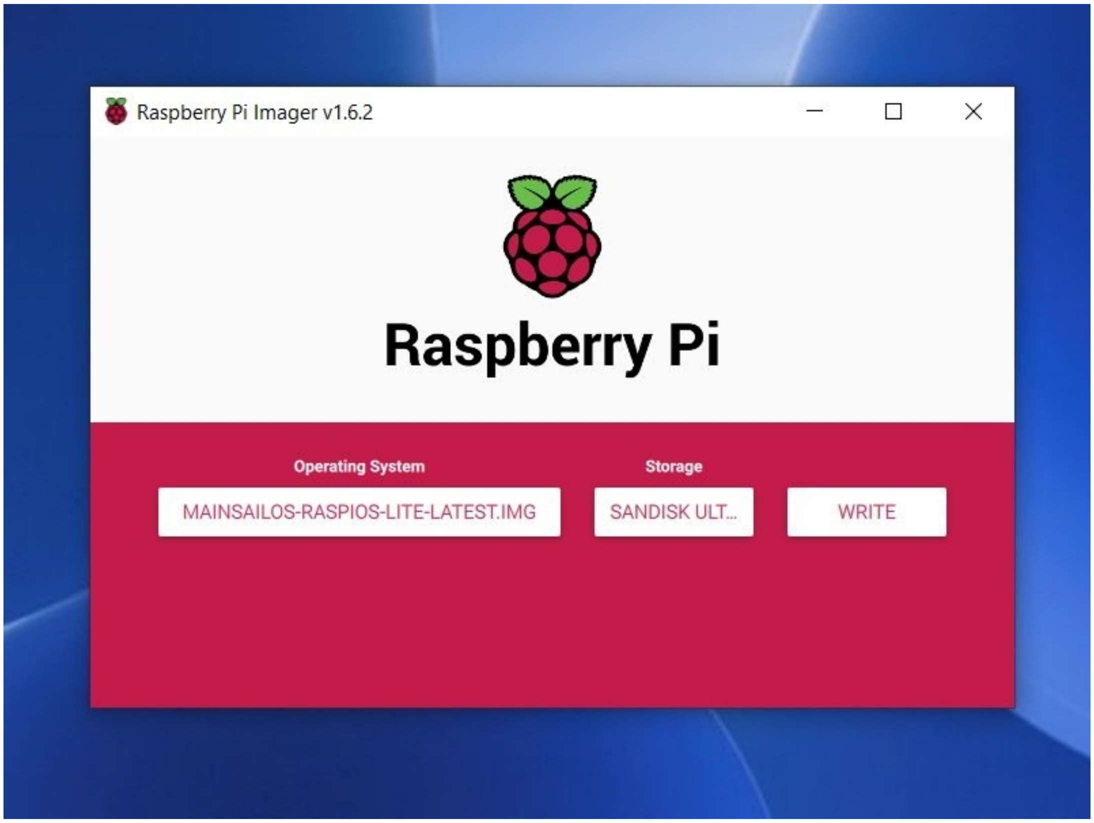
Klipper+Mainsail Install

This guide is still a WIP. The Arduino +...

Written By: David Husolo



Raspberry Pi Imager v1.6.2
Raspberry Pi
Operating System
MAINSAILOS-RASPIOS-LITE-LATEST.IMG
Storage
SANDISK ULT...
WRITE

INTRODUCTION

This guide is still a WIP. The Arduino + NeoPixels are not compatible with Klipper as is. Instead Pawel Drylenko has developed a script that runs on the Pi. This guide is missing the wiring diagrams and setup files for the NeoPixels as well as the custom Mainsail theme from Ben Levi. This guide will be fully completed within the next few days.

There are a number of methods to install Klipper. After a number of botched installs I found this method to be the most consistent and easiest.

You can use virtually any RPI model but it’s recommended to use Raspberry Pi 4. You’ll need at least a Class-10 8gb SD card. If a UHS(Ultra High Speed) is available, you’ll want at least a UHS-1.

For my setup I’m using a Raspberry Pi 4 8GB and a SanDisk 64GB USB3.1 Flash Drive.

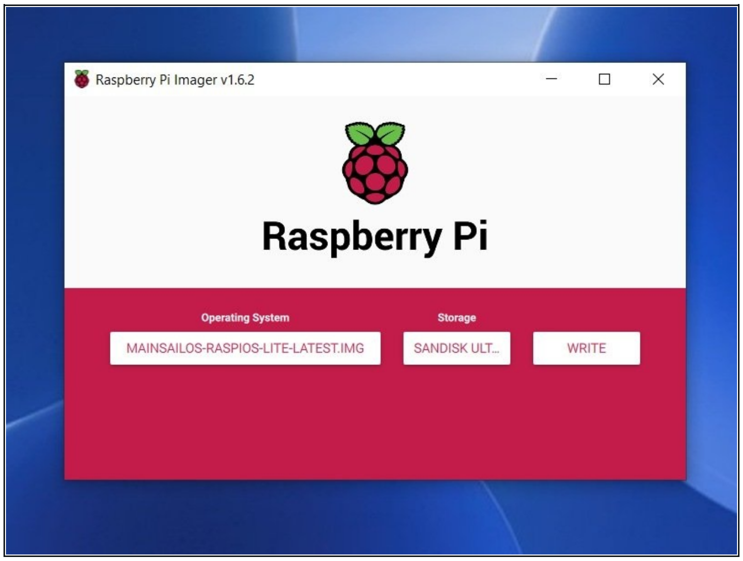
I’m using a Raspberry Pi4 8GB because I had a spare one laying around and I wanted to be able to connect to 5G wifi.

I’ve also chosen to use a flash drive instead of a SD card because the date transfer rate is 3-4x faster.

I’m powering the RPI with a Raspberry Pi Foundation approved power adapter

Project Link

Step 1 — Download/Setup



Operating System: MAINSAILOS-RASPIOS-LITE-LATEST.IMG

Storage: SANDISK ULT...

