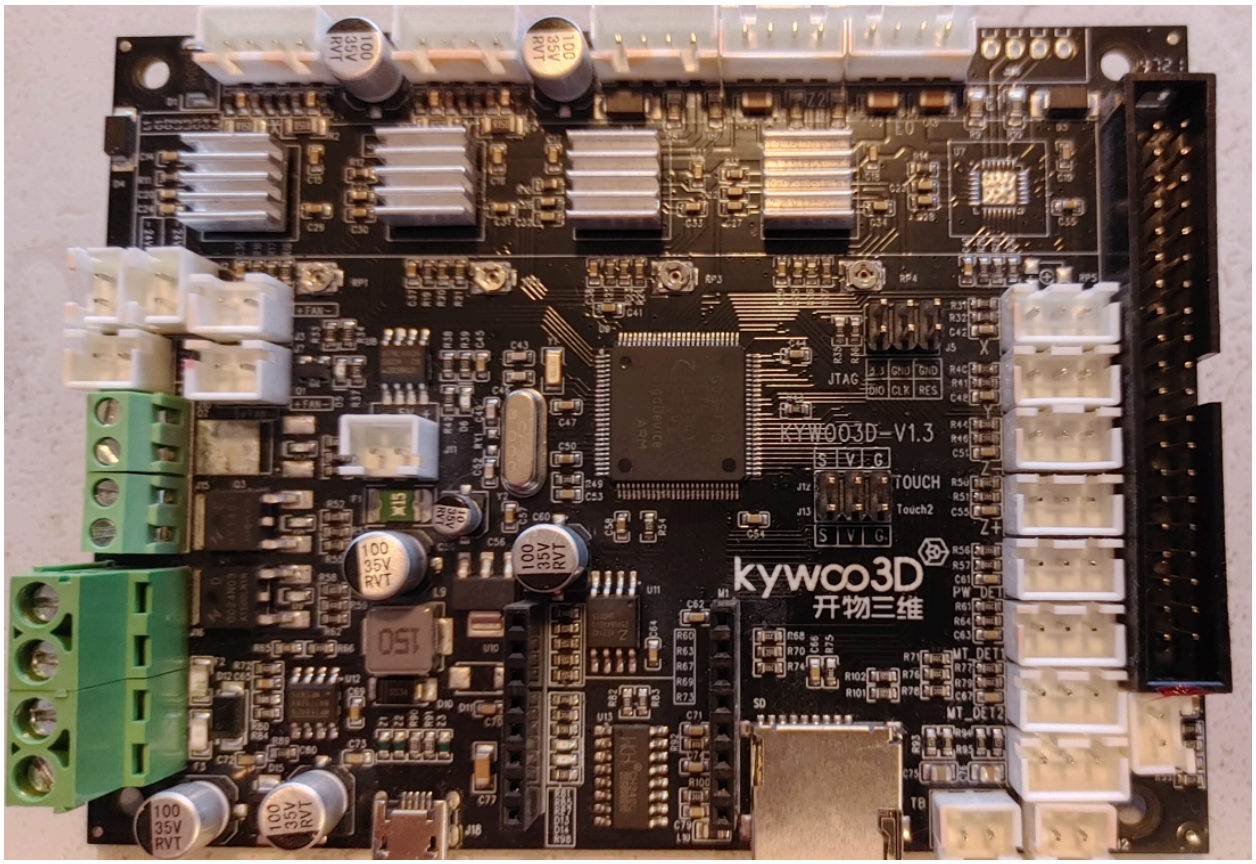
This project is intended to enable flashing of firmware for klipper on the stock mainboard of a KyWoo3D Tycoon slim. The mainboard is marked KwWoo3D V1.3 and I believe they are a slightly customized clone of an MKS Robin Nano V2. They use the GD32F303 VET6 chip and the CH340C chip instead of direct USB. This guide assumes you are familiar with installing and setting up Klipper (I recommend KIAUH) and this guide is only to get this board working with your klipper installation.



.

**** Special notes for Klipper setup: You can use the included printer.cfg to get pins setup but the rest will still need to be set (ie offsets and build volume etc) by you. Pins are virtually identical to Robin Nano V2. As a bonus you can use pin PB2 (PW_OFF) for neopixels. I am currently looking to see what other pins might work and I will update if I find any.

