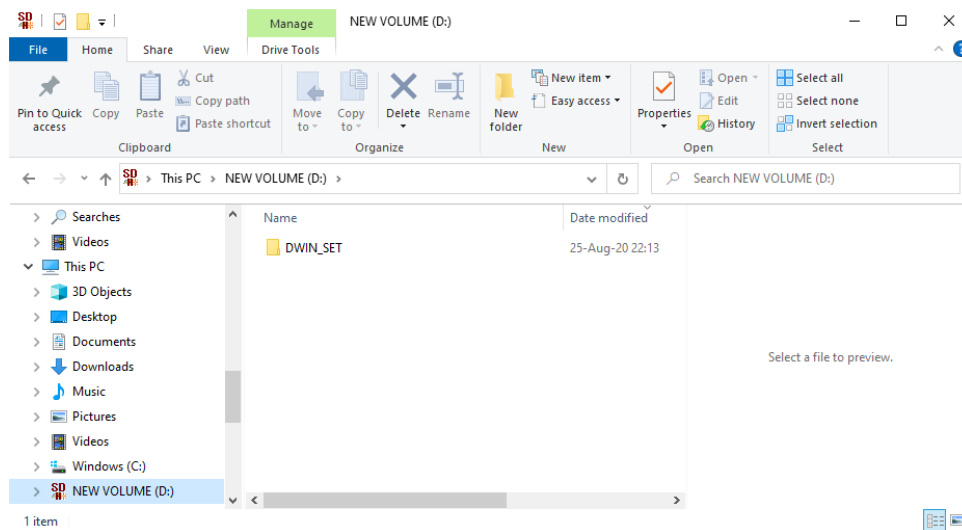
2. Install DGUS_Reloaded Screen Firmware

- 2.1 Format a micro-SD card (max 8GB) as FAT32 and 4096 Bytes as Allocation Unit size



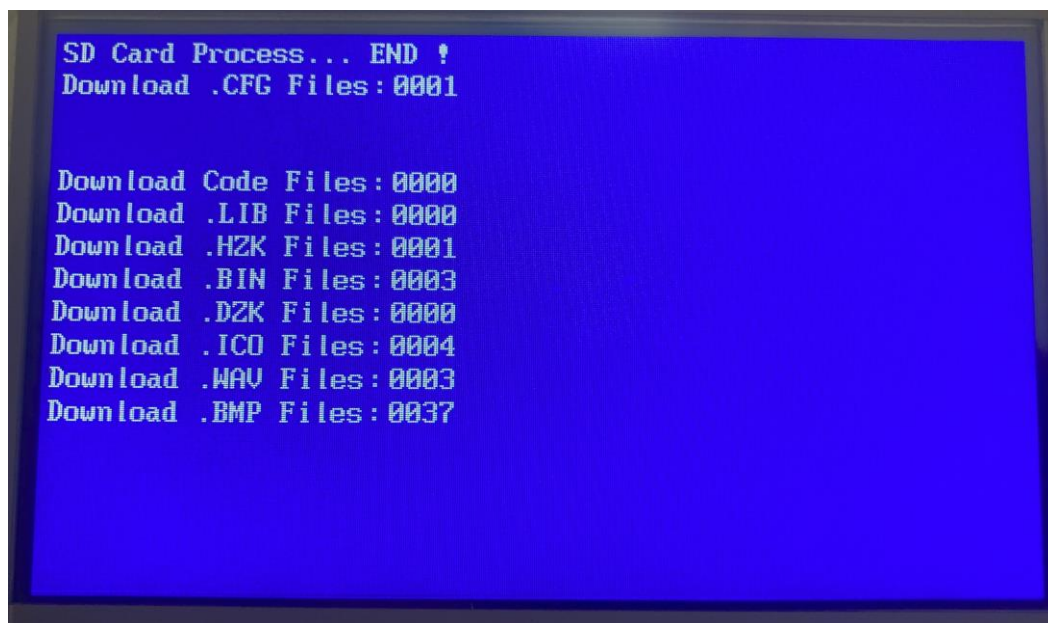
- 2.2 Download the screen Firmware from:
https://github.com/Desuuuu/DGUS-reloaded-Klipper/releases/download/1.0.2/DWIN_SET.tar.gz

2.3 Unzip the directory **DWIN_SET** of the screen Firmware on the Micro-SD card . You can use [7zip](#) for that.



2.4 Remove the Micro-SD card from the computer and install it in the screen slot **with the printer powered-off**.

2.5 Power-on the printer. The Screen Firmware is being programmed and you will see the different screens being loaded. After a few seconds, you will get:



2.6 Power-off the printer and remove the micro-SD card from the screen.

3. Install Klipper-Moonraker-Mainsail RPI Software

3.1 Insert in the computer the micro-SD card that you will use on the RPI

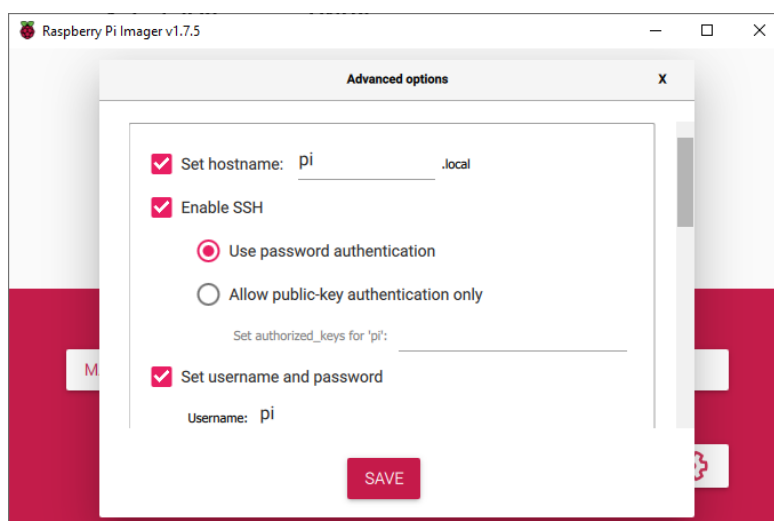
3.2 Download the Raspberry Pi Imager from <https://www.raspberrypi.com/software/> and execute it.

