Rust WebAssembly in Electron

Daniel Maslowski
Agenda

- Rust and WebAssembly
- Rust Wasm in Fiedka
- Rust Wasm in dtvis
Rust and WebAssembly
What if…

...we compile Rust to Wasm and use it in an app?

Magic happens — we can use native code on the web platform!
What if…

… we compile Rust…
What if…

... we compile Rust…

... to Wasm…

Magi loves code!
What if...

... we compile Rust...

... to Wasm...

... and use it in an app?
What if…

… we compile Rust… … to Wasm… … and use it in an app?

Magic happens - we can use native code on web platforms!
Howto

Getting started

https://lannonbr.com/blog/2020-01-07-rust-wasm-pack/
https://rustwasm.github.io/docs/wasm-pack/

TL;DR

cargo install wasm-pack
wasm-pack new my-rust-wasm-foo

The glue

https://github.com/wasm-tool/wasm-pack-plugin

More glue

cargo add gloo-utils
Howto

Getting started
https://lannonbr.com/blog/2020-01-07-rust-wasmpack/
https://rustwasm.github.io/docs/wasm-pack/
Howto

Getting started
https://lannonbr.com/blog/2020-01-07-rust-wasmpack/
https://rustwasm.github.io/docs/wasm-pack/

TL;DR
cargo install wasm-pack
wasm-pack new my-rust-wasm-foo
Howto

Getting started
https://lannonbr.com/blog/2020-01-07-rust-wasmpack/
https://rustwasm.github.io/docs/wasm-pack/

TL;DR
cargo install wasm-pack
wasm-pack new my-rust-wasm-foo

The glue
https://github.com/wasm-tool/wasm-pack-plugin
Howto

Getting started
https://lannonbr.com/blog/2020-01-07-rust-wasmpack/
https://rustwasm.github.io/docs/wasm-pack/

TL;DR
cargo install wasm-pack
wasm-pack new my-rust-wasm-foo

The glue
https://github.com/wasm-tool/wasm-pack-plugin

More glue
cargo add gloo-utils
The Rust side
The Rust side

```rust
extern crate wasm_bindgen;
use gloo_utils::format::JsValueSerdeExt;
use serde::{Serialize, Deserialize};
use wasm_bindgen::prelude::*;

/// ...
#[derive(Serialize, Deserialize)]
struct Foo {
    bar: u32,
    baz: String,
}

#[wasm_bindgen]
pub fn some_fun(data: JsValue) -> JsValue {
    /// ...
    let foo = Foo::new { bar: 42, baz: "Rust Wasm" };
    JsValue::from_serde(&foo).unwrap()
}
```
import { some_fun } from "./rs/pkg";

/* ... */
const res = some_fun({ woopWoop: 1337 });
console.info(res);
/* ... */
The JavaScript side

```javascript
import { some_fun } from "./rs/pkg";

/* ... */

const res = some_fun({ woopWoop: 1337 });
console.info(res);
/* ... */

But that is synchronous and blocking!
```
The JavaScript side

```javascript
import { some_fun } from "./rs/pkg";

/* ... */
    const res = some_fun({ woopWoop: 1337 });
    console.info(res);
/* ... */
```

But that is synchronous and blocking!


https://rustwasm.github.io/wasm-bindgen/api/wasm_bindgen_futures/
Rust Wasm in Fiedka
What is Fiedka again?

Fiedka is the graphical desktop firmware analyzer and editor.

https://fiedka.app

The current backend is written in Go and runs in WebAssembly. Rust support is being added because there are tools written in it, e.g., Romulan:

https://github.com/system76/romulan
What is Fiedka again?

Fiedka is the graphical desktop firmware analyzer and editor.

https://fiedka.app
What is Fiedka again?

Fiedka is the graphical desktop **firmware** analyzer and **editor**.

The current backend is written in Go and runs in Web Assembly.

https://fiedka.app
What is Fiedka again?

Fiedka is the graphical desktop firmware analyzer and editor.

