Z-Wave Alliance Summit 2024

Building a Z-Wave SDK

Peter Shorty, Trident IoT
Christian Salmony, Trident IoT
Philippe Coval, Silicon Labs

Agenda

- Open source SDK overview
- What was changed in Z-Wave Open Source
- How Trident IoT has build an SDK with OS Z-Wave
- How to build Open Source for Silicon labs chips

The Structure of Open-Source Z-Wave

Z-Wave open source was made as a self contained software project

This means that:

1. **The build system expects to be the root build system**
2. **Everything must exist in the Open Source directory structure**

The Structure of Open-Source Z-Wave

When making a vendor specific SDK using Open Source Z-Wave as source, Z-Wave Open should not be the root project.

This means that:

1. **Z-Wave open must be a sub module to another build system**

The Structure of Open-Source Z-Wave

```
Z-wave-stack \
              ───PAL
             │    ├───src
             │    └───inc
             ├───Apps
             ├───platform
             │    └───x86
             ├───ThirdParty
             │    ├───FreeRTOS-Kernel
             │    ├───mbedtls
             ├───ZAF
             ├───ZWave
             ├───SubTree
             │    └───libs2
```

The Structure of an SDK

```
SDK \
      ───Platform
      |      └───TridentIoT
      |           └───PAL
      ├───Apps
      ├───Drivers
      |   └───TC3202
      ├───Linkerscripts
      ├───z-wave-stack
              ├───Apps
              ├───PAL
              |    └─x86
              ├───...ZWave
              |
              .
```

What was changed in the Open-Source Project

- Given: CMake supports “project in project”
- Using CMake functions defined in z-wave-stack
- Moved all apps to Apps/
- Inherit CMake presets
- Invoke custom callback for each app

Upcoming changes in the Open-Source Project

- Add support for template file naming
- Add support for building for different platform variants
- Define versions in the parent project

- Probably more…

Live demo of Trident IoT Z-Wave SDK

Z-Wave Alliance Summit 2024 Deploying ZWA Z-Wave stack to Silicon Labs' hardware Philippe Coval Silicon Labs