3.3 Choose **Mainsail OS 32-Bit** as Operating System in the 3D Printing Operating Systems:

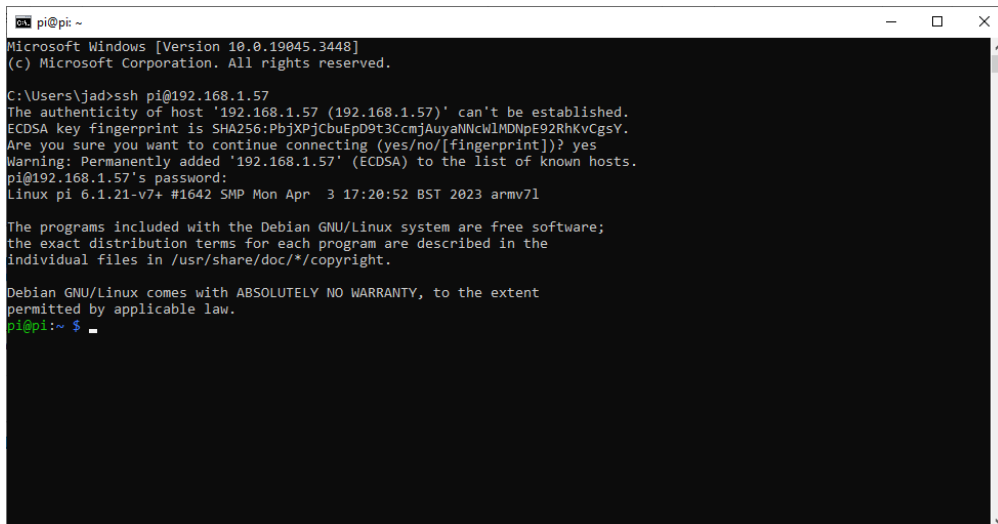


3.4 Choose your micro-SD card location in the MASS STORAGE option

3.5 In the settings, set the HOST name, enable SSH, the username and password, configure the wireless lan with your local WIFI network and SAVE these settings:



- 3.6 Click on the WRITE button in order to install the RPI Software in the micro-SD card.
That will take something like 20 minutes.
- 3.7 When finished, remove the Micro-SD card from the computer and install it in the RPI slot **with the RPI powered-off**.
- 3.8 Power on the RPI and the printer. Wait 5 minutes in order to have the RPI configured initially.
- 3.9 Find out the IP address of your RPI on your router (192.168.1.57 in my case).
- 3.10 Start a console on your computer (cmd on Windows).
- 3.11 Access your RPI via **ssh pi@192.168.1.57**
pi being the Username. You can also use pi@pi.local with pi.local being the HOST name.



```
pi@pi: ~
Microsoft Windows [Version 10.0.19045.3448]
(c) Microsoft Corporation. All rights reserved.

C:\Users\jad>ssh pi@192.168.1.57
The authenticity of host '192.168.1.57 (192.168.1.57)' can't be established.
ECDSA key fingerprint is SHA256:PbjXPjCbuEpD9t3CcmjAuyaNNcWlMDNpE92RhKvCgsY.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.57' (ECDSA) to the list of known hosts.
pi@192.168.1.57's password:
Linux pi 6.1.21-v7+ #1642 SMP Mon Apr  3 17:20:52 BST 2023 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
pi@pi:~$
```

- 3.12 Enter the following commands in order to upgrade the operating system to the latest status: **sudo apt update && sudo apt full-upgrade && sudo apt clean**
That will take some time depending on the latest upgrade status of the RPI image and the speed of your internet connection
- 3.13 Reboot the RPI by entering: **sudo reboot**
- 3.14 Reconnect via ssh to the RPI: **ssh pi@192.168.1.57**

4. Install Kiauh on the RPI

4.1 Install KIAUH: `git clone https://github.com/dw-0/kiauh.git`

4.2 Enter the Klipper repository containing the DGUS_Reloaded version to be used:

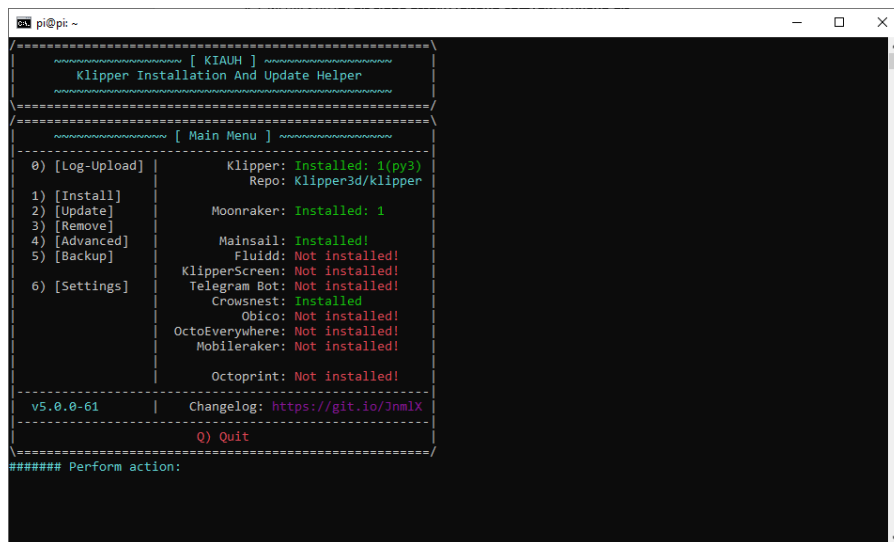
`cd kiauh`

`nano klipper_repos.txt`

Enter <https://github.com/ploucandco/klipper> and further Exit (^) and save the file.