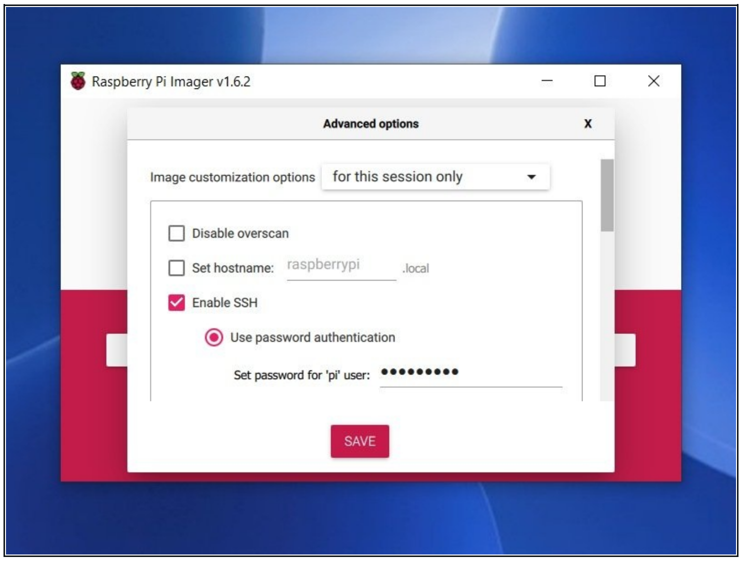


Image customization options: for this session only

☐ Disable overscan

☐ Set hostname: raspberrypi .local

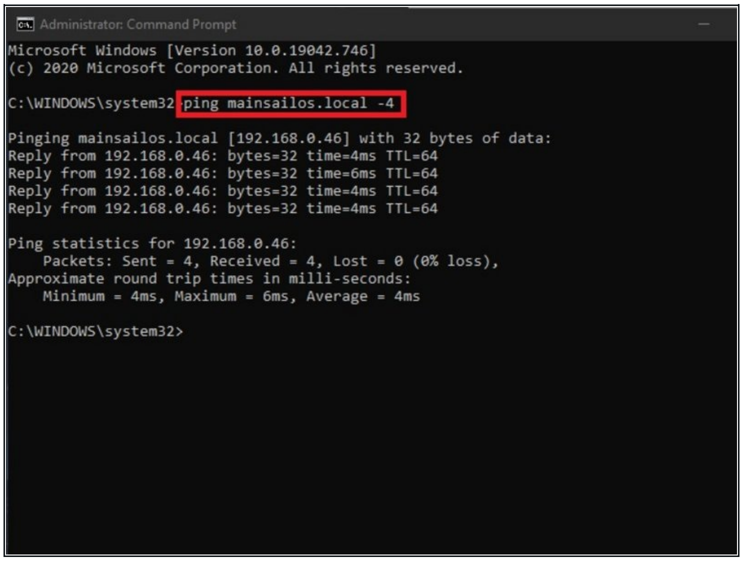
☑ Enable SSH

● Use password authentication

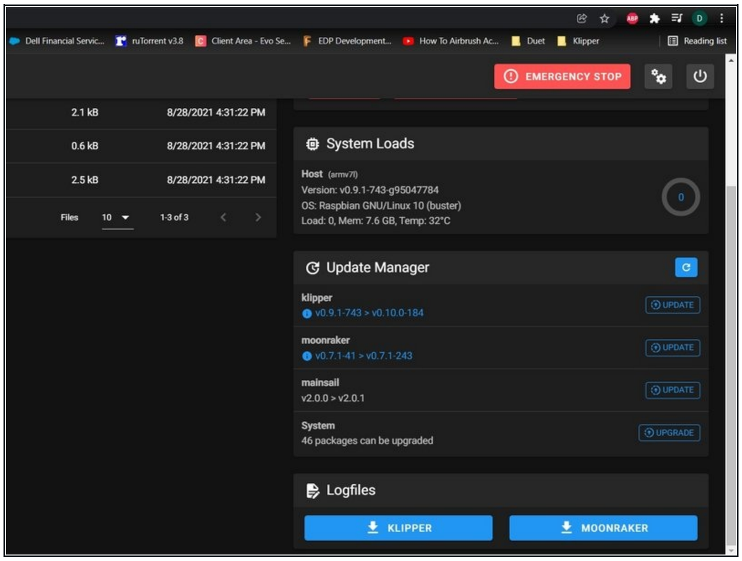
Set password for 'pi' user: •••••••••

- Download and extract the latest version of Mainsail from their Github
- Open the Raspberry Pi Imager
- Select Operating System > Use custom > Select the Mainsail OS you downloaded.
- Select Storage > Select your SD.
- Bring up the Advanced Settings menu by pressing "Ctrl + Shift + X".
- Here you can setup your WiFi, locale, enable SSH and numerous other options.
- Write > Yes to confirm
- After imaging is complete the SD is automatically ejected. Insert it into the RPI and power it on.

Step 2 — Configuring Mainsail



```
Microsoft Windows [Version 10.0.19042.746]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>ping mainsailos.local -4

Pinging mainsailos.local [192.168.0.46] with 32 bytes of data:
Reply from 192.168.0.46: bytes=32 time=4ms TTL=64
Reply from 192.168.0.46: bytes=32 time=6ms TTL=64
Reply from 192.168.0.46: bytes=32 time=4ms TTL=64
Reply from 192.168.0.46: bytes=32 time=4ms TTL=64

Ping statistics for 192.168.0.46:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 4ms, Maximum = 6ms, Average = 4ms

C:\WINDOWS\system32>
```



2.1 kB	8/28/2021 4:31:22 PM
0.6 kB	8/28/2021 4:31:22 PM
2.5 kB	8/28/2021 4:31:22 PM

Files 10 1-3 of 3

System Loads

Host (armv7l)
Version: v0.9.1-743-g95047784
OS: Raspbian GNU/Linux 10 (buster)
Load: 0, Mem: 7.6 GB, Temp: 32°C