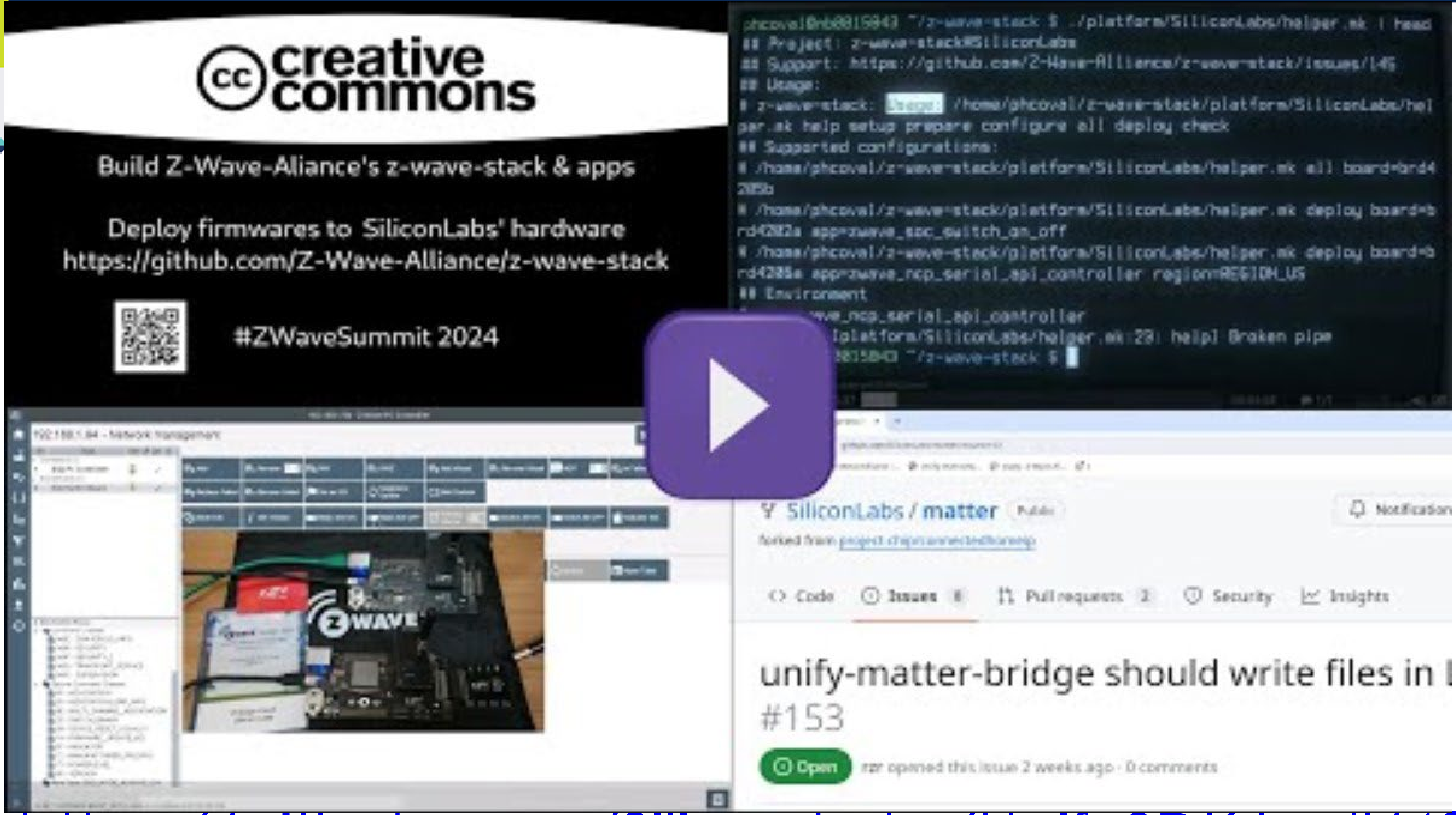
Z-Wave stack to product

- SiliconLabs provides Z-Wave as market ready product (HW+SW)
  - Through Simplicity Studio or GitHub GSDK
  - Sample apps as source linking to SiLabs' Z-Wave stack and PAL as binary
  - Industry ready, QA, consumer support etc
- Silicon Labs is part of ZWA cooperative development
  - Z-Wave stack reference platform, x86: it's a **simulated** network
    - No consumer support, we don't advise to go to production
  - For community, experimental, debugging purposes
  - Take into account that code base **differs**
    - Different upstreaming strategies or timelines, TTM...

Hardware support of opensource z-wave-stack

- Experimental support of OSS stack on SiLabs hardware
  - With a single minimalist makefile with a few rules
  - Try: **./platform/SiliconLabs/helper.mk help**
  - **setup** : It will install needed dependencies your system
    - Debian 12 supported, can be adapted to WSL and other Linux distros
  - **all**: build the ZWA stack and selected app (app=…) for board, region
    - app=zwave_ncp_serial_api_controller or app=zwave_soc_switch_on_off
  - **deploy**: flash the application to the selected hardware
    - ZGM230 is default board=brd4205b
  - Adapt if needed, note that some boards are too constrained
    - E.g.: ZG14 (board=brd4206a) can support only controller
- Feedback welcome, check:
  - https://github.com/Z-Wave-Alliance/z-wave-stack/wiki/Hardware

Demo: Build z-wave-stack for SiliconLabs' hardware...



creative commons

Build Z-Wave-Aliance's z-wave-stack & apps

Deploy firmwares to SiliconLabs' hardware
https://github.com/Z-Wave-Alliance/z-wave-stack

#ZWaveSummit 2024

```
phcoval@nb0815843 ~/z-wave-stack $ ./platform/SiliconLabs/helper.mk | head
## Project: z-wave-stack@SiliconLabs
## Support: https://github.com/Z-Wave-Alliance/z-wave-stack/issues/145
## Usage:
# z-wave-stack: Usage: /home/phcoval/z-wave-stack/platform/SiliconLabs/hel
per.mk help setup prepare configure all deploy check
## Supported configurations:
# /home/phcoval/z-wave-stack/platform/SiliconLabs/helper.mk all board=brd4
205b
# /home/phcoval/z-wave-stack/platform/SiliconLabs/helper.mk deploy board=b
rd4202a app=zwave_soc_switch_on_off
# /home/phcoval/z-wave-stack/platform/SiliconLabs/helper.mk deploy board=b
rd4205a app=zwave_ncp_serial_api_controller region=REGION_US
## Environment
        ve_ncp_serial_api_controller
        [platform/SiliconLabs/helper.mk:23: help] Broken pipe
        0815843 ~/z-wave-stack $
```

192.168.1.84 - Network management

SiliconLabs / matter Public

forked from project-chip/connectedhomeip

unify-matter-bridge should write files in
#153

Open rzr opened this issue 2 weeks ago · 0 comments

https://github.com/SiliconLabs/UnifySDK/pull/43

Cooperation, Competition, Convergence ?

- Challenges and limitations
  - Any PAL API changes will break hardware support
    - Until a new release of PAL (months at SiLabs)
  - Any PAL related changes should be also upstreamed on time
    - Hard to predict/control **integration** flow
- How to mitigate API breakages?
  - Work in branches
  - Become an **additional supported hardware** platform?
    - x86 will remain the reference model in z-wave-stack
    - CI enabled, should it block merges?
  - Hold breaking changes will slow down project
  - SiLabs to share API related changes to keep flowing
- Scalability challenge for more PAL vendors

Q&A

Thank you!

Z-WAVE®
ALLIANCE