`cd ..`

4.3 So you should be in the home directory. Enter the following command in order to start Kiauh: `kiauh/kiauh.sh`



```
pi@pi: ~
===== [ KIAUH ] =====
Klipper Installation And Update Helper
=====
===== [ Main Menu ] =====

0) [Log-Upload]      Klipper: Installed: 1(py3)
                    Repo: Klipper3d/klipper

1) [Install]         Moonraker: Installed: 1

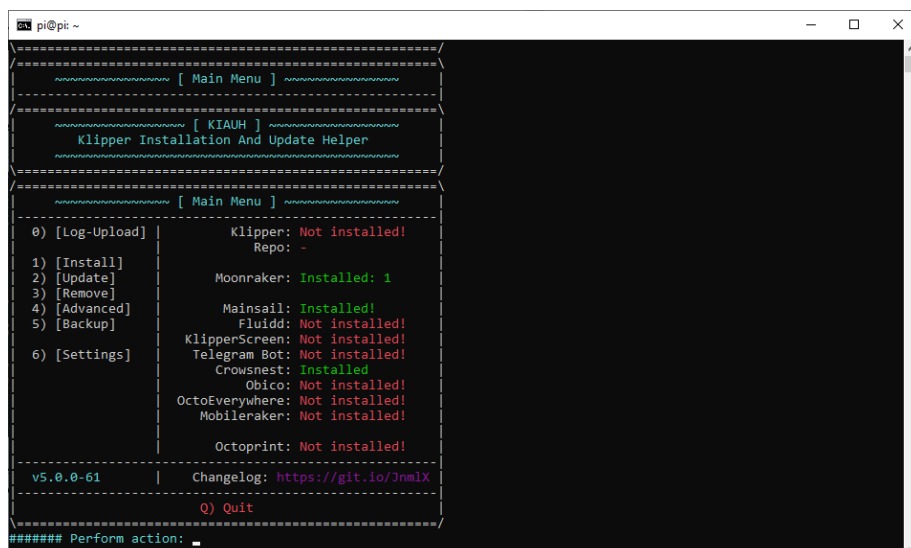
2) [Update]         Mainsail: Installed!
3) [Remove]         Fluid: Not installed!
4) [Advanced]       KlipperScreen: Not installed!
5) [Backup]         Telegram Bot: Not installed!
                    Crowsnest: Installed
                    Obico: Not installed!
                    OctoEverywhere: Not installed!
                    Mobileraker: Not installed!

6) [Settings]       Octoprint: Not installed!

v5.0.0-61 | Changelog: https://git.io/JnmlX
Q) Quit

##### Perform action:
```

4.4 Enter **3)** for Remove and **1)** for Klipper. That will remove the current instance of Klipper. Enter **B)** to come back on the main menu.



```
pi@pi: ~
===== [ KIAUH ] =====
Klipper Installation And Update Helper
=====
===== [ Main Menu ] =====

0) [Log-Upload]      Klipper: Not installed!
                    Repo: -

1) [Install]         Moonraker: Installed: 1

2) [Update]         Mainsail: Installed!
3) [Remove]         Fluid: Not installed!
4) [Advanced]       KlipperScreen: Not installed!
5) [Backup]         Telegram Bot: Not installed!
                    Crowsnest: Installed
                    Obico: Not installed!
                    OctoEverywhere: Not installed!
                    Mobileraker: Not installed!

6) [Settings]       Octoprint: Not installed!

v5.0.0-61 | Changelog: https://git.io/JnmlX
Q) Quit

##### Perform action: _
```

4.5 Now Select the right Klipper depository: **6)** for Settings, **1)** Set custom Klipper repository, **0)** for ploucandco/klipper, and 2 times **B)** for Back to main menu

```
pi@pi: ~
===== [ KIAUH ] =====
Klipper Installation And Update Helper
=====
===== [ Set custom Klipper repo ] =====

0) ploucandco/klipper -> master

0) <- Back      |      H) Help [?]
=====
##### Perform action: 0
[0] Repo: ploucandco/klipper Branch: master

##### Set custom Klipper repository to:
  * Repository: ploucandco/klipper
  * Branch: master
[B OK] This repo will now be used for new Klipper installations!

##### Perform action:
```

4.6 Now install the new instance of Klipper: **1)** for Install, **1)** for Klipper, **1)** for Python 3.x and **1)** for 1 instance of Klipper. That will take some minutes. Then come back to the main menu **B)**.

```
pi@pi: ~
===== [ KIAUH ] =====
Klipper Installation And Update Helper
=====
===== [ Main Menu ] =====

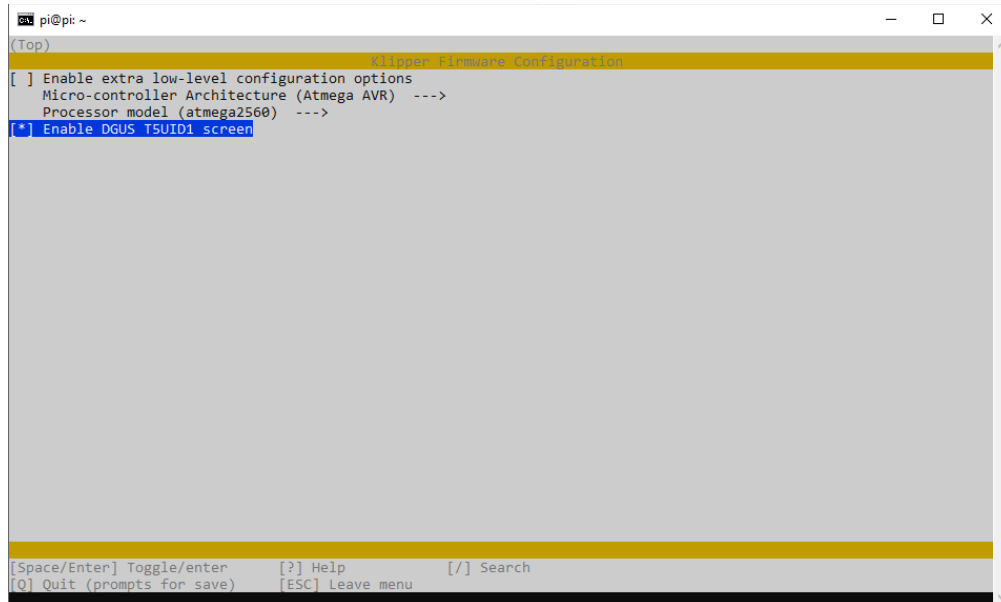
0) [Log-Upload]      Klipper: Installed: 1(py3)
                    Repo: custom
1) [Install]         Moonraker: Installed: 1
2) [Update]
3) [Remove]          Mainsail: Installed!
4) [Advanced]        Fluid: Not installed!
5) [Backup]          KlipperScreen: Not installed!
                    Telegram Bot: Not installed!
6) [Settings]        Crowsnest: Installed
                    Obico: Not installed!
                    OctoEverywhere: Not installed!
                    Mobileraker: Not installed!
                    Octoprint: Not installed!

v5.0.0-61          | Changelog: https://git.io/3nmlX
                    Q) Quit

##### Perform action: _
```