[neopixel my_neopixel]
pin: PB2
chain_count: 1
color_order: GRB # or RGB, depending on your LED type

Klipper Firmware Flashing Guide for Kywoo3D Tycoon Slim

Printer & Mainboard Details

Model: Kywoo3D Tycoon Slim

Mainboard: MKS Robin Nano V1.3 clone (likely based on Robin Nano V2)

Processor: GD32F303

Flashing Dependency: Original LCD MIGHT need to be connected to flash properly. Flashing was inconsistent and unreliable if LCD was not connected during flashing but it is NOT needed during normal operation once flashing is complete.

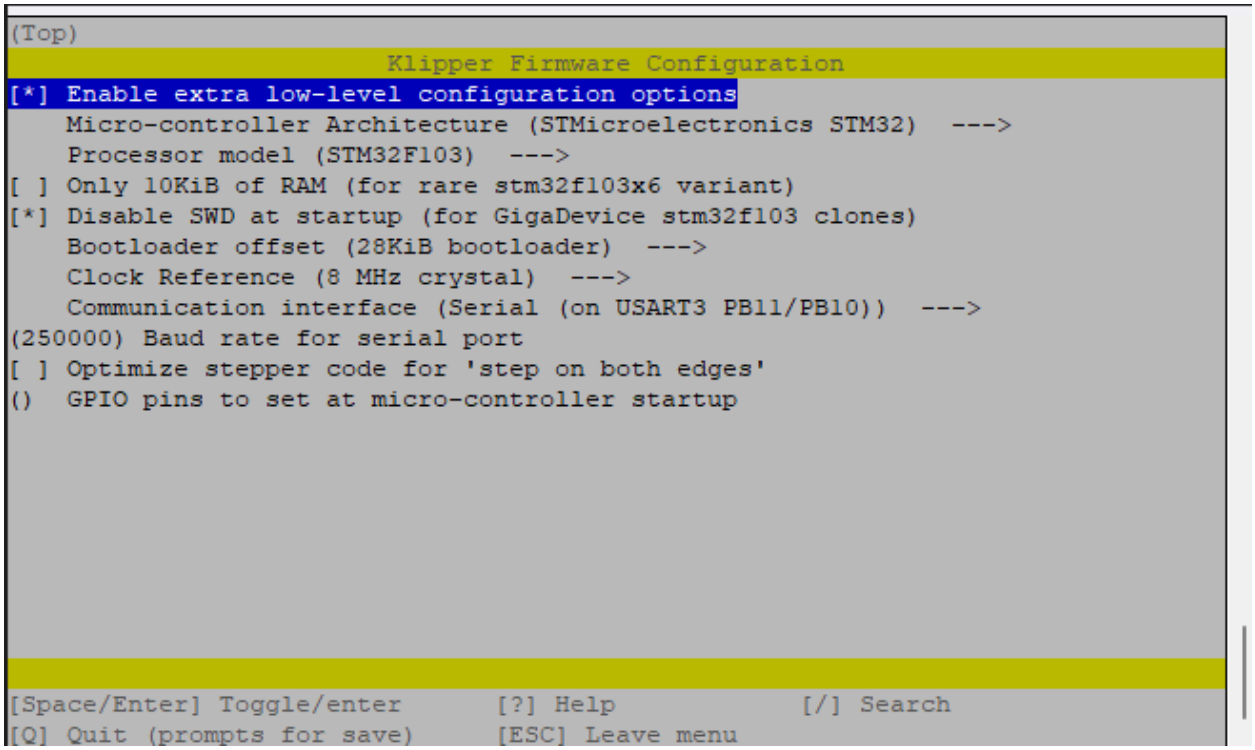
Firmware Name: Robin_nano35.bin

Successful Flashing: Extension changes from .bin → .CUR

SD Card Requirements: Ideally 4GB or smaller, Formatted as FAT32, Root folder must be empty

Make Menuconfig Settings for Klipper

✅ Enable Extra Low-Level Options
✅ Microcontroller Architecture: STM32
✅ Processor Model: STM32F103
✅ Disable SWD at startup (for GigaDevices)
✅ Bootloader Offset: 28kB
✅ Clock Reference: 8MHz
✅ Communication Mode: Serial (USART3 on PB11/PB10)
✅ Baud Rate: 250000
🚫 DO NOT enable "Optimize stepper code for step on both edges"
🚫 No GPIO pins set



Flashing Instructions:

At prompt run:

make menuconfig.
make clean
make
./scripts/update_mks_robin.py out/klipper.bin out/Robin_nano35.bin

📌 **Important:** Copy Robin_nano35.bin to a 4GB or smaller FAT32 SD card with an empty root folder. Typically I use WinSCP for this.