Update Manager

klipper
v0.9.1-743 > v0.10.0-184 UPDATE

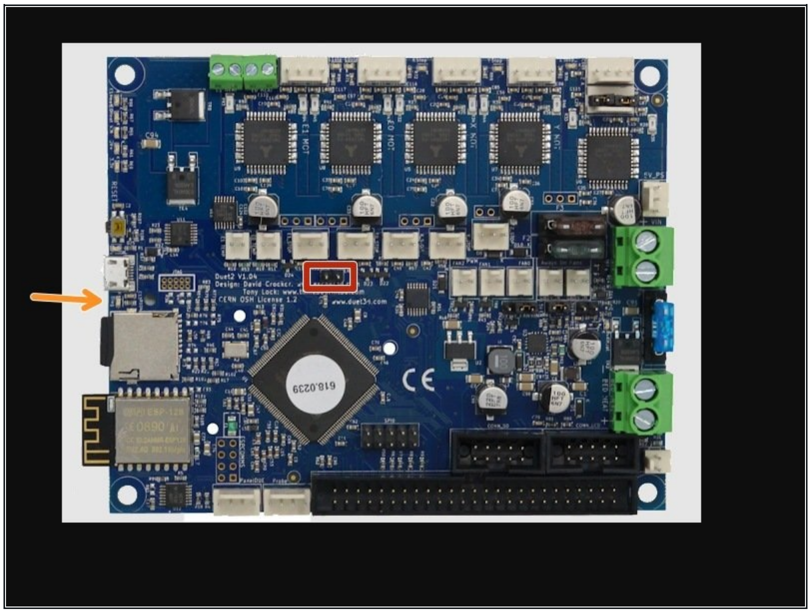
moonraker
v0.7.1-41 > v0.7.1-243 UPDATE

mainsail
v2.0.0 > v2.0.1 UPDATE

System
46 packages can be upgraded UPGRADE

- With the Raspberry Pi connected to the Duet 2 WiFi via USB, power on the RPI and SSH into the Pi.
- You can find the IP address of the Pi by looking at the connected client list in your router or by opening cmd.exe on your PC and running
  - ping mainsailos.local -4
- In a webpage enter http://mainsailos.local or your printers IP address
- Navigate to the Machine tab on the left.
- Select Upgrade to update the RPI image and Update to install updates for Klipper, Mainsail and Moonraker. Once the updates are complete reboot the RPI

Step 3 — Factory Reset Duet 2 WiFi



ⓘ Originally when I wrote this guide I saw many sources say you do not need to manually erase the firmware when you configure the Duet 2 WiFi for Klipper for the first time. However there's been many cases that has proven false.

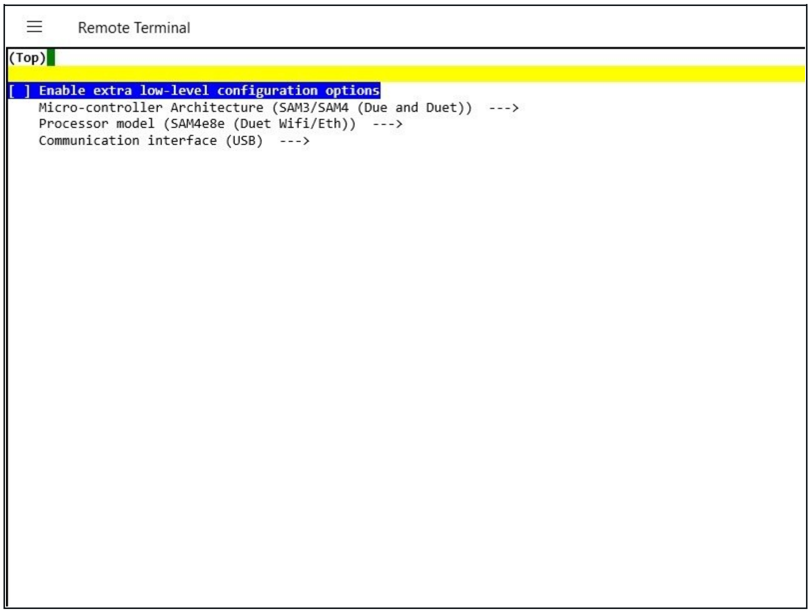
- With the board powered off place a jumper on the erase pins.

- Connect the USB cable from the Duet 2 WiFi to a 5v power source(your PCs USB Port).

- Once the diag light comes on remove the jumper

- Any firmware saved to the Duet has been erased.

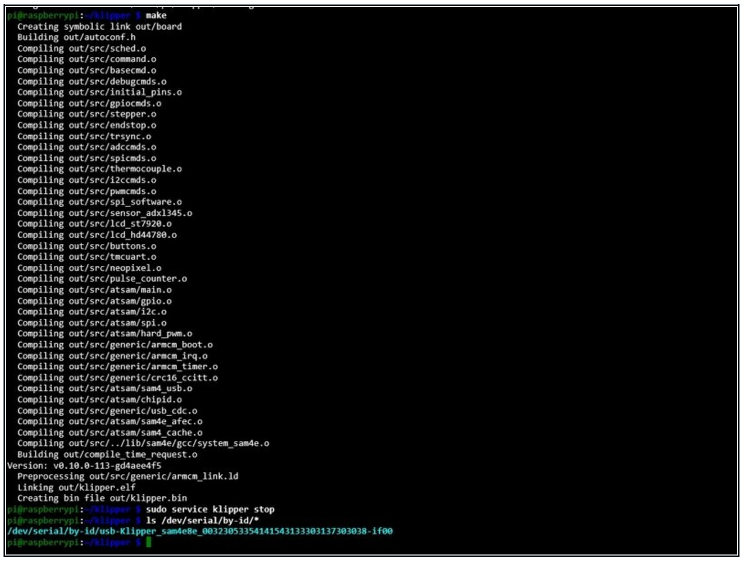
Step 4 — Flash Duet 2 WiFi 1



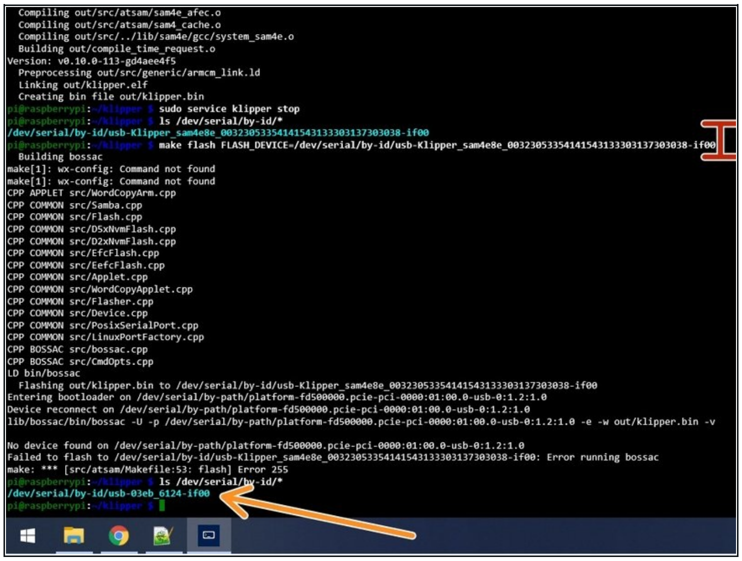
```
(Top)

[ ] Enable extra low-level configuration options
    Micro-controller Architecture (SAM3/SAM4 (Due and Duet))  --->
    Processor model (SAM4e8e (Duet Wifi/Eth))  --->
    Communication interface (USB)  --->
```