The current backend is written in Go and runs in Web Assembly.

https://fiedka.app

Rust support is being added because there are tools written in it, e.g., Romulan: https://github.com/system76/romulan
DEMO: Rust Wasm in Fiedka
Rust Wasm in dtvis
What is dtvis?

1https://devicetree.org
What is dtvis?

*dtvis* is a new project - a DeviceTree\(^1\) visualizer.

\(^1\)https://devicetree.org
What is dtvis?

*dtvis* is a new project - a DeviceTree\(^1\) visualizer.

https://github.com/platform-system-interface/dtvis

\(^1\)https://devicetree.org
… so what is a Device Tree?

IEEE 1275\(^2\) Standard for Boot (Initialization Configuration)
Firmware: Core Requirements and Practices / Open Firmware\(^3\)

\(^2\)https://standards.ieee.org/ieee/1275/1932/
\(^3\)https://www.openfirmware.info/data/docs/of1275.pdf
… so what is a Device Tree?

IEEE 1275\(^2\) Standard for Boot (Initialization Configuration)
Firmware: Core Requirements and Practices / Open Firmware\(^3\)
describing hardware topology for non-discoverable devices

\(^2\)https://standards.ieee.org/ieee/1275/1932/
\(^3\)https://www.openfirmware.info/data/docs/of1275.pdf
... so what is a Device Tree?

- IEEE 1275\(^2\) Standard for Boot (Initialization Configuration) Firmware: Core Requirements and Practices / Open Firmware\(^3\)
- describing hardware topology for non-discoverable devices
- Linux, U-Boot, Zephyr, OLPC, FreeBSD and other projects use it ▶ including Apple back in the days

\(^2\)https://standards.ieee.org/ieee/1275/1932/
\(^3\)https://www.openfirmware.info/data/docs/of1275.pdf
… so what is a Device Tree?

IEEE 1275² Standard for Boot (Initialization Configuration) Firmware: Core Requirements and Practices / Open Firmware³

describing hardware topology for non-discoverable devices

Linux, U-Boot, Zephyr, OLPC, FreeBSD and other projects use it including Apple back in the days

not too elegant, attached to kernel via bindings

²https://standards.ieee.org/ieee/1275/1932/
³https://www.openfirmware.info/data/docs/of1275.pdf
... so what is a Device Tree?

IEEE 1275\(^2\) Standard for Boot (Initialization Configuration) Firmware: Core Requirements and Practices / Open Firmware\(^3\)

describing hardware topology for non-discoverable devices

Linux, U-Boot, Zephyr, OLPC, FreeBSD and other projects use it
    including Apple back in the days

not too elegant, attached to kernel via bindings

only few specified fields, most are “as someone wrote them”
    compare e.g. Amlogic vs Allwinner SoC based trees

\(^2\)https://standards.ieee.org/ieee/1275/1932/
\(^3\)https://www.openfirmware.info/data/docs/of1275.pdf
... so what is a Device Tree?

IEEE 1275\(^2\) Standard for Boot (Initialization Configuration) Firmware: Core Requirements and Practices / Open Firmware\(^3\)

describing hardware topology for non-discoverable devices

Linux, U-Boot, Zephyr, OLPC, FreeBSD and other projects use it

including Apple back in the days

not too elegant, attached to kernel via bindings

only few specified fields, most are “as someone wrote them”

compare e.g. Amlogic vs Allwinner SoC based trees

can range from a few hundred to a thousand nodes

\(^2\)https://standards.ieee.org/ieee/1275/1932/
\(^3\)https://www.openfirmware.info/data/docs/of1275.pdf
... so what is a Device Tree?

IEEE 1275\textsuperscript{2} Standard for Boot (Initialization Configuration) Firmware: Core Requirements and Practices / Open Firmware\textsuperscript{3}

describing hardware topology for non-discoverable devices

Linux, U-Boot, Zephyr, OLPC, FreeBSD and other projects use it
  ▶ including Apple back in the days

not too elegant, attached to kernel via bindings

only few specified fields, most are “as someone wrote them”
  ▶ compare e.g. Amlogic vs Allwinner SoC based trees

can range from a few hundred to a thousand nodes

the tree is a lie; there are cycles, e.g., power supplies and clocks

\textsuperscript{2}https://standards.ieee.org/ieee/1275/1932/
\textsuperscript{3}https://www.openfirmware.info/data/docs/of1275.pdf
Previous attempts

There were discussions on tooling at Linux Plumbers\(^4\), partially stalled.

