

Rust for embedded devices



Build a complete AI agent app for your device

EchOKit

Star, clone and fork 

EchoKit devices: https://github.com/second-state/echokit_box

EchoKit server: https://github.com/second-state/echokit_server

Introduction:

<https://opencamp.cn/Rust/camp/S02>

Sign up here:

<https://opencamp.cn/Rust/camp/S02/register?code=cHsXplq2vGdaM>

Learning Rust Camp S2

「 Rust Embedded 」

联合主办：Rust 基金会、SecondState、RustCC 社区、

清华大学开源操作系统训练营

学习时间：8月16日 至 9月6日

基础阶段 (8.17 ~ 8.23) 1周

- 介绍 Rust 的 firmware flash tool
- 介绍 Echokit 的使用与架构
- 介绍怎么用 Rust 连接 ESP32 的 BT

专业阶段 (8.24 ~ 8.30) 1周

- 使用 Rust 操作 ESP32 的麦克风与喇叭
- 使用 Rust 操作 ESP32 的显示屏
- 使用 Rust 实现 Web Socket 通讯

项目阶段 (8.31 ~ 9.5) 1周

- 介绍 Echokit 的 Rust-based AI server
- 在自己的机器上起开源的 AI 模型
- 在 AI server 上 MCP 服务



扫码报名



训练营小助手

The EchoKit device

An ESP32-S3 SoC + audio processor + microphone + speaker + buttons + USB

<https://opencamp.ai/Rust/bbs/2>

Echokit

08/04 16:37:59



嵌入式Rust训练营专用设备

EchoKit

🎓【训练营简介】嵌入式 Rust 训练营是一门面向初学者的项目制学习课程，涵盖嵌入式...

¥168

长按识别小程序 跟团购买



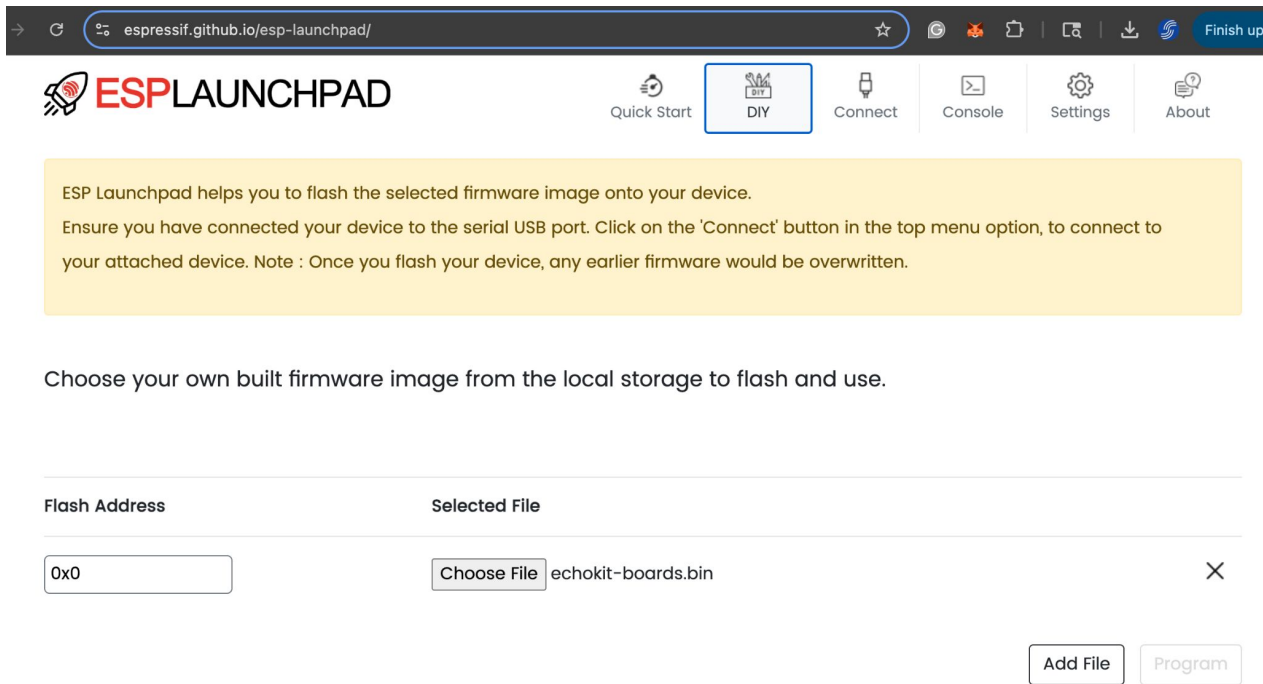
<https://echokit.dev/docs/hardware/assemble-echokit>

Assemble the device

Buttons

- RST - The “reset” button on the main board
- K0 - The “action” button on the top left of the extension board
 - It is the SAME as the “boot” button on the main board
- The buttons on the top right of the extension board? Make them your own!

