

Meshtastic on the Raspberry Pi

Do Not Buy the UART HAT

Also, hats based on the SX1302/SX1303 chip are not supported.

Binary is named meshtasticd, and requires running either as root, or as a user with permissions to access GPIO, SPI, and other interfaces.

.deb package is available as part of the release. This installs the binary, systemd service, and config file. .deb is compiled for Debian Bookworm, and will not install on Bullseye.

This binary could in theory work with any arm64 device, but has only been tested for the Raspberry Pi at this point.

Uses portduino to run the meshtastic firmware under Linux

So far has been tested on the Pi4, Pi400, and Pi5

[2024-01-22] Tested on Ubuntu 22.04 X86_64 by markbirss using CH341-SX1262 diy dongle

[2024-02-01] Tested on Debian GNU/Linux trixie/sid riscv64 Cvitek. CV180X ASIC. C906.(Milk-V Duo) on SPI bus

Pi5 uses gpiochip4, Pi 4/400 uses gpiochip0

For SPI support, add the following to your config.txt, which may be located at

/boot/config.txt or /boot/firmware/config.txt:

```
dtparam=spi=on
```

```
dtoverlay=spi0-0cs
```

For I2c support:

```
dtparam=i2c_arm=on
```

The meshtasticd configuration is at /etc/meshtasticd/config.yaml by default.

If a config.yaml is found in the current directory, that takes precedence. And a config file specified with the -c/--config option has the highest precedence.

/root/.portduino/default/prefs/

WIP: The config has some control over debugging levels.

Uncomment the sections that correspond to your hardware. Note that the yaml configuration is white-space sensitive.

Tested working so far for radio is the

Waveshare SX126X XXXM SPI interface
Adafruit RFM9x
Elecrow Lora RFM95 IOT

Stacking hats does work, but beware of pin conflicts. For instance, the Waveshare LoRa hat uses pin 18 for radio reset, and the Waveshare display uses pin 18 as the backlight toggle. Setting up both pins will cause the radio to stop responding when the backlight is turned off. In that specific case, the two hats can be used together by not defining the backlight pin in the config file. Other combinations may just not work at all.

The Pi does support I2c displays (SDA=02, SCL=03) and GPIO input for the user button. Tested is the
Adafruit 128x64 OLED Bonnet for Raspberry Pi

DPI displays generally use too many GPIO pins to be usable with any other hardware.

SPI displays do work. Tested is the
waveshare 2.8inch Screen for Raspberry Pi Resistive Touch TFT LCD with 320x240 Resolution
Touchscreen does work

Also working is keyboard input, handled like a t-deck input. Tested working is the Pi400 built-in keyboard.

GPS hats over UART do work. The Waveshare Lora + GPS hat has been confirmed working. On Pi5 you need to enable the UART on the 40pin GPIO header by adding

```
enable_uart=1  
dtoverlay=uart0
```

To /boot/firmware/config.txt

The right port for the UART GPS after a reboot is /dev/ttyAMA0

Persistence

The persistent .proto db files of the portduino version of meshtasticd are stored under:
/root/.portduino/default/prefs/

Reboot behavior

CLI Configuration (via TCP port 4403):

```
$ meshtastic --host localhost ...
```

To build the meshtasticd program under linux it is required to install two system libraries:

```
$ sudo apt install libgpiod-dev libyaml-cpp-dev
```

To install the meshtasticd debian package program under linux:

```
$ sudo apt install ./meshtasticd_{version}arm64.deb
```

Pinedio

Fedora was very odd. Debian 12 runs great. Anything shipping libgpiod 2.x won't work until we finish writing support.

Kernels equal to or earlier than 5.16 seems to have a couple issues. One being that they need to blacklist ch341 to avoid a different driver claiming the Pinedio device. Kernels 6.0 and newer seem to work more reliably.

The pinedio seems to have a problem with its crystal oscillator heating. Using LONG_FAST, messages longer than about 56 bytes were failing CRC checks.

The MEDIUM_FAST preset seems to work flawlessly

Meshtastic master native-unity branch. Target native (requires system libs libgpiod-dev, libyaml-cpp-dev, and libbluetooth-dev)

Before you start, check for /dev/gpiochipx devices. You'll want to use the one that gets added by the driver.

