Flask MIDI Router Web GUI - Setup Guide for Debian-based Systems (including Raspberry Pi / ARM)

This guide walks you through setting up the Flask-based MIDI router and web GUI on a fresh Debian-based Linux system, including support for Raspberry Pi and ARM boards like Armbian.

✅System Requirements

- Python 3.7+
- `pip` (Python package manager)
- Linux system with ALSA (Advanced Linux Sound Architecture)

🧰Step 1: Install Required System Packages

Open a terminal and run:

```
sudo apt update
sudo apt install -y \
  python3 python3-pip python3-venv \
  libasound2-dev libjack-jackd2-dev \
  build-essential \
  libportmidi-dev \
  libffi-dev \
  libxcb1 libx11-xcb1 libxrender1 libxext6 libgl1-mesa-glx \
  alsa-utils
```

These dependencies cover MIDI handling, building Python packages, and running GUI/web tools if needed.

🧡Step 2: Install Python Packages

You can install these globally or inside a virtual environment:

```
pip3 install flask mido python-rtmidi
```

> Tip: Use `python3 -m pip` if `pip3` is not found.

✡️Optional: Set Up a Python Virtual Environment

```
python3 -m venv midi-env
source midi-env/bin/activate
pip install flask mido python-rtmidi
```

To deactivate later:

```
deactivate
```

🚯Step 3: Run the Server

In the folder containing your Flask MIDI router script:

```
python3 your_flask_script.py
```

If using a virtual environment:

```
source midi-env/bin/activate
python your_flask_script.py
```

Then open your browser and go to:

```
http://localhost:5050
```

Or use your device's IP address if accessing from another computer.

🔁Step 4: Set Up systemd Service (Autostart on Boot)

Create a service file:

```
sudo nano /etc/systemd/system/midi-router.service
```

Paste the following:

```
[Unit]
Description=MIDI Router Flask Server
After=network.target sound.target
```

```
[Service]
Type=simple
ExecStart=/usr/bin/python3 /path/to/your_flask_script.py
WorkingDirectory=/path/to/
Restart=always
User=pi

[Install]
WantedBy=multi-user.target
```

> Replace `/path/to/your_flask_script.py` with the **full path** to your Python script.
> Replace `User=pi` with the username you're running under (e.g. `armbian`, `ubuntu`, or your custom user).

Then enable and start the service:

```
sudo systemctl daemon-reexec
sudo systemctl daemon-reload
sudo systemctl enable midi-router.service
sudo systemctl start midi-router.service
```

Check status:

```
sudo systemctl status midi-router.service
```

🟦Raspberry Pi / ARM Notes

- All steps above work the same on Raspberry Pi OS or Armbian.
- `python-rtmidi` works well with both ALSA and JACK.
- Make sure to use `libasound2-dev` instead of older ALSA packages.

Optional: Enable real-time performance tweaks for better MIDI timing on Raspberry Pi:

```
sudo nano /boot/cmdline.txt
```

Add at the end (on the same line):

```
threadirqs
```

Then reboot:

```
sudo reboot
```

6 Optional Clean-Up

To stop and disable the service:

```
sudo systemctl stop midi-router.service
sudo systemctl disable midi-router.service
```

To remove:

```
sudo rm /etc/systemd/system/midi-router.service
sudo systemctl daemon-reload
```

♒Build as an AppImage

[... unchanged ...]

♒Build as a .deb Package

Step 1: Create Folder Structure

```
mkdir -p midi-router-deb/usr/local/bin
mkdir -p midi-router-deb/DEBIAN
```

Step 2: Copy Files

Place your script in the `bin` folder:

```
cp your_flask_script.py midi-router-deb/usr/local/bin/midi-router.py
chmod +x midi-router-deb/usr/local/bin/midi-router.py
```

Step 3: Create Control File

```
nano midi-router-deb/DEBIAN/control
```

Paste this:

```
Package: midi-router
Version: 1.0
Section: base
```

```
Priority: optional
Architecture: all
Depends: python3, python3-flask, python3-mido, python3-rtmidi
Maintainer: Percaine
Description: Flask-based USB MIDI Router GUI Server
```

Step 4: Build the Package

```
dpkg-deb --build midi-router-deb
```

This will produce `midi-router-deb.deb`.

Step 5: Install the .deb

```
sudo dpkg -i midi-router-deb.deb
```

If dependencies are missing:

```
sudo apt --fix-broken install
```

♒Ready to Go!

Your Flask-based MIDI router and GUI should now be fully operational on any Debian-based system, including Raspberry Pi or ARM boards. You can now connect and disconnect MIDI devices via your browser interface, portable AppImage, or installable .deb package.

Let me know if you'd like the packaging scripts or an actual prebuilt `.deb` archive.