

## Schritt 1: Raspberry Pi Konfiguration

- **Raspi-Config Änderungen:**
- Öffnen Sie das Raspberry Pi Konfigurationsmenü mit dem Befehl **sudo raspi-config**.
  - Aktivieren Sie die serielle Schnittstelle:
  - Navigieren Sie zu **Interfacing Options**.
  - Wählen Sie **Serial**.
  - Wählen Sie **<No>** für die Login-Shell, aber **<Yes>** für die serielle Schnittstelle.
- Beenden Sie das Konfigurationsmenü und starten Sie den Raspberry Pi neu.

## Schritt 2: Boot-Konfiguration

- **Änderungen in der boot/config.txt:**
- Öffnen Sie die Datei mit **sudo nano /boot/config.txt**.
- Stellen Sie sicher, dass die Einstellungen für die serielle Kommunikation korrekt sind. Es könnte notwendig sein, **enable\_uart=1** hinzuzufügen.
- Speichern und schließen Sie die Datei.

## Schritt 3: Installation notwendiger Pakete

- **Aktualisieren Sie Ihr System:**
  - Führen Sie **sudo apt update** und **sudo apt upgrade** aus, um sicherzustellen, dass Ihr System auf dem neuesten Stand ist.
- **Installieren von Python-Paketen über pip:**
  - Stellen Sie sicher, dass pip installiert ist mit **sudo apt install python3-pip**.
  - Installieren Sie die benötigte pySerial-Bibliothek mit **pip3 install pyserial**.

## Schritt 4: Überprüfen der Installation

- **Überprüfen der seriellen Schnittstelle:**
  - Führen Sie **ls -l /dev/serial0** aus, um sicherzustellen, dass die serielle Schnittstelle verfügbar ist.
- **Python-Version überprüfen:**
  - Überprüfen Sie die Python-Version mit **python3 --version**.

## Schritt 5: Bereitstellung des Skripts

- **Skript-Transfer:**
- Kopieren Sie das Skript **lora\_service\_v5.py** auf den neuen Raspberry Pi.
- Stellen Sie sicher, dass das Skript Ausführungsrechte hat (nutzen Sie **chmod +x lora\_service\_v5.py**).

## Schritt 6: Testlauf

- **Führen Sie das Skript aus:**
- Starten Sie das Skript mit **python3 lora\_service\_v5.py**.
- Überprüfen Sie die Funktionalität und achten Sie auf Fehlermeldungen im Terminal oder in der Log-Datei.