- Connect the Duet 2 WiFi to the Pi with a USB cable. Power on the Pi and SSH into it.
- cd klipper
- make clean
- make menuconfig
- Micro-Controller > SAM3/SAM4 (Due and Duet)
- Processor > SAM4e8e (Duet WiFi/Eth)
- Communication Interface > USB
- Press the escape key on your keyboard and enter Y to save the config

Step 5 — Flash Duet 2 WiFi 2



```
pi@raspberrypi:~/klipper $ make
  Creating symbolic link out/board
  Building out/autoconf.h
  Compiling out/src/sched.o
  Compiling out/src/command.o
  Compiling out/src/basecmd.o
  Compiling out/src/debugcmds.o
  Compiling out/src/initial_pins.o
  Compiling out/src/gpiocmds.o
  Compiling out/src/stepper.o
  Compiling out/src/endstop.o
  Compiling out/src/trsync.o
  Compiling out/src/adccmds.o
  Compiling out/src/spicmds.o
  Compiling out/src/thermocouple.o
  Compiling out/src/i2ccmds.o
  Compiling out/src/pwmcmds.o
  Compiling out/src/spi_software.o
  Compiling out/src/sensor_adxl345.o
  Compiling out/src/lcd_st7920.o
  Compiling out/src/lcd_hd44780.o
  Compiling out/src/buttons.o
  Compiling out/src/tmcuart.o
  Compiling out/src/neopixel.o
  Compiling out/src/pulse_counter.o
  Compiling out/src/atsam/main.o
  Compiling out/src/atsam/gpio.o
  Compiling out/src/atsam/i2c.o
  Compiling out/src/atsam/spi.o
  Compiling out/src/atsam/hard_pwm.o
  Compiling out/src/generic/armcm_boot.o
  Compiling out/src/generic/armcm_irq.o
  Compiling out/src/generic/armcm_timer.o
  Compiling out/src/generic/crc16_ccitt.o
  Compiling out/src/atsam/sam4_usb.o
  Compiling out/src/atsam/chipid.o
  Compiling out/src/generic/usb_cdc.o
  Compiling out/src/atsam/sam4e_afec.o
  Compiling out/src/atsam/sam4_cache.o
  Compiling out/src/../lib/sam4e/gcc/system_sam4e.o
  Building out/compile_time_request.o
Version: v0.10.0-113-gd4aee4f5
  Preprocessing out/src/generic/armcm_link.ld
  Linking out/klipper.elf
  Creating bin file out/klipper.bin
pi@raspberrypi:~/klipper $ sudo service klipper stop
pi@raspberrypi:~/klipper $ ls /dev/serial/by-id/*
/dev/serial/by-id/usb-Klipper_sam4e8e_00323053354141543133303137303038-if00
pi@raspberrypi:~/klipper $ 
```



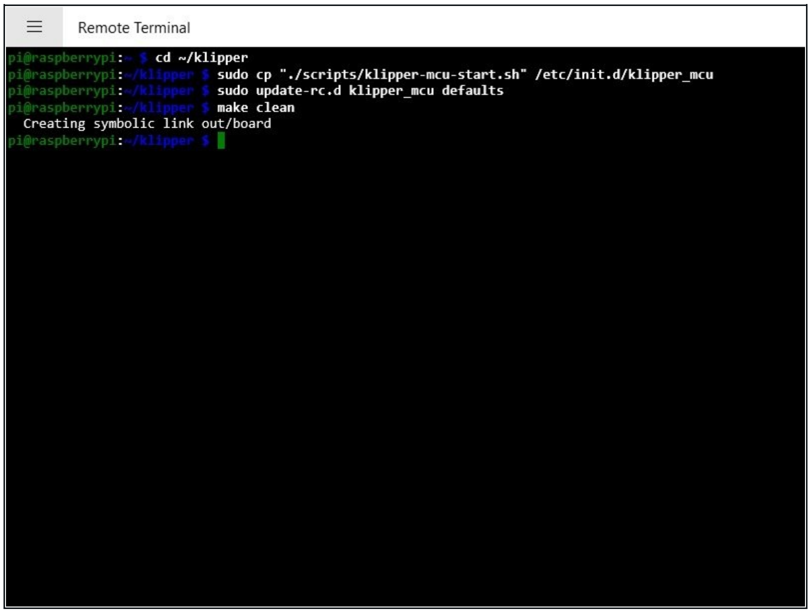
```
  Compiling out/src/atsam/sam4e_afec.o
  Compiling out/src/atsam/sam4_cache.o
  Compiling out/src/../lib/sam4e/gcc/system_sam4e.o
  Building out/compile_time_request.o
Version: v0.10.0-113-gd4aee4f5
  Preprocessing out/src/generic/armcm_link.ld
  Linking out/klipper.elf
  Creating bin file out/klipper.bin
pi@raspberrypi:~/klipper $ sudo service klipper stop
pi@raspberrypi:~/klipper $ ls /dev/serial/by-id/*
/dev/serial/by-id/usb-Klipper_sam4e8e_00323053354141543133303137303038-if00
pi@raspberrypi:~/klipper $ make flash FLASH_DEVICE=/dev/serial/by-id/usb-Klipper_sam4e8e_00323053354141543133303137303038-if00
  Building bossac
make[1]: wx-config: Command not found
make[1]: wx-config: Command not found
CPP APPLET src/WordCopyArm.cpp
CPP COMMON src/Samba.cpp
CPP COMMON src/Flash.cpp
CPP COMMON src/D5xNvmFlash.cpp
CPP COMMON src/D2xNvmFlash.cpp
CPP COMMON src/EfcFlash.cpp
CPP COMMON src/EefcFlash.cpp
CPP COMMON src/Applet.cpp
CPP COMMON src/WordCopyApplet.cpp
CPP COMMON src/Flasher.cpp
CPP COMMON src/Device.cpp
CPP COMMON src/PosixSerialPort.cpp
CPP COMMON src/LinuxPortFactory.cpp
CPP BOSSAC src/bossac.cpp
CPP BOSSAC src/CmdOpts.cpp
LD bin/bossac
  Flashing out/klipper.bin to /dev/serial/by-id/usb-Klipper_sam4e8e_00323053354141543133303137303038-if00
Entering bootloader on /dev/serial/by-path/platform-fd500000.pcie-pci-0000:01:00.0-usb-0:1.2:1.0
Device reconnect on /dev/serial/by-path/platform-fd500000.pcie-pci-0000:01:00.0-usb-0:1.2:1.0
lib/bossac/bin/bossac -U -p /dev/serial/by-path/platform-fd500000.pcie-pci-0000:01:00.0-usb-0:1.2:1.0 -e -w out/klipper.bin -v

No device found on /dev/serial/by-path/platform-fd500000.pcie-pci-0000:01:00.0-usb-0:1.2:1.0
Failed to flash to /dev/serial/by-id/usb-Klipper_sam4e8e_00323053354141543133303137303038-if00: Error running bossac
make: *** [src/atsam/Makefile:53: flash] Error 255
pi@raspberrypi:~/klipper $ ls /dev/serial/by-id/*
/dev/serial/by-id/usb-03eb_6124-if00
pi@raspberrypi:~/klipper $ 
```