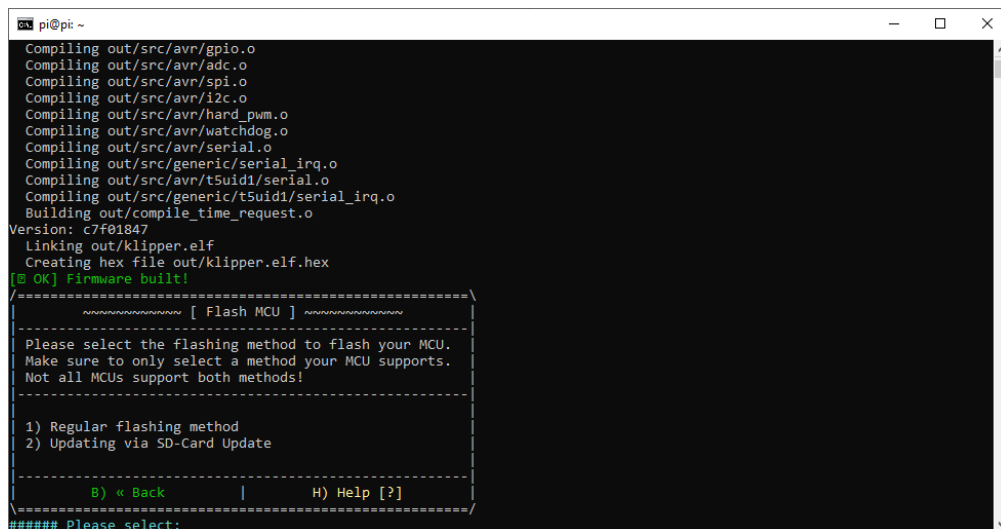
5. Install Klipper on Creality v2.2 board

5.1 In Kiuh, enter **4**) Advanced and **4**) Build+Flash and Select: **[*]** Enable DGUS T5UID1 Screen. The other options should be already correct:



```
pi@pi: ~  
(Top) Klipper Firmware Configuration  
[ ] Enable extra low-level configuration options  
    Micro-controller Architecture (Atmega AVR) --->  
    Processor model (atmega2560) --->  
[*] Enable DGUS T5UID1 screen  
  
[Space/Enter] Toggle/enter    [?] Help    [/] Search  
[Q] Quit (prompts for save)    [ESC] Leave menu
```

5.2 Enter **Q** and **Y** in order to save the configuration and Quit.



```
pi@pi: ~  
Compiling out/src/avr/gpio.o  
Compiling out/src/avr/adc.o  
Compiling out/src/avr/spi.o  
Compiling out/src/avr/i2c.o  
Compiling out/src/avr/hard_pwm.o  
Compiling out/src/avr/watchdog.o  
Compiling out/src/avr/serial.o  
Compiling out/src/generic/serial_irq.o  
Compiling out/src/avr/t5uid1/serial.o  
Compiling out/src/generic/t5uid1/serial_irq.o  
Building out/compile_time_request.o  
Version: c7f01847  
Linking out/klipper.elf  
Creating hex file out/klipper.elf.hex  
[E OK] Firmware built!  
##### [ Flash MCU ] #####  
Please select the flashing method to flash your MCU.  
Make sure to only select a method your MCU supports.  
Not all MCUs support both methods!  
  
1) Regular flashing method  
2) Updating via SD-Card Update  
  
B) « Back    |    H) Help [?]  
##### Please select:
```

5.3 Enter **1)** for Regular flashing method, **1)** for USB, **1)** for the MCU 1 and **Y** in order to continue.

```
pi@pi: ~
##### Select MCU to flash: 1
##### You selected:
  • MCU #1: /dev/serial/by-id/usb-FTDI_FT232R_USB_UART_AB0K4TOX-if00-port0
##### Continue? (Y/n):
  [Y] Yes
##### Flashing /dev/serial/by-id/usb-FTDI_FT232R_USB_UART_AB0K4TOX-if00-port0 ...
##### Stop klipper.service ...
[OK] Stop klipper.service successfull!
Flashing out/klipper.elf.hex to /dev/serial/by-id/usb-FTDI_FT232R_USB_UART_AB0K4TOX-if00-port0 via avrdude
avrdude: AVR device initialized and ready to accept instructions

Reading | ##### | 100% 0.04s
avrdude: Device signature = 0x1e9801 (probably m2560)
avrdude: reading input file "out/klipper.elf.hex"
avrdude: writing flash (34142 bytes):

Writing | ##### | 100% 8.57s

avrdude: 34142 bytes of flash written
avrdude: verifying flash memory against out/klipper.elf.hex:
avrdude: load data flash data from input file out/klipper.elf.hex:
avrdude: input file out/klipper.elf.hex contains 34142 bytes
avrdude: reading on-chip flash data:

Reading | ##### | 100% 7.37s

avrdude: verifying ...
avrdude: 34142 bytes of flash verified

avrdude: safemode: Fuses OK (E:FD, H:DB, L:FF)

avrdude done. Thank you.

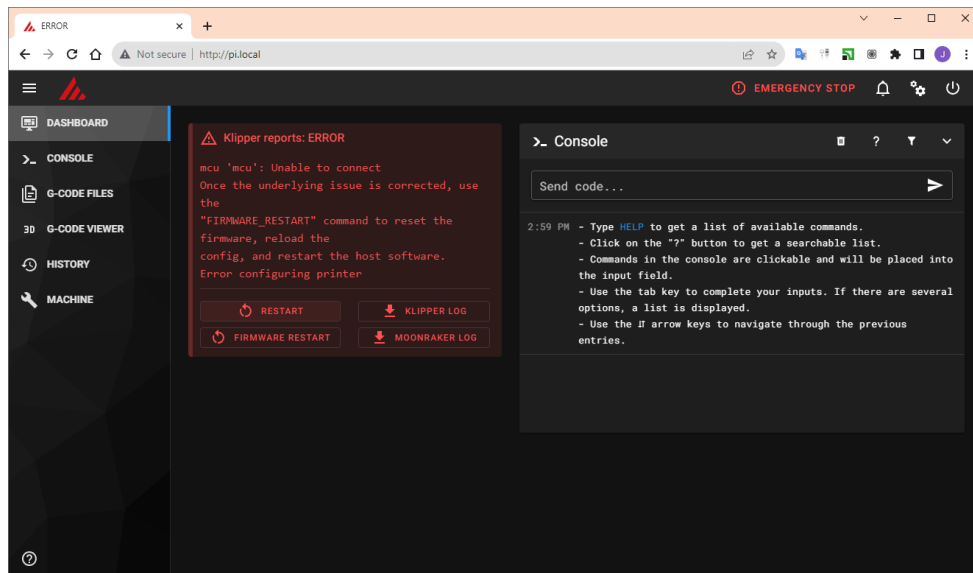
[OK] Flashing successfull!
##### Start klipper.service ...
[OK] Start klipper.service successfull!

=====
          [ Advanced Menu ]
=====
Klipper & API:      | Mainsail:
  1) [Rollback]      | 6) [Theme installer]
Firmware:          | System:
  2) [Build only]    | 7) [Change hostname]
  3) [Flash only]    | Extras:
  4) [Build + Flash] | 8) [G-Code Shell Command]
  5) [Get MCU ID]    |
                    |
                    | B) « Back
=====
##### Perform action: 
```


