

# Manual for Setting Up and Running TravelBot on Raspberry Pi 5

## Prerequisites

- A Raspberry Pi 5 with Raspberry Pi OS installed
  - Internet connection
- 

## 1. Update and Upgrade Raspberry Pi

Before starting, ensure Raspberry Pi is up-to-date:

```
sudo apt-get update  
sudo apt-get upgrade
```

---

## 2. Install Python and Required Tools

### 2.1 Install Python 3 (if not preinstalled)

Check if Python 3 is installed:

```
python3 --version
```

If it is not installed, run:

```
sudo apt-get install python3
```

## 2.2 Install **pip** for Python package management

```
sudo apt-get install python3-pip
```

---

## 3. Set Up Virtual Environment (Optional but Recommended)

A virtual environment isolates project dependencies:

```
sudo apt-get install python3-venv  
python3 -m venv myenv  
source myenv/bin/activate
```

Every time you work on this project, you'll need to activate the virtual environment:

```
source myenv/bin/activate
```

---

## 4. Install Python Dependencies

Install the required Python libraries using **pip**:

```
pip install sounddevice requests pyttsx3 vosk
```

### Notes on Dependencies:

- **sounddevice**: Required for capturing microphone input.
- **requests**: For HTTP requests.
- **pyttsx3**: For Text-to-Speech.

- **vosk**: For offline speech recognition.

### **Additional system dependencies:**

```
sudo apt-get install portaudio19-dev espeak libespeak-ng1
```

---

## **5. Install the Ollama Application**

Ollama is the platform you'll use to run local LLM models like **tinydolphin**.

### **5.1 Download and Install Ollama**

To install Ollama, follow these steps:

```
curl -o ollama.deb  
https://ollama.com/download/Ollama_RaspberryPi5_arm64.deb  
sudo dpkg -i ollama.deb  
sudo apt --fix-broken install # Resolves any missing dependencies
```

Verify that Ollama is installed and running:

```
ollama
```

### **5.2 Download the Tinydolphin LLM Model**

Once Ollama is installed, download the model:

```
ollama pull tinydolphin
```

Ensure that Ollama is running:

```
ollama serve &
```

This will start the Ollama API locally on port 11434, which the code will use to communicate with the model.

---

## 6. Download Vosk Speech Recognition Model

Vosk is an offline speech recognition engine that requires a language model.

### 6.1 Download Vosk English Model

Download and extract the English model for Vosk:

```
mkdir -p ~/vosk_models
cd ~/vosk_models
wget
https://alphacephei.com/vosk/models/vosk-model-small-en-us-0.15.zip
unzip vosk-model-small-en-us-0.15.zip
```

This will create a folder with the Vosk English model files.

---

## 7. Test Audio Input and Output

To ensure that microphone and speakers are working correctly, you can list the available audio devices:

```
python3 -m sounddevice
```

Make sure to note the device ID for the microphone you'll be using, as you'll need it for running the code.

---

## 8. Configure and Run the Code

Now that all the dependencies and models are set up, you can run the code.

### 8.1 Copy the Code to Raspberry Pi

Ensure Python script (`your_script.py`) is copied to the Raspberry Pi.

### 8.2 Run the Code with the Correct Arguments

You'll need to specify the correct device ID for microphone and the path to the Vosk model. Here's an example:

```
python3 your_script.py --device 0 --model  
~/vosk_models/vosk-model-small-en-us-0.15
```

- **--device**: The input device ID (check the output of `python3 -m sounddevice`).
- **--model**: Path to the Vosk model directory.

---

## 9. Set Up OpenAI API (Optional)

If you are using OpenAI's GPT models, ensure you have an API key. Replace the hardcoded API key in code with own:

```
python
```

```
api_key = "sk-Your-OpenAI-API-Key"
```

You can securely store and retrieve this key using environment variables:

```
export OPENAI_API_KEY="sk-Your-OpenAI-API-Key"
```

In the code, retrieve the key like this:

```
python
```

```
import os
api_key = os.getenv("OPENAI_API_KEY")
```

---

## 10. Testing and Troubleshooting

Once everything is set up, you can test the full functionality:

- Ensure Ollama is running by checking the API on `localhost:11434`.
  - Use the Vosk model for speech recognition.
  - Test the wake word, commands, and responses from the local LLM and OpenAI APIs.
- 

## 11. Running the Script on Startup (Optional)

If you want this script to run on startup (e.g., for a voice assistant), you can set it up as a systemd service.

### 11.1 Create a Service File

```
sudo nano /etc/systemd/system/voice_assistant.service
```

Add the following content to the file:

```
[Unit]
Description=Voice Assistant

[Service]
ExecStart=/path/to/myenv/bin/python3 /path/to/your_script.py --device
0 --model /path/to/vosk_models/vosk-model-small-en-us-0.15
Restart=always
User=pi
```

```
[Install]
```

```
WantedBy=multi-user.target
```

## 11.2 Enable and Start the Service

```
sudo systemctl enable voice_assistant.service
```

```
sudo systemctl start voice_assistant.service
```

Now script will run automatically when the Raspberry Pi boots.