- Enter make
- Stop the Klipper service.
  - sudo service klipper stop
- Get the Duet's serial ID # and copy it down.
  - ls /dev/serial/by-id/*
- Enter "make flash FLASH_DEVICE=" and paste your serial after the =

ⓘ If the flash fails usually this will place the board into a bootloader mode like original install does when you hit the erase pins. If this occurs please rerun ls /dev/serial/by-id/* so that you can get the correct serial

- Enter make "flash FLASH_DEVICE=" and paste your serial your new serial after the =

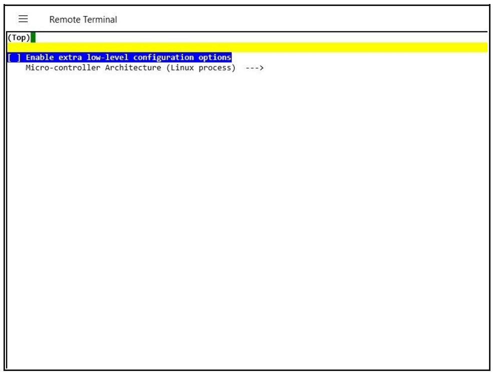
Step 6 — Flash Linux CPU 1



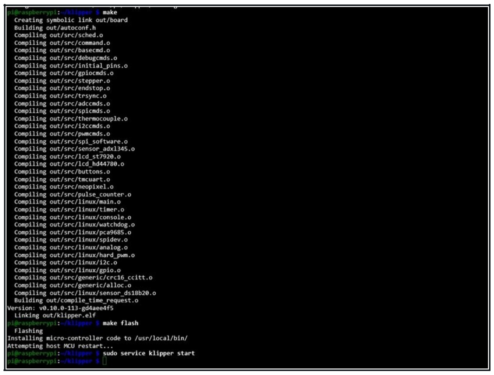
```
pi@raspberrypi:~ $ cd ~/klipper
pi@raspberrypi:~/klipper $ sudo cp "./scripts/klipper-mcu-start.sh" /etc/init.d/klipper_mcu
pi@raspberrypi:~/klipper $ sudo update-rc.d klipper_mcu defaults
pi@raspberrypi:~/klipper $ make clean
  Creating symbolic link out/board
pi@raspberrypi:~/klipper $ 
```

- Make sure you're in the klipper directory:
  - cd ~/klipper/
- Enter the following commands:
  - sudo cp ./scripts/klipper-mcu.service /etc/systemd/system/
  - sudo systemctl enable klipper-mcu.service
  - make clean
  - make menuconfig

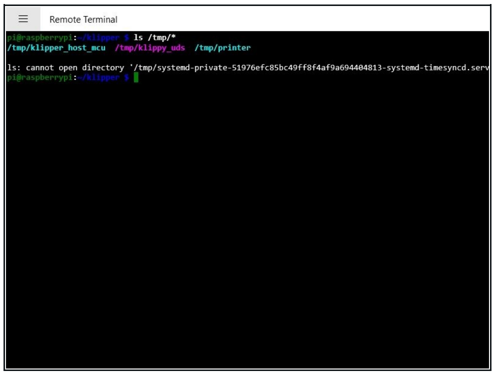
Step 7 — Flash Linux CPU 2



```
(Top)

[ ] Enable extra low-level configuration options
    Micro-controller Architecture (Linux process)  --->
```



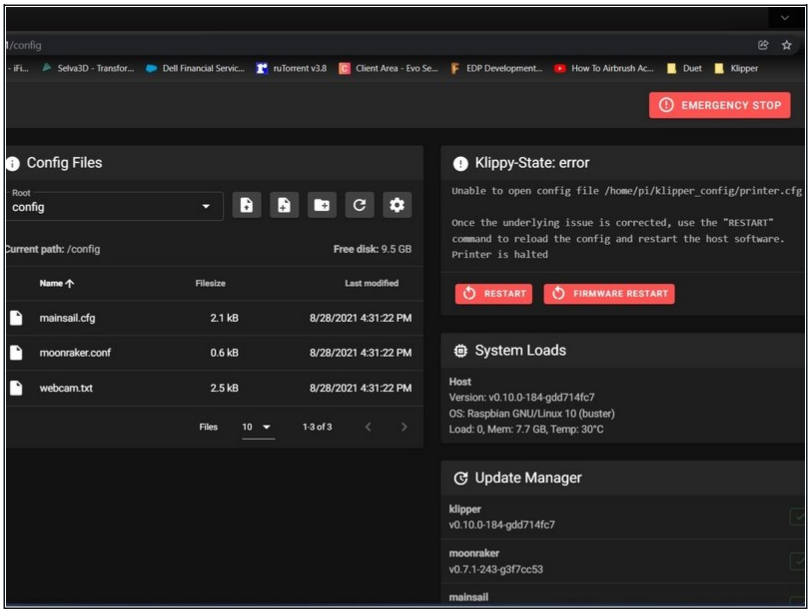
```
pi@raspberrypi:~/klipper $ make
  Creating symbolic link out/board
  Building out/autoconf.h
  Compiling out/src/sched.o
  Compiling out/src/command.o
  Compiling out/src/basecmd.o
  Compiling out/src/debugcmds.o
  Compiling out/src/initial_pins.o
  Compiling out/src/gpiocmds.o
  Compiling out/src/stepper.o
  Compiling out/src/endstop.o
  Compiling out/src/trsync.o
  Compiling out/src/adccmds.o
  Compiling out/src/spicmds.o
  Compiling out/src/thermocouple.o
  Compiling out/src/i2ccmds.o
  Compiling out/src/pwmcmds.o
  Compiling out/src/spi_software.o
  Compiling out/src/sensor_adxl345.o
  Compiling out/src/lcd_st7920.o
  Compiling out/src/lcd_hd44780.o
  Compiling out/src/buttons.o
  Compiling out/src/tmcuart.o
  Compiling out/src/neopixel.o
  Compiling out/src/pulse_counter.o
  Compiling out/src/linux/main.o
  Compiling out/src/linux/timer.o
  Compiling out/src/linux/console.o
  Compiling out/src/linux/watchdog.o
  Compiling out/src/linux/pca9685.o
  Compiling out/src/linux/spidev.o
  Compiling out/src/linux/analog.o
  Compiling out/src/linux/hard_pwm.o
  Compiling out/src/linux/i2c.o
  Compiling out/src/linux/gpio.o
  Compiling out/src/generic/crc16_ccitt.o
  Compiling out/src/generic/alloc.o
  Compiling out/src/linux/sensor_ds18b20.o
  Building out/compile_time_request.o
Version: v0.10.0-113-gd4aee4f5
  Linking out/klipper.elf
pi@raspberrypi:~/klipper $ make flash
  Flashing
Installing micro-controller code to /usr/local/bin/
Attempting host MCU restart...
pi@raspberrypi:~/klipper $ sudo service klipper start
pi@raspberrypi:~/klipper $
```



```
pi@raspberrypi:~/klipper $ ls /tmp/*
/tmp/klipper_host_mcu  /tmp/klippy_uds  /tmp/printer

ls: cannot open directory '/tmp/systemd-private-51976efc85bc49ff8f4af9a694404813-systemd-timesyncd.serv
pi@raspberrypi:~/klipper $
```

