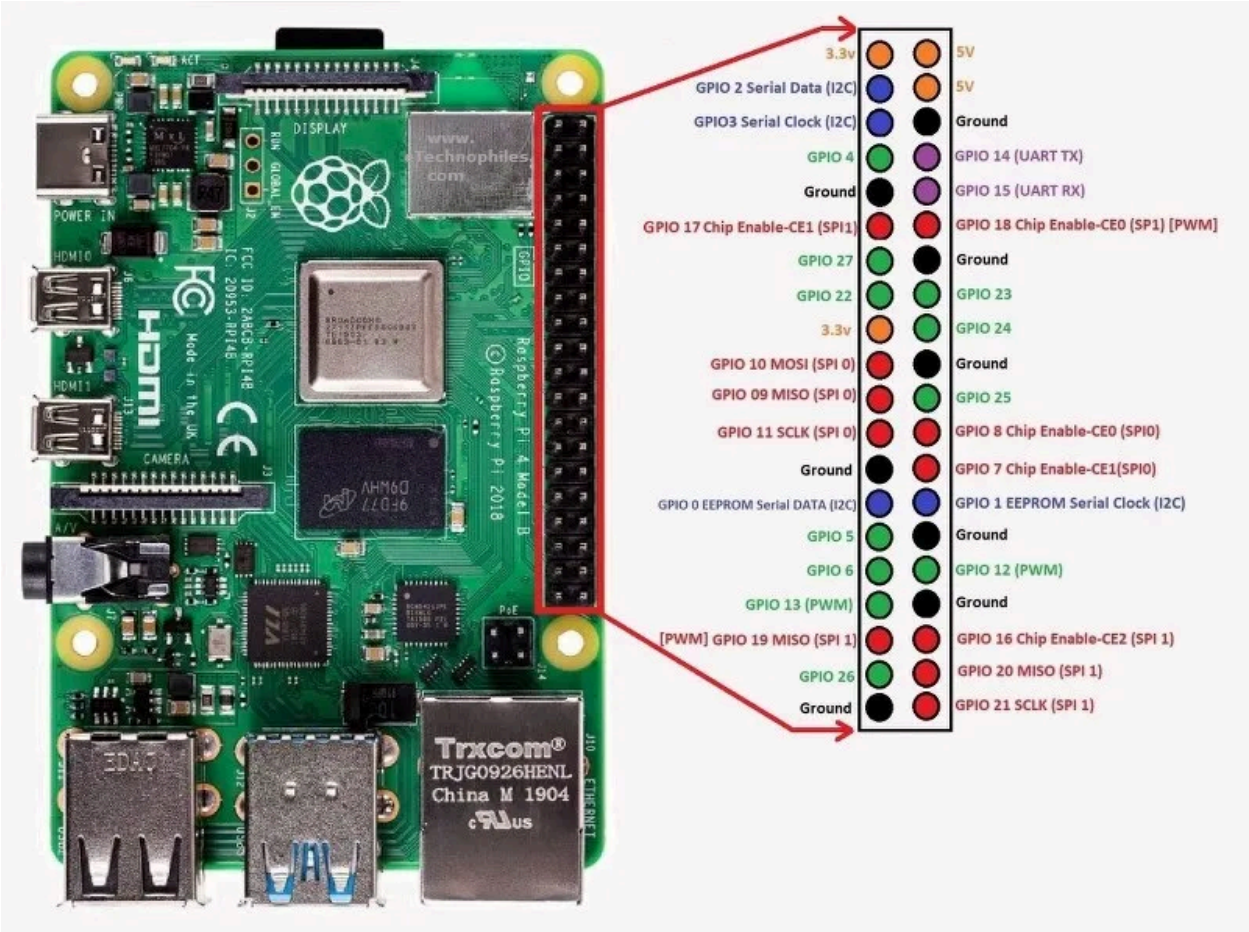
MKS Robin Nano v1.2 to Raspberry Pi 4 with internal serial & power.