After you have the firmware on the SD card it is time to flash the mainboard. I encountered a LOT of trouble with this and found that if the original LCD was not connected flashing almost never succeeded. When the LCD is connected the flashing process will take approximately 10 seconds and you will see an “Updating” message on the LCD showing the percentage increasing from 0 to 100%. At 100% flashing is complete and successful. Power off the board, remove the SD card and disconnect the LCD. You can theoretically leave the LCD connected but it will only show “Booting” and it may actually interfere with USB/serial connections so I STRONGLY advise you remove it. Since it is not going to work with klipper there is really no point in leaving it connected.

*** I encountered a few cases where flashing failed and the board APPEARED to be bricked. First off DON'T PANIC. ***

Recovery Steps if Flashing Fails:

Try reflashing the original stock Marlin firmware (I used version 2.1.3) with LCD connected. You should see the same "Updating" message showing a growing percentage from 0-100% but it will take upwards of a minute to complete. When flashing is complete you should be in the Marlin UI. Browse into the settings and select the option to Reinitialize EEPROM. Reboot the board and if it loads into the Marlin UI you are good to try flashing Klipper firmware again. This is where I found that leaving the LCD connected during klipper firmware flashing was most important. Without it connected, flashing failed often and "appeared" to brick the board.

After flashing you should be able to open Mainsail and begin to edit your Printer.cfg. I encountered many issues with a stable MCU connection on both a linux mint PC and a BTT Pi+CB1 which appear to be mostly around the board's use of the CH340C chip. This created an unstable Klipper Serial Connection causing the USB to repeatedly disconnect. If you encounter this problem please employ the commands below. Essentially you will need to remove BRLTTY, manually install the CH340 driver, force recognition of the CH340 chip and then specify the MCU in the pinter.cfg.

1️⃣ **Check if CH340 is detected:**

lsusb | grep -i ch340

Expected output:

Bus 001 Device 005: ID 1a86:7523 QinHeng Electronics CH340 Serial Adapter

2️⃣ **Remove BRLTTY (if blocking USB serial):**

sudo apt remove brltty

3️⃣ **Reinstall CH340 driver manually:**

git clone https://github.com/juliagoda/CH341SER
cd CH341SER
sudo make clean
sudo make

sudo make load

4️⃣ **Force Linux to recognize CH340 chip:**

sudo modprobe usbserial vendor=0x1a86 product=0x7523

5️⃣ **Verify serial port /dev/ttyUSB0 exists:**

ls /dev/ttyUSB*

6️⃣ **Update printer.cfg**:

[mcu}
serial: /dev/serial/by-id/usb-1a86_USB2.0-Serial-if00-port0 (your port might be different)
Baud: 250000 (same baud rate specified in make menuconfig)
restart_method: command

7️⃣ **Restart Klipper and confirm connection:**

sudo service klipper restart
journalctl -u klipper -n 50 --no-pager

I also experienced a constant USB disconnect issue and this was traced to the USB Auto-Suspend function. If you experience MCU connection failures and your klippy log shows disconnects you may want to try disabling Auto_suspend as well. **Side note:** this also resolved an issue I had with a 7” HDMI touch screen that would constantly lose touch function until it was shut off and restarted.

1️⃣ **Disable Auto-Suspend for the Specific Device:**

Identify USB device path using:

ls /sys/bus/usb/devices/*

2️⃣ force it to stay powered on with:

echo 'on' | sudo tee /sys/bus/usb/devices/2-1.2/power/control

*** This ensured that the CH340 USB-to-serial adapter wouldn't automatically suspend

3️⃣ **Check System-Wide USB Auto-Suspend Settings**

verifiy whether auto-suspend was active in the kernel with:

cat /sys/module/usbcore/parameters/autosuspend

4️⃣ **If auto-suspend was enabled, disable it globally using:**

sudo echo -1 > /sys/module/usbcore/parameters/autosuspend

**** This ensured all USB devices stay active instead of sleeping.

5️⃣ **Restart Klipper and Verify Stability**

After making these changes, restart Klipper to confirm the connection is stable:

sudo systemctl restart klipper

*** These steps prevented USB from cutting power to the CH341 adapter, allowing Klipper to maintain a stable connection.

Lastly you might find it is required to specify the baud rate in your MCU section in the printer.cfg. It should be the same as the baud rate specified in your make menuconfig (250000 is default). Without this I could not get my MCU to connect and remain connected. The micro USB cable was important as well. I tried 4 different micro USB cables before I found one that worked reliably and consistently. Oddly enough ALL of them worked with this very same printer with Marlin when connected to a PC running Pronterface. Basically…don't cheap out. Use good, shielded cables.