- Micro-Controller Architecture > Linux Process. Press esc than Y
  - make
  - make flash
  - sudo service klipper start
- Verify klipper_host_mcu is in the tmp directory with ls /tmp/*
- Add the pi to the TTY group
  - sudo usermod -a -G tty pi
- sudo reboot -h now

Step 8 — Loading Config



Current path: /config Free disk: 9.5 GB

Name	Filesize	Last modified
mainsail.cfg	2.1 kB	8/28/2021 4:31:22 PM
moonraker.conf	0.6 kB	8/28/2021 4:31:22 PM
webcam.txt	2.5 kB	8/28/2021 4:31:22 PM

```
Unable to open config file /home/pi/klipper_config/printer.cfg

Once the underlying issue is corrected, use the "RESTART"
command to reload the config and restart the host software.
Printer is halted
```

Host
Version: v0.10.0-184-gdd714fc7
OS: Raspbian GNU/Linux 10 (buster)
Load: 0, Mem: 7.7 GB, Temp: 30°C

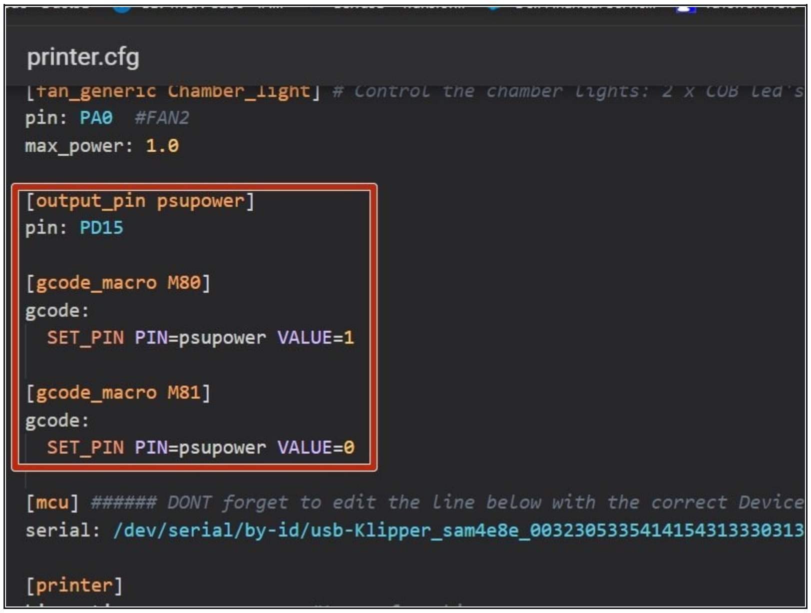
klipper
v0.10.0-184-gdd714fc7

moonraker
v0.7.1-243-g3f7cc53

mainsail

- Next we will go to a Web browser to the Pi's IP address. You will see an error for "Unable to open config file /home/pi/klipper_config/printer.cfg"
- Download the printer.cfg file from github and upload it to the config directory
- Open your newly created "printer.cfg". Scroll down to line 169. Make sure you update the config with your serial ID you got from "ls /dev/serial/by-id/*"
- Double check the mcu rpi serial ID in line 181 maches "ls /tmp/*"
- Click SAVE and restart the printer

Step 9 — Optional-Configure PS_ON

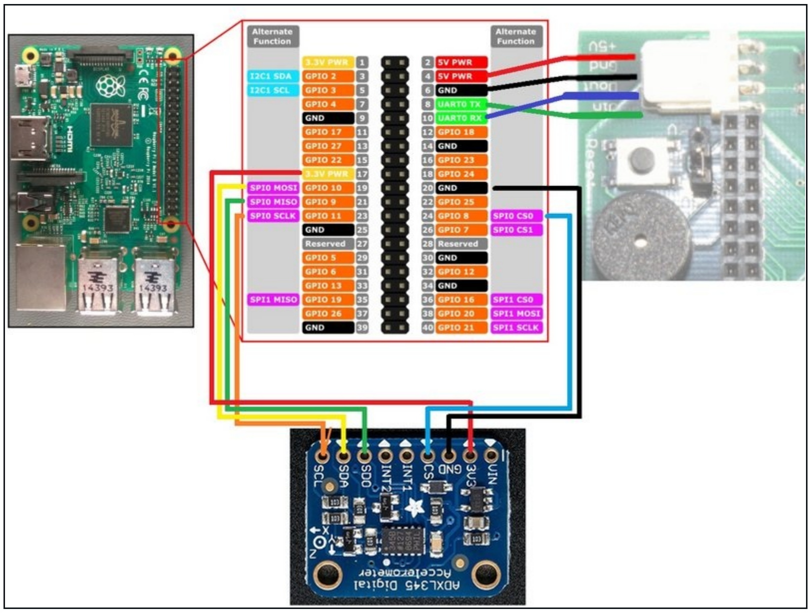


printer.cfg

```
[fan_generic Chamber_light] # Control the chamber lights: 2 x COB led's
pin: PA0  #FAN2
max_power: 1.0

[output_pin psupower]
pin: PD15

[gcode_macro M80]
gcode:
  SET_PIN PIN=psupower VALUE=1

[gcode_macro M81]
gcode:
  SET_PIN PIN=psupower VALUE=0

[mcu] ###### DONT forget to edit the line below with the correct Device
serial: /dev/serial/by-id/usb-Klipper_sam4e8e_0032305335414154313330313

[printer]
```