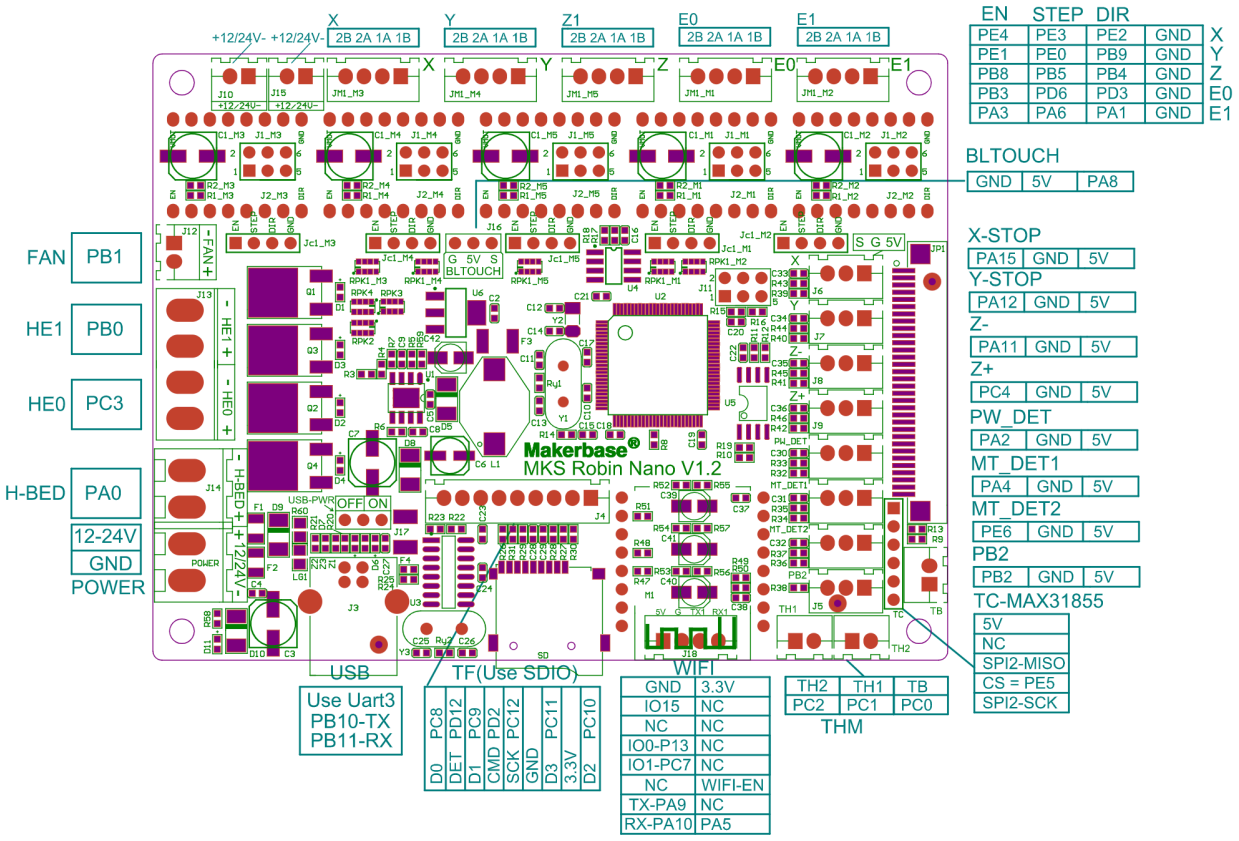
You can use OctoPrint and the “PSU control” plugin for power/sleep automation.