6. Modify printer.cfg

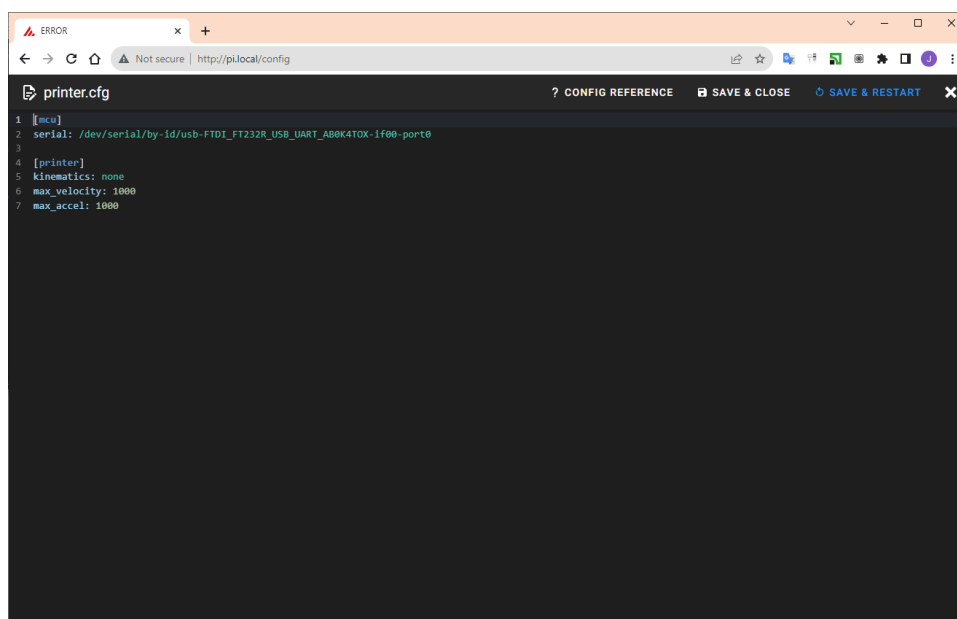
6.1 In the KIAUH advanced screen, enter **5) Get MCU ID** and **1) USB**. Copy MCU 1#: **usb-FTDI_FT232R_USB_UART_AB0K4TOX-if00-port0**