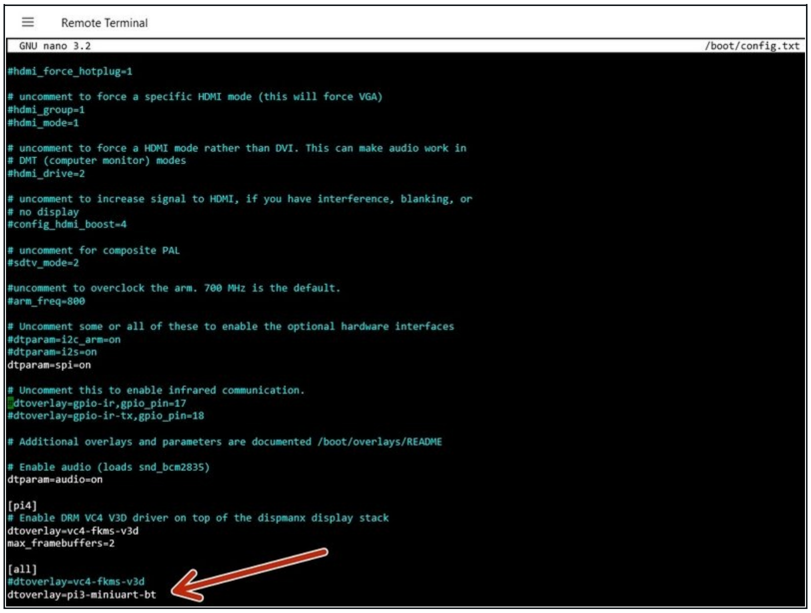
- If you use a separate 5v PSU for PS_ON You'll need to add the following lines
  - [output_pin psupower] pin: PD15
  - [gcode_macro M80] gcode: SET_PIN PIN=psupower VALUE=1
  - [gcode_macro M81] gcode: SET_PIN PIN=psupower VALUE=0

Step 10 — PanelDue 1



- PanelDue is compatible with Klipper but only with [v1.24](). Instructions can be found [here]()
- Connect the PanelDue to the RPI.
  - 🔴 5v > 5v
  - ● GND > GND
  - 🟢 UART0 TX > DIN
  - 🔵 UART0 RX > DOut

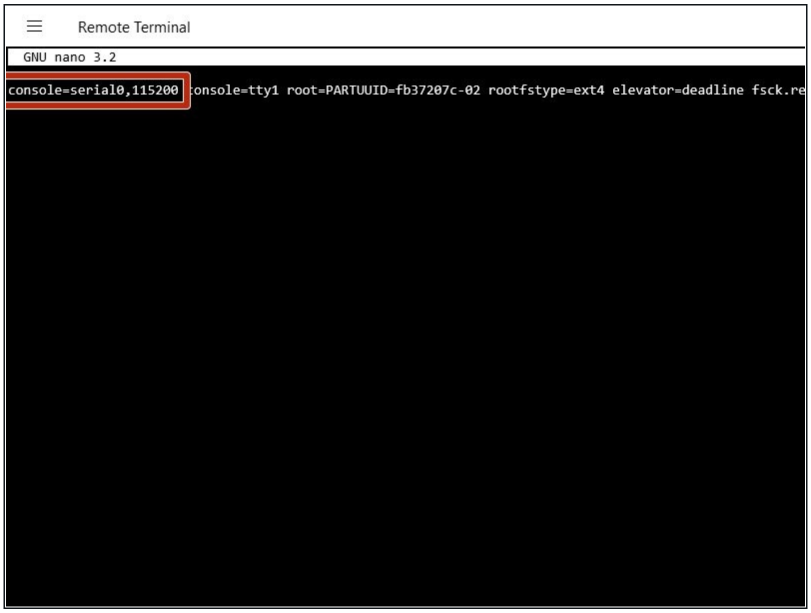
Step 11 — PanelDue Configuration 2



```
GNU nano 3.2                                                    /boot/config.txt

#hdmi_force_hotplug=1

# uncomment to force a specific HDMI mode (this will force VGA)
#hdmi_group=1
#hdmi_mode=1

# uncomment to force a HDMI mode rather than DVI. This can make audio work in
# DMT (computer monitor) modes
#hdmi_drive=2

# uncomment to increase signal to HDMI, if you have interference, blanking, or
# no display
#config_hdmi_boost=4

# uncomment for composite PAL
#sdtv_mode=2

#uncomment to overclock the arm. 700 MHz is the default.
#arm_freq=800

# Uncomment some or all of these to enable the optional hardware interfaces
#dtparam=i2c_arm=on
#dtparam=i2s=on
dtparam=spi=on

# Uncomment this to enable infrared communication.
#dtoverlay=gpio-ir,gpio_pin=17
#dtoverlay=gpio-ir-tx,gpio_pin=18

# Additional overlays and parameters are documented /boot/overlays/README

# Enable audio (loads snd_bcm2835)
dtparam=audio=on

[pi4]
# Enable DRM VC4 V3D driver on top of the dispmanx display stack
dtoverlay=vc4-fkms-v3d
max_framebuffers=2

[all]
#dtoverlay=vc4-fkms-v3d
dtoverlay=pi3-miniuart-bt
```

- SSH into the RPI.
  - sudo nano /boot/config.txt
- At the bottom of the config file enter:
  - dtoverlay=pi3-miniuart-bt
- Press Ctrl+X > Y to save > Enter to accept