\(^4\)https://elinux.org/images/8/83/Plumbers_2016_dt_device_tree_tools.pdf
Previous attempts

There were discussions on tooling at Linux Plumbers⁴, partially stalled.

Component Inspector (by Freescale, now NXP)
- proprietary, closed source Eclipse plugin
- was part of QorIQ Configuration Suite, no longer available

---

Previous attempts

There were discussions on tooling at Linux Plumbers\(^4\), partially stalled.

- **Component Inspector** (by Freescale, now NXP)
  - proprietary, closed source Eclipse plugin
  - was part of QorIQ Configuration Suite, no longer available

- **https://github.com/dev-0x7C6/fdt-viewer**
  - mixed tree + hex/text viewer, C++ + Qt
  - supports dtb, dtbo (overlay) and itb (FIT image)

---

\(^4\)https://elinux.org/images/8/83/Plumbers_2016_dt_device_tree_tools.pdf
\(^5\)https://www.spinics.net/lists/devicetree-spec/msg00950.html
Previous attempts

There were discussions on tooling at Linux Plumbers\textsuperscript{4}, partially stalled.

- **Component Inspector** (by Freescale, now NXP)
  - proprietary, closed source Eclipse plugin
  - was part of QorIQ Configuration Suite, no longer available

- [https://github.com/dev-0x7C6/fdt-viewer](https://github.com/dev-0x7C6/fdt-viewer)
  - mixed tree + hex/text viewer, C++ + Qt
  - supports dtb, dtbo (overlay) and itb (FIT image)

- [https://github.com/bmx666/dtv-demo](https://github.com/bmx666/dtv-demo)
  - “RFC - DTV (Device Tree Visualiser)” on mailing list\textsuperscript{5}
  - dt_s_ only, more of a text editor, Python + Qt\textsuperscript{6}


\textsuperscript{5}[https://www.spinics.net/lists/devicetree-spec/msg00950.html](https://www.spinics.net/lists/devicetree-spec/msg00950.html)

Previous attempts

There were discussions on tooling at Linux Plumbers\(^4\), partially stalled.

- **Component Inspector (by Freescale, now NXP)**
  - proprietary, closed source Eclipse plugin
  - was part of QorIQ Configuration Suite, no longer available

- **https://github.com/dev-0x7C6/fdt-viewer**
  - mixed tree + hex/text viewer, C++ + Qt
  - supports dtb, dtbo (overlay) and itb (FIT image)

- **https://github.com/bmx666/dtv-demo**
  - “RFC - DTV (Device Tree Visualiser)” on mailing list\(^5\)
  - dt_s_ only, more of a text editor, Python + Qt\(^6\)

- **VS Code plugin plorefice.devicetree\(^6\)**
  - syntax highlighting + collapsing
  - could be enhanced with *dtvis* :-)

\(^4\)https://elinux.org/images/8/83/Plumbers_2016_dt_device_tree_tools.pdf
\(^5\)https://www.spinics.net/lists/devicetree-spec/msg00950.html
\(^6\)https://marketplace.visualstudio.com/items?itemName=plorefice.devicetree
DEMO: Rust Wasm in dtvis
Related

Fiedka the Firmware Editor (OSFC 2021)

Platform System Interface - Design und Evaluation holistischer Computerarchitektur (rC3 2022)

Hack the Gadget! (MRMCD 2023)
https://talks.mrmcd.net/2023/talk/SLLVT8/
Thank you! :)}
Follow Me

https://github.com/orangecms
https://twitter.com/orangecms
https://mastodon.social/@cyrevolt
https://youtube.com/@cyrevolt
https://twitch.tv/cyrevolt

Daniel Maslowski


License: CC BY 4.0 https://creativecommons.org/licenses/by/4.0/