Also check for existing SPI devices in /sys/class/spi_master/

```
git clone https://github.com/frank-zago/ch341-i2c-spi-gpio.git
```

```
cd ch341-i2c-spi-gpio
```

```
make
```

```
sudo insmod ch341-core.ko
```

```
sudo insmod gpio-ch341.ko
```

```
sudo insmod i2c-ch341.ko
```

```
sudo insmod spi-ch341.ko
```

```
echo "spidev 1" | sudo tee /sys/class/spi_master/spi0/new_device
```

```
echo spidev | sudo tee /sys/bus/spi/devices/spi0.1/driver_override
```

```
echo spi0.1 | sudo tee /sys/bus/spi/drivers/spidev/bind
```

In /etc/meshtasticd/config.yaml:

Lora:

Module: sx1262

CS: 0

IRQ: 10

Busy: 11

gpiochip: x

spidev: spidev0.1

Super duper **rough** initial formatting. Please redline/modify as necessary. Don't hold back.

id: linux-native

title: Meshtastic on Linux-Native Devices

sidebar_label: Linux Native Devices

sidebar_position: 11

This guide outlines the setup of Meshtastic on Linux-native devices, utilizing portduino to run the Meshtastic firmware under Linux.

Prerequisites and Hardware Compatibility

Before proceeding with the setup, ensure the device meets the following requirements:

Tested Devices

- Raspberry Pi zero, zero 2 , 3,4, Pi 400, and Pi 5
- Ubuntu 22.04 X86_64 with a CH341-SX1262 DIY dongle
- Debian GNU/Linux trixie/sid riscv64 on Cvitek CV180X ASIC, C906 (Milk-V Duo), on the SPI bus

Hardware Compatibility

- Tested radios include Waveshare SX126X, Adafruit RFM9x, and Elecrow Lora RFM95 IOT.
- Support for I2C displays, SPI displays, and keyboard input has been confirmed. It is necessary to be aware of potential pin conflicts when stacking hats.
- UART HATs and SX1302/SX1303 chip-based HATs are not supported.

System Requirements

- The Meshtastic binary, `meshtasticd`, necessitates root access or a user with permissions to access GPIO, SPI, and other interfaces.
- A Linux distribution compatible with the Meshtastic installation package, which is compiled for Debian Bookworm and not compatible with Bullseye.

Installation

Installing Meshtasticd

- Necessary system libraries should be installed before building or installing Meshtastic.

```
```bash
```

```
sudo apt install libgpod-dev libyaml-cpp-dev
```

```
```
```

- The .deb Package is available as part of the release, installing the binary, a systemd service, and a config file. It is compiled for Debian Bookworm and incompatible with Bullseye.

```
```bash
sudo apt install ./meshtasticd_{version}arm64.deb
```
```

Configuration

Hardware Interfaces

For devices requiring SPI or I2C:

- SPI support can be enabled in `/boot/config.txt` or `/boot/firmware/config.txt`:

```
```plaintext
dtparam=spi=on
dtoverlay=spi0-0cs
```
```

- I2C support is enabled with:

```
```plaintext
dtparam=i2c_arm=on
```
```

Meshtasticd Configuration

- The meshtasticd configuration is at `/etc/meshtasticd/config.yaml` by default. If a `config.yaml` is found in the current directory, that takes precedence. And a config file specified with the `-c/--config` option has the highest precedence.

GPS Support

- Enabling UART for GPS hats on Pi5 requires modifications in `/boot/firmware/config.txt`:

```
```plaintext
enable_uart=1
dtoverlay=uart0
```
```

The correct port for UART GPS after a reboot is `/dev/ttyAMA0`.

Persistence

- The persistent .proto db files of the portduino version of meshtasticd are stored under:
`/root/.portduino/default/prefs/`.

Advanced Setup and Troubleshooting

- Installation of drivers for CH341 is required for Ubuntu 22.04 and other systems for SPI/I2C/GPIO support.

```
```bash
git clone https://github.com/frank-zago/ch341-i2c-spi-gpio.git
cd ch341-i2c-spi-gpio
make
sudo insmod ch341-core.ko
sudo insmod gpio-ch341.ko
sudo insmod i2c-ch341.ko
sudo insmod spi-ch341.ko
```
```

- Devices with kernels older than 5.16 may need to blacklist ch341, while kernels 6.0 and newer are observed to work more reliably.

CLI Configuration

Interaction with Meshtastic can be conducted via the command line:

```
```bash
meshtastic --host localhost ...
```
```