Credits: Weazool (thingiverse), Mpalpha (reddit)

Rpi4 pinout:



MKS Robin nano 1.2 diagram:



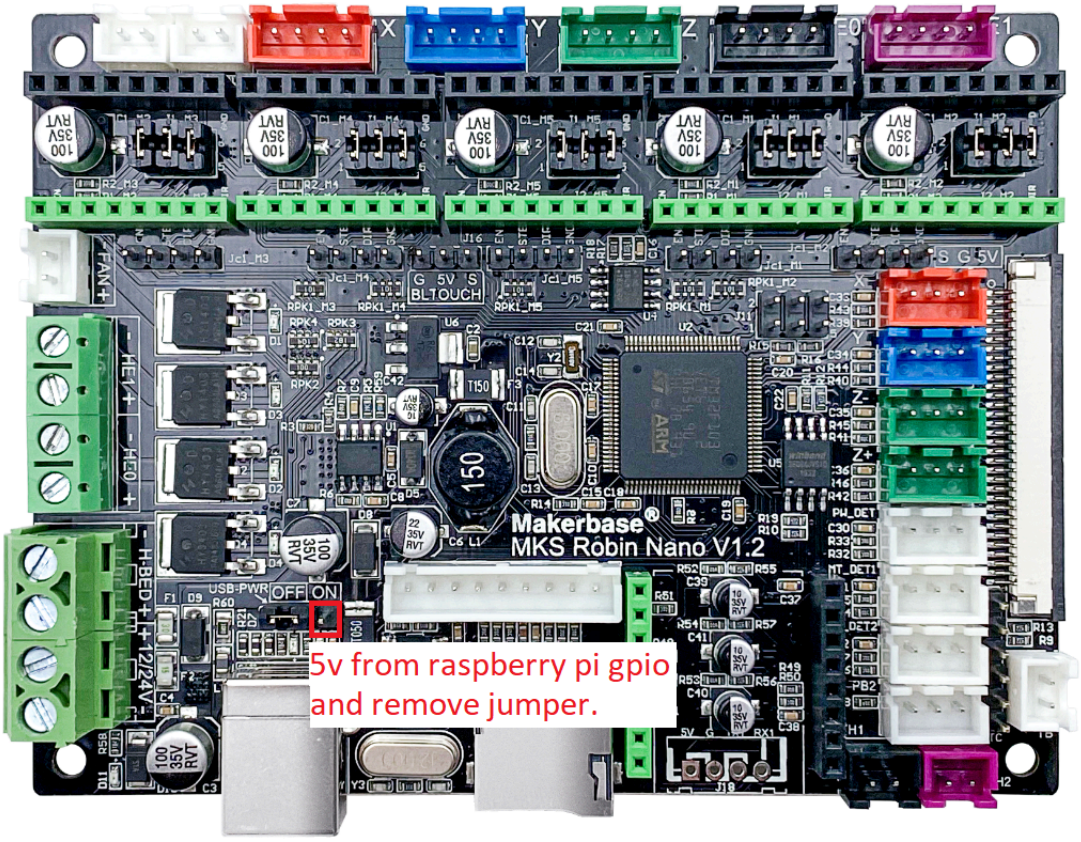
1.WIRING Rpi4 to MKS Robin Nano 1.2

TX gpio 14 (pin8) to RX PA10

RX gpio 15 (pin10) to TX PA9

GND (pin6) to GND

5v (either one) to USB-PWR->ON



1a. Remove the USB-PWR jumper, rpi4 will power the 5v mcu.

1b. Choose a power source for the raspberry pi 4, some suggested options are:

1. rpi4 power adapter to 120/240v PWC board input terminal using a cut ext cord for a plug.
2. buck converter from 12/24v power to 5v.

2.Octoprint config:

2.1 either SSH into the octopi.local or place the microsd on another machine and edit /boot/config.txt and add the following lines:

2.1.1 "enable_uart=1" and "dtoverlay=disable_bt" (or the legacy pi3-disable-bt, see /boot/overlays/README / this frees up ttyAMA0 from BT)

2.1.2 disable the service as well with "sudo systemctl disable hciuart"

2.2 remove the serial console with either 2.2a or 2.2b

2.2a edit /boot/cmdline.txt and remove the string "console=serial0,115200"

--or--

2.2b run "sudo raspi-config" > 3. interface options > p6. serial port > "would you like a login shell...." choose NO > "Would you like the serial port to be enabled" choose YES