6.2 Connect to Mainsail on a browser with the IP address of the RPI or the HOST name.

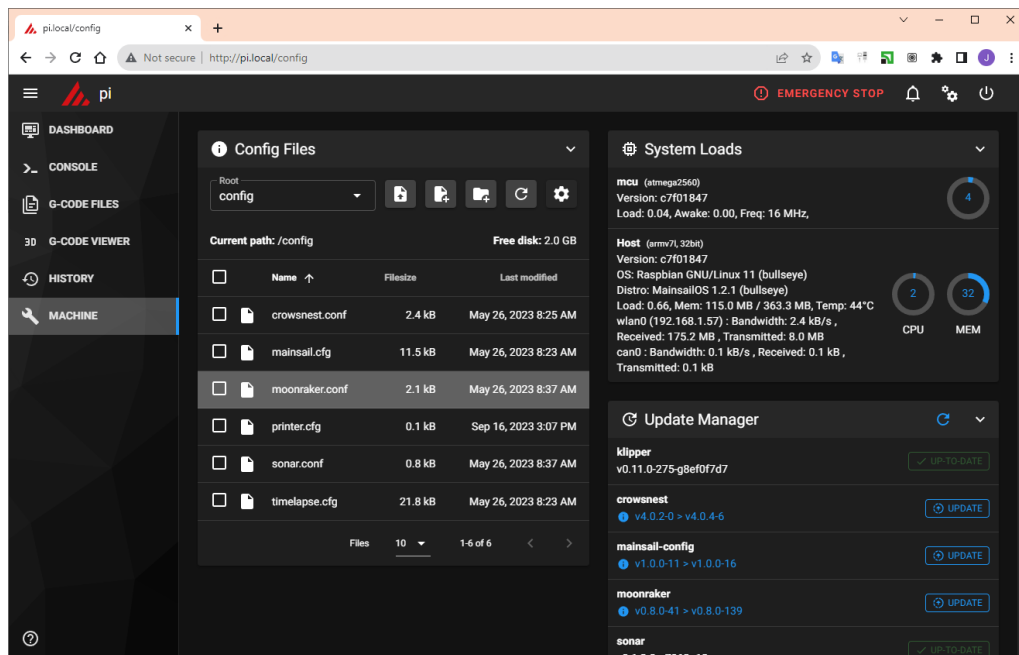


6.3 Select **Machine**. If the Left screen menu is not visible, select the 3 horizontal lines.

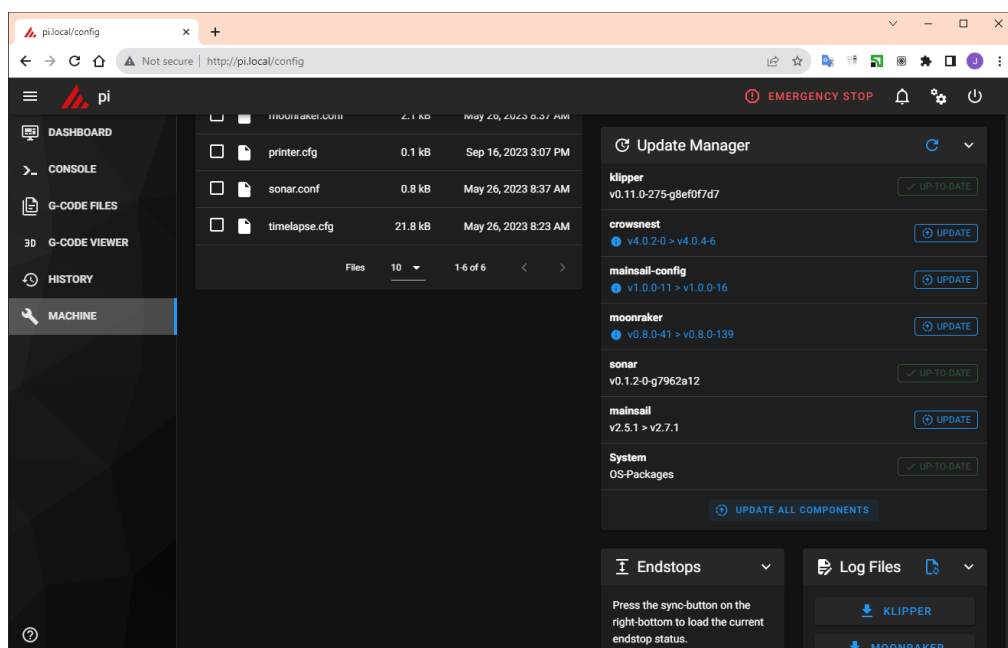
6.4 In Select **printer.cfg**, paste the right **MCU name** and select **SAVE & RESTART**.



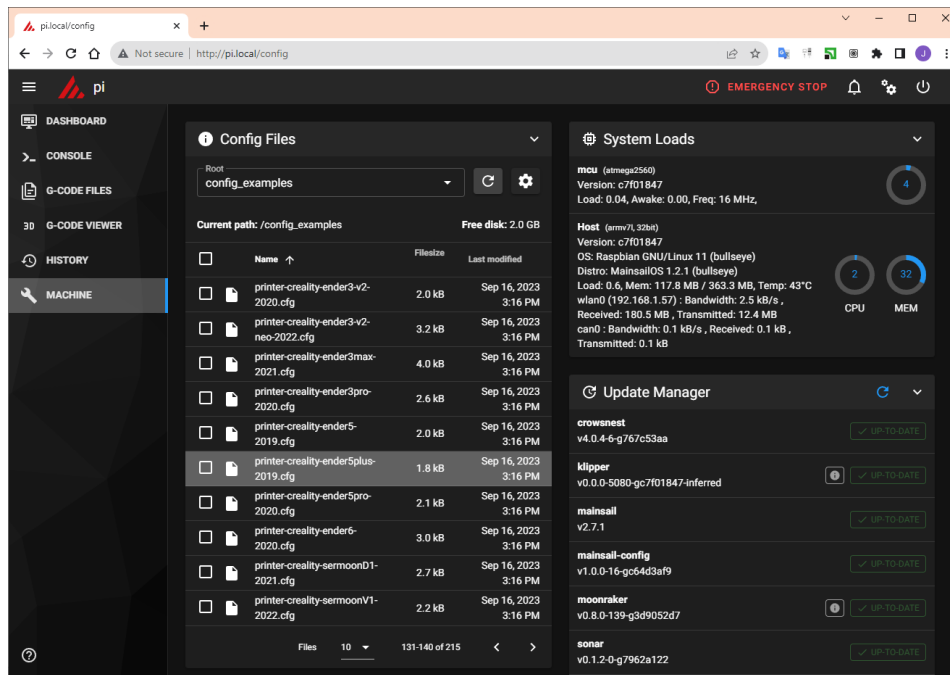
6.5 Klipper should restart and you should get connection to the mcu (atmega2560)



