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

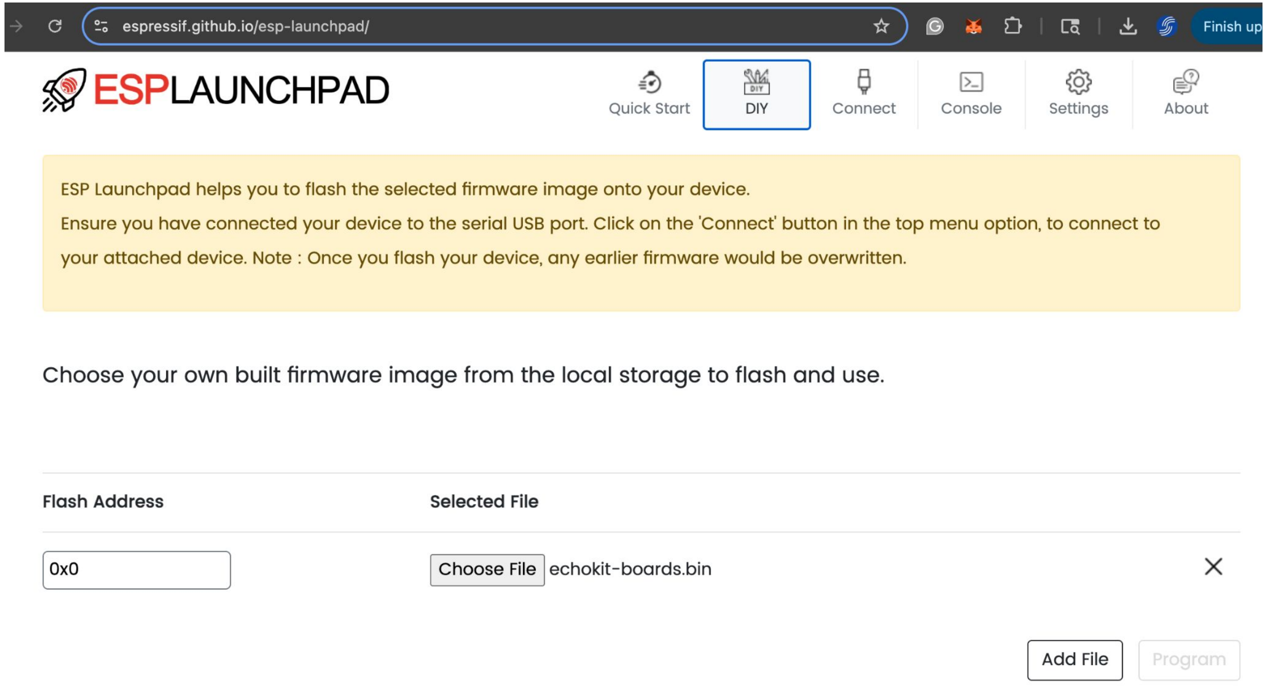
EchoKit server: https://github.com/second-state/echokit_server https://echokit.dev/docs/hardware/assemble-echokit

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



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
0x0	Choose File echokit-boards.bin

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


espflash 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

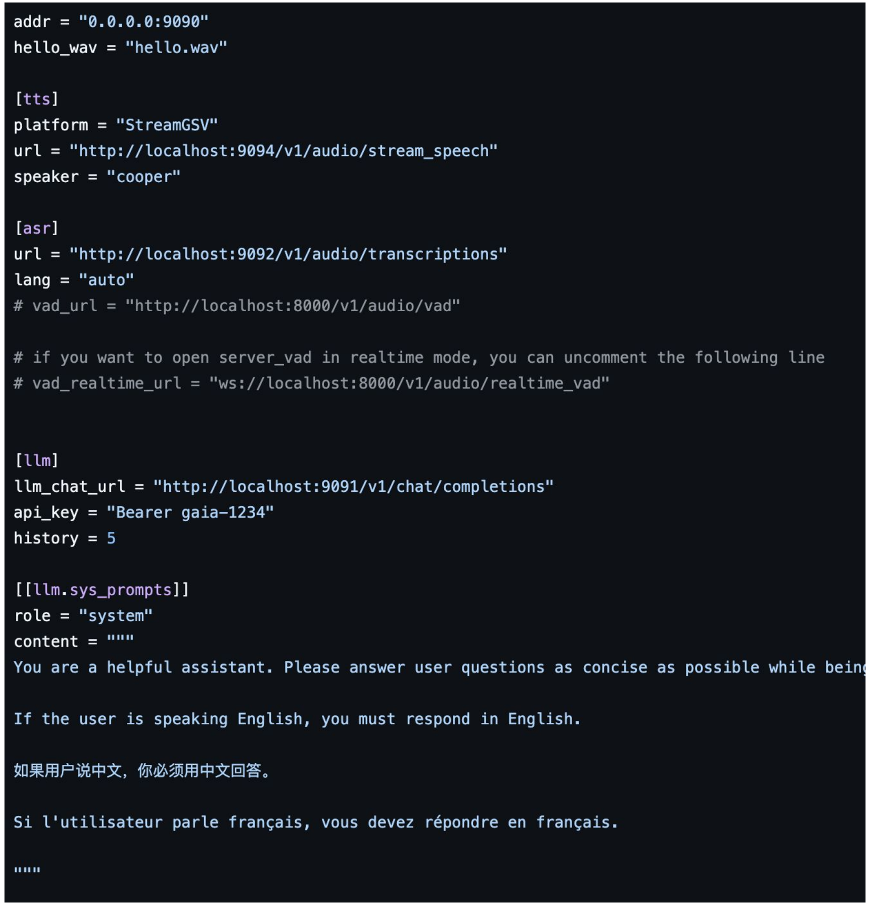
Build the server

```
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 bein

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!