2.3 Reboot

2.4 Install the “PSU Control” plugin and open it’s settings.

2.4.1 configure the following options:

a. Switching Method > G-Code Command
b. On G-Code Command M80
c. Off G-Code Command M81
d. Sensing > Internal
e. Power On Options check “Automatically turn PSU ON”
f. Power On Options > Trigger Commands > Add “M21” to the existing list
g. Power On Options > Post On GCode Script STATUS
h. Power On Options > check “Connect when powered on”
i. Power Off Options > check “Automatically turn PSU OFF When idle”
j. Power Off Options > Choose an Idle Timeout.
k. Power On Options > check “Disconnect on power off”

2.4.2 go to “GCODE Scripts” in octoprint settings

a. After serial connection to printer is established > M80
b. Before serial connection to printer is closed > M81

2.5.1 on the octoprint interface go to "settings" and then "serial connection" and add "/dev/ttyAMA0" in the "Additional serial ports" textbox

2.5.2 "ls /dev/ttyAMA*" and see if gives out a "/dev/ttyAMA0" on SSH

2.5.3 you can just connect GPIO 14 to 15 and click connect on the octoprint interface with ttyAMA0 selected - it will both send and receive the m110

Marlin Firmware Setup:

(you can browse around there to figure out how to build using vscode and platformIO)

in Configuration.h:

"#define SERIAL_PORT 1" (this is the port number for the wifi pins uart on PA9 and PA10)

"#define SERIAL_PORT_2 3" (this is for usb (optional) if you want that uart as well)

comment out with "//" the "#define MKS_WIFI_MODULE" line (right below the "#if ENABLED(TFT_LVGL_UI)" (it's gonna mess up the connection)

if you get any error messages from sanithcheck.h or hal.h or something else just comment out the #error line

if it's preventing you from using the serial ports above

Klipper Firmware Setup:

Install the “OctoKlipper” plugin for octoprint.

SSH into the raspberry pi and run:

a. “cd klipper”
b. “make menuconfig”
c. Choose the following options:
    i. Enable extra low-level configuration options
    ii. Micro-controller Architecture (STMicroelectronics STM32)
    iii. Processor Model (STM32F103)
    iv. Bootloader offset (28KiB bootloader)
    v. Clock Reference (8MHz crystal)
    vi. Communication interface (Serial (on USART1 PA10/PA9))
    vii. (250000) Baud rate for serial port
    viii. (!PC6,!PD13) GPIO pins to set at micro-controller startup
d. Quit and save.
e. “sudo service klipper stop”
f. “make”
g. “sudo service klipper start”
h. “./scripts/update_mks_robin.py out/klipper.bin out/Robin_nano35.bin”

Winscp into the raspberry pi and download “out/Robin_nano35.bin” to a microsd card.

Flash the firmware to the printer, the display will turn off when finished.

Add the following to your klipper “printer.cfg”

```
[static_digital_output reset_display]
pins: !PC6, !PD13 # disable unsupported display

[mcu]
serial: /dev/ttyAMA0
restart_method: command

#for elf pwc v2.0 (model 2020+)
#use psu control plugin and externally power rpi
#set auto on and off in psu control plugin
#set mcu power to usb 5v via jumper on robin nano v1.2
#on gcode command = SET_PIN PIN=power VALUE=0
#off gcode command = SET_PIN PIN=power VALUE=1
[output_pin power]
pin:!PB2
#value: 0.0
shutdown_value: 1.0

# This adds the 'respond' G-Code that you can use to send M118 commands back to host
[respond]

[gcode_macro NOTIFY]
# Notify macro
gcode:
  {% if 'MSG' in params %}
    RESPOND TYPE=command MSG="action:notification {params.MSG}"
  {% endif %}

[gcode_macro M80]
gcode:
  SET_PIN PIN=power VALUE=0
  STATUS

[gcode_macro M81]
gcode:
  SET_PIN PIN=power VALUE=1
  STATUS
```