6.6 Select **UPDATE ALL COMPONENTS** in order to have a fully updated system. That will take some time.

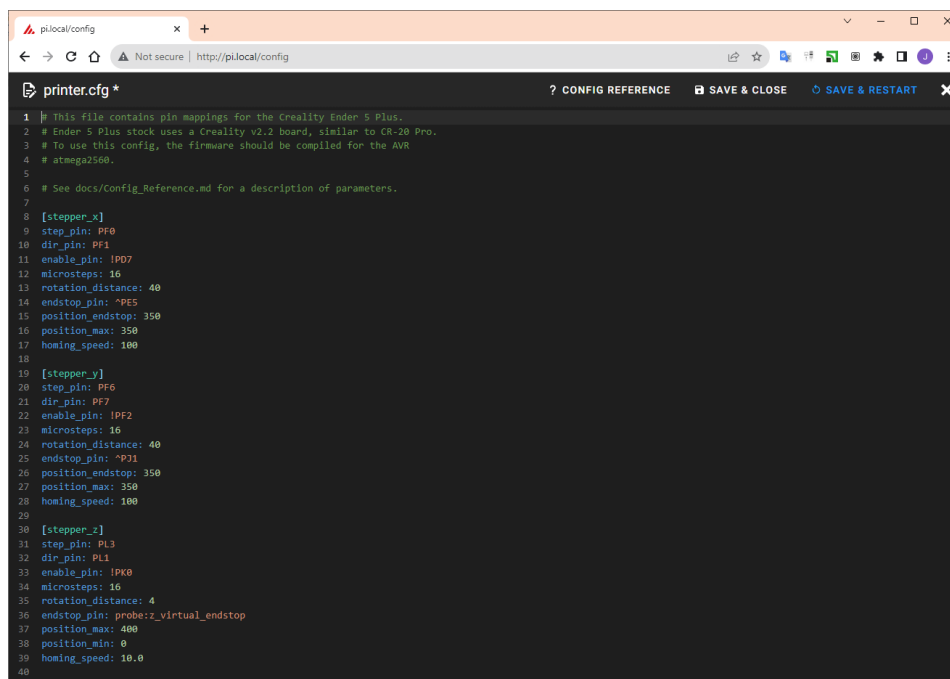


6.7 Select the file **printer-creality-ender5plus-2019.cfg** in the directory **config_examples**:



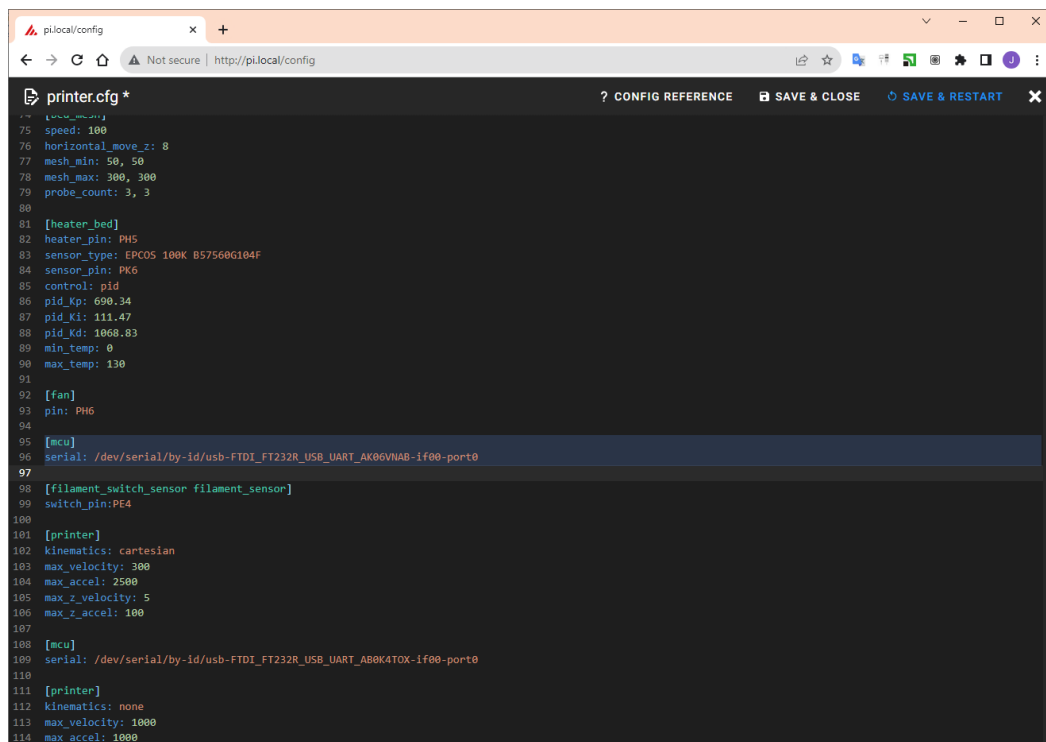
6.8 Copy the whole content of the file and close the editor by selecting **X**.

6.9 Paste in the beginning of **printer.cfg** in the directory **config**:



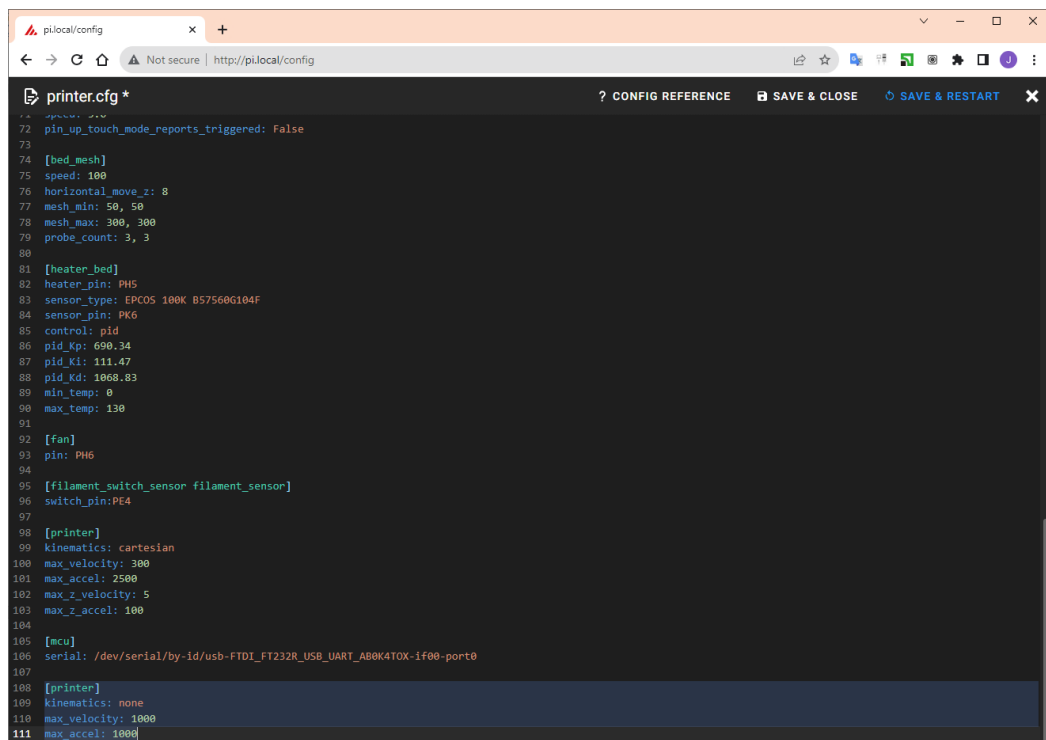
6.10 In the beginning of the file, add **[include mainsail.cfg]**

6.11 At the end of the file, remove the first **[mcu]** entry:



```
1 4 speed: 100
75 speed: 100
76 horizontal_move_z: 8
77 mesh_min: 50, 50
78 mesh_max: 300, 300
79 probe_count: 3, 3
80
81 [heater_bed]
82 heater_pin: PH5
83 sensor_type: EPCOS 100K B57560G104F
84 sensor_pin: PK6
85 control: pid
86 pid_Kp: 690.34
87 pid_Ki: 111.47
88 pid_Kd: 1068.83
89 min_temp: 0
90 max_temp: 130
91
92 [fan]
93 pin: PH6
94
95 [mcu]
96 serial: /dev/serial/by-id/usb-FTDI_FT232R_USB_UART_AK06VNAB-if00-port0
97
98 [filament_switch_sensor filament_sensor]
99 switch_pin: PE4
100
101 [printer]
102 kinematics: cartesian
103 max_velocity: 300
104 max_accel: 2500
105 max_z_velocity: 5
106 max_z_accel: 100
107
108 [mcu]
109 serial: /dev/serial/by-id/usb-FTDI_FT232R_USB_UART_AB0K4TOX-if00-port0
110
111 [printer]
112 kinematics: none
113 max_velocity: 1000
114 max_accel: 1000
```