Flash the firmware to the device



The screenshot shows the ESP Launchpad web interface in a browser. The address bar shows 'espressif.github.io/esp-launchpad/'. The navigation bar includes 'Quick Start', 'DIY' (highlighted with a blue box), 'Connect', 'Console', 'Settings', and 'About'. A yellow informational box contains instructions on how to flash firmware. Below this, a section titled 'Choose your own built firmware image from the local storage to flash and use.' contains a table with two columns: 'Flash Address' and 'Selected File'. The 'Flash Address' column has a text input with '0x0'. The 'Selected File' column has a 'Choose File' button and the text 'echokit-boards.bin'. To the right of the table is a close button 'X'. At the bottom right, there are 'Add File' and 'Program' buttons.

espressif.github.io/esp-launchpad/

ESPLAUNCHPAD

Quick Start **DIY** Connect Console Settings About

ESP Launchpad helps you to flash the selected firmware image onto your device.

Ensure you have connected your device to the serial USB port. Click on the 'Connect' button in the top menu option, to connect to your attached device. Note : Once you flash your device, any earlier firmware would be overwritten.

Choose your own built firmware image from the local storage to flash and use.

Flash Address	Selected File
<input type="text" value="0x0"/>	<input type="button" value="Choose File"/> echokit-boards.bin <input type="button" value="X"/>

<https://espressif.github.io/esp-launchpad/>

The Rust way

Install dependencies

See: <https://docs.espressif.com/projects/rust/book/installation/std-requirements.html>

Linux:

```
sudo apt-get install git wget flex bison gperf python3 python3-pip  
python3-venv cmake ninja-build ccache libffi-dev libssl-dev dfu-util  
libusb-1.0-0
```

Install Rust and the Cargo toolchain

```
curl --proto '=https' --tlsv1.2 -sSf https://sh.rustup.rs | sh
```

See: <https://www.rust-lang.org/tools/install>

Install Rust toolchain

```
cargo install espup --locked
```

```
espup install
```

```
. $HOME/export-esp.sh
```

It installs

- Espressif Rust fork with support for Espressif targets
- nightly toolchain with support for RISC-V targets
- LLVM fork with support for Xtensa targets
- GCC toolchain that links the final binary

Install flash tools

```
cargo install --locked cargo-espflash espflash ldproxy  
cargo-generate
```

Flash the firmware

- Connect to the device's TTL (or OTG) USB port
- Allow connection on your computer
- Build or download and then flash the firmware

```
git clone https://github.com/second-state/echokit_box
```

```
cargo build --release
```

```
esflash flash --monitor --flash-size 16mb echokit
```

<https://echokit.dev/docs/hardware/flash-firmware>

**Configure the
device**

Use Bluetooth

- Go to: <https://echokit.dev/setup/>
- Connect and pair
- Enter WiFi credentials
- Enter server URL
- Upload a background image
- Restart the device
 - You should hear a greeting message and see the screen light up

Troubleshooting

- Flashing fails
 - Enter the “download” mode:
 - Press and hold RST
 - Press and release the K0 once
 - Release RST
- Restart the config process
 - Press and release RST to restart
 - Press and hold K0 during restart

Test the device

Start the server

Build the server

```
git clone https://github.com/second-state/echokit_server
```

```
cargo build --release
```

<https://echokit.dev/docs/server/echokit-server>

Configure the server

```
addr = "0.0.0.0:9090"
hello_wav = "hello.wav"

[tts]
platform = "StreamGSV"
url = "http://localhost:9094/v1/audio/stream_speech"
speaker = "cooper"

[asr]
url = "http://localhost:9092/v1/audio/transcriptions"
lang = "auto"
# vad_url = "http://localhost:8000/v1/audio/vad"

# if you want to open server_vad in realtime mode, you can uncomment the following line
# vad_realtime_url = "ws://localhost:8000/v1/audio/realtime_vad"

[llm]
llm_chat_url = "http://localhost:9091/v1/chat/completions"
api_key = "Bearer gaia-1234"
history = 5

[[llm.sys_prompts]]
role = "system"
content = ""
You are a helpful assistant. Please answer user questions as concise as possible while being helpful.

If the user is speaking English, you must respond in English.

如果用户说中文，你必须用中文回答。

Si l'utilisateur parle français, vous devez répondre en français.

""""
```

This assumes that you are using local LLMs and AI servers running LlamaEdge API servers.

You can also use OpenAI or other commercial APIs

Configure the device

- Press and release RST once to restart
- Press and hold K0 while restarting
- Enter your own server URL
 - E.g., ws://192.168.2.102:9090/ws
- Restart again!

Until next time!