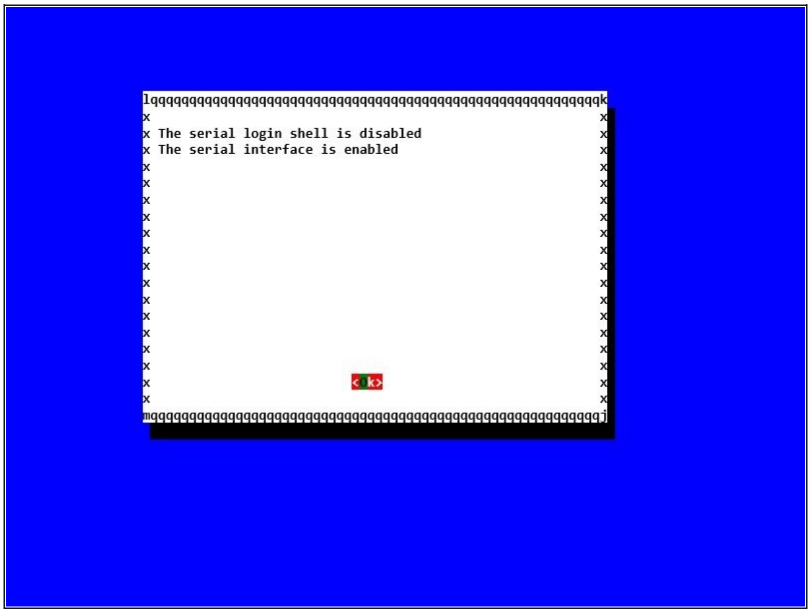
Step 12 — PanelDue Configuration 3



```
GNU nano 3.2
console=serial0,115200 console=tty1 root=PARTUUID=fb37207c-02 rootfstype=ext4 elevator=deadline fsck.re
```

- We need to disable the serial console
  - sudo nano /boot/cmdline.txt
  - Remove the following string
    - console=serial0,115200
  - Press Ctrl+X > Y to save > Enter to accept

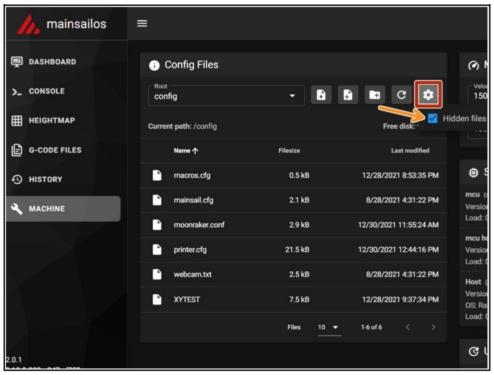
Step 13 — PanelDue Configuration 4



```
The serial login shell is disabled
The serial interface is enabled
```

- Disable serial console
  - Sudo raspi-config
  - 3 Interface Options > P6 Serial Port > No > Yes
- Yes to reboot the RPI
- Make sure you change the baud rate on the paneldue to 115200
- There's also a number of configs that need to be added to your moonraker.cfg. Those details can be found here.

Step 14 — BLV Mainsail Theme



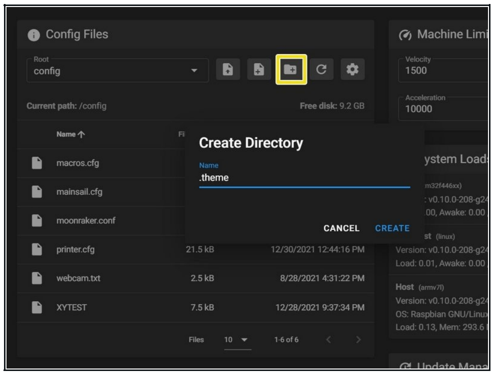
Config Files

Root: config

Current path: /config

Name	Filesize	Last modified
macros.cfg	0.5 kB	12/28/2021 8:53:35 PM
mainsail.cfg	2.1 kB	8/28/2021 4:31:22 PM
moonraker.conf	2.9 kB	12/30/2021 11:55:24 AM
printer.cfg	21.5 kB	12/30/2021 12:44:16 PM
webcam.txt	2.5 kB	8/28/2021 4:31:22 PM
XYTEST	7.5 kB	12/28/2021 9:37:34 PM

Files 10 1-6 of 6



Config Files

Root: config

Current path: /config

Free disk: 9.2 GB

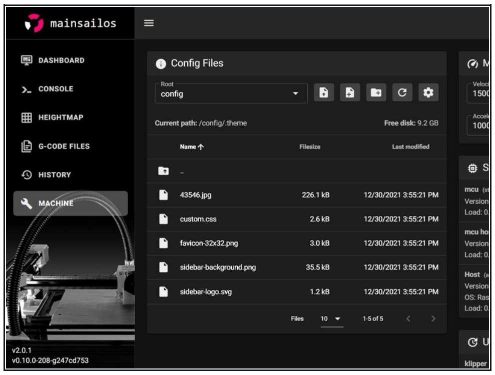
Create Directory

Name: .theme

Name	Filesize	Last modified
macros.cfg		
mainsail.cfg		
moonraker.conf		
printer.cfg	21.5 kB	12/30/2021 12:44:16 PM
webcam.txt	2.5 kB	8/28/2021 4:31:22 PM
XYTEST	7.5 kB	12/28/2021 9:37:34 PM

Files 10 1-6 of 6

Host (armv7l)
Version: v0.10.0-208-g24
OS: Raspbian GNU/Linux
Load: 0.13, Mem: 293.6



Config Files

Root: config

Current path: /config/.theme

Free disk: 9.2 GB

Name	Filesize	Last modified
..		
43546.jpg	226.1 kB	12/30/2021 3:55:21 PM
custom.css	2.6 kB	12/30/2021 3:55:21 PM
favicon-32x32.png	3.0 kB	12/30/2021 3:55:21 PM
sidebar-background.png	35.5 kB	12/30/2021 3:55:21 PM
sidebar-logo.svg	1.2 kB	12/30/2021 3:55:21 PM

Files 10 1-5 of 5

v2.0.1
v0.10.0-208-g247cd753

- Download the file and extract.
- Under "machine" menu, click on the Gear icon and select the checkbox for "show hidden files".
- Select "Create Directory".
- In the name field enter “.theme” and select CREATE.
- it has to be named as seen in the screenshot.
- Select the newly created ".theme" folder. Upload all the files from the zip archive into .theme folder.
- hit Ctrl+F5 and Enjoy.

I wasn’t paid to write these guides. Ben asked me to do it as a favor to him. Considering all that he’s done for our community I felt it was the least I could do. To be honest even though it took about 2 months to build and document, I had a lot of fun doing it. It forced me to write guides in a manner that was easy for everyone to understand and cleanup my Github so I could share any files that were used that aren’t part of the original download. You’re not obligated but if you would like, feel free to donate.