6.12 And remove the second **[printer]** entry:



```
1 4 speed: 100
72 pin_up_touch_mode_reports_triggered: False
73
74 [bed_mesh]
75 speed: 100
76 horizontal_move_z: 8
77 mesh_min: 50, 50
78 mesh_max: 300, 300
79 probe_count: 3, 3
80
81 [heater_bed]
82 heater_pin: PH5
83 sensor_type: EPCOS 100K B57560G104F
84 sensor_pin: PK6
85 control: pid
86 pid_Kp: 690.34
87 pid_Ki: 111.47
88 pid_Kd: 1068.83
89 min_temp: 0
90 max_temp: 130
91
92 [fan]
93 pin: PH6
94
95 [filament_switch_sensor filament_sensor]
96 switch_pin: PE4
97
98 [printer]
99 kinematics: cartesian
100 max_velocity: 300
101 max_accel: 2500
102 max_z_velocity: 5
103 max_z_accel: 100
104
105 [mcu]
106 serial: /dev/serial/by-id/usb-FTDI_FT232R_USB_UART_AB0K4TOX-if00-port0
107
108 [printer]
109 kinematics: none
110 max_velocity: 1000
111 max_accel: 1000
```

6.13 And add the following entries:

[t5uid1]

firmware: dgus_reloaded

machine_name: Ender 5 Plus

[bed_screws]

screw1:30,40

screw1_name:FrontLeft

screw2:325,40

screw2_name:FrontRight

screw3:325,295

screw3_name:BackRight

screw4:30,295

screw4_name:BackLeft

[screws_tilt_adjust]

screw1: 61,80

screw1_name: FrontLeftScrew

screw2: 350,80

screw2_name: FrontRightScrew

screw3: 350,345

screw3_name: BackRightScrew

screw4: 61,345

screw4_name: BackLeftScrew

horizontal_move_z: 10

speed: 200

screw_thread: CW-M4 # Use CW for Clowise and CCW for Counter

[temperature_sensor raspberry_pi]

sensor_type: temperature_host

min_temp: 10

max_temp: 100

[gcode_macro G29]

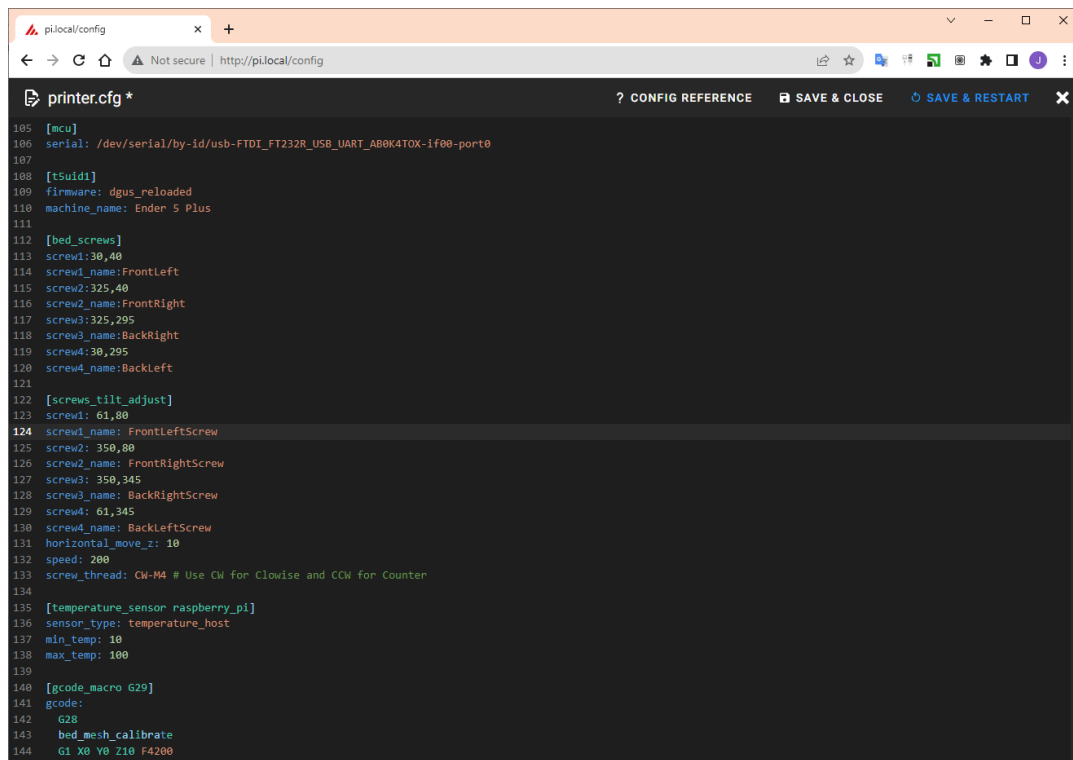
gcode:

G28

bed_mesh_calibrate

G1 X0 Y0 Z10 F4200

6.14 So the end of the printer.cfg file should look like:



The screenshot shows a web browser window with the address bar displaying 'http://pi:local/config'. The page title is 'printer.cfg *'. The main content area shows the configuration file with the following settings:

```
105 [mcu]
106 serial: /dev/serial/by-id/usb-FTDI_FT232R_USB_UART_A80K4TOX-if00-port0
107
108 [tsuid1]
109 firmware: dgus_reloaded
110 machine_name: Ender 5 Plus
111
112 [bed_screws]
113 screw1:30,40
114 screw1_name:FrontLeft
115 screw2:325,40
116 screw2_name:FrontRight
117 screw3:325,295
118 screw3_name:BackRight
119 screw4:30,295
120 screw4_name:BackLeft
121
122 [screws_tilt_adjust]
123 screw1: 61,80
124 screw1_name: FrontLeftScrew
125 screw2: 350,80
126 screw2_name: FrontRightScrew
127 screw3: 350,345
128 screw3_name: BackRightScrew
129 screw4: 61,345
130 screw4_name: BackLeftScrew
131 horizontal_move_z: 10
132 speed: 200
133 screw_thread: CW-M4 # Use CW for Clowise and CCW for Counter
134
135 [temperature_sensor raspberry_pi]
136 sensor_type: temperature_host
137 min_temp: 10
138 max_temp: 100
139
140 [gcode_macro G29]
141 gcode:
142     G28
143     bed_mesh_calibrate
144     G1 X0 Y0 Z10 F4200
```

6.15 Select **SAVE & RESTART**. Wait 1 minute in order for Klipper to restart and you will see the following on the printer screen:

