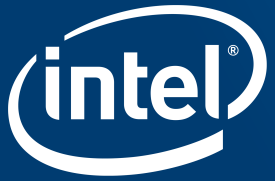


Zephyr® Project
Developer Summit 2022

Zephyr and You

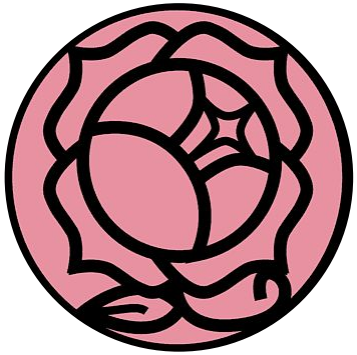
A Developer Environment for Newcomers

Presented by Lauren Murphy

Summary

- Setting up a Zephyr development environment
  - Linux (Ubuntu 20)
  - Visual Studio Code
- Streamlining Zephyr development workflow
- Assumes no knowledge of Zephyr
- Targeted to contributors, but useful for users

THIS IS NOT A TUTORIAL



Lauren Murphy
laurenmurphyx64

Intel contributor to the Zephyr Project.
CS BS & MS UTD '21. Go Comets!

Zephyr™ Project

What IDE do you use with Zephyr?





Resources

- Documentation
- Mailing list(s)
- Discord
  - #devenv
  - #user-experience (WG)
  - #ci
- GitHub issues / PRs
- Golioth's Awesome Zephyr RTOS

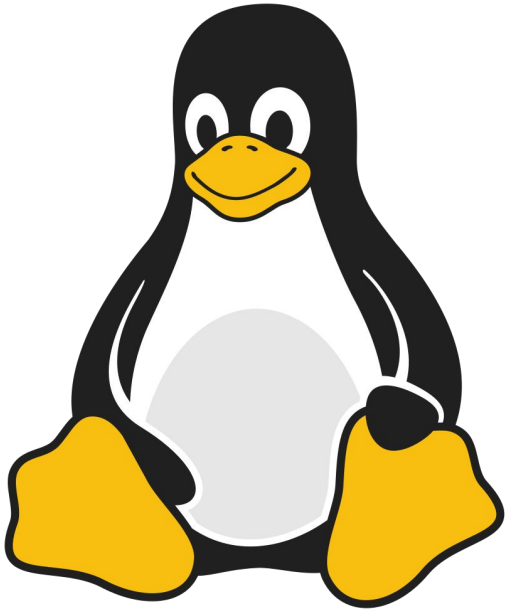
Profile Picture	Count
Real face (?)	57
Meme	27
Anime	18
Kpop (stan LOONA?)	1

Survey of 422 online users in Zephyr Discord

Setting Up Environment

Linux Environment

- Zephyr-specific environment variables
  - ZEPHYR_SDK_INSTALL_DIR
  - ZEPHYR_TOOLCHAIN (zephyr, etc.)
  - ZEPHYR_BASE
- Zephyr build-specific e.g., BOARD=qemu_xtensa
- Toolchain-specific
  - <TOOLCHAIN>_TOOLCHAIN_PATH (e.g., ESPRESSIF)
- Issues? Check env, e.g., env | grep ZEPHYR



Bash Scripting

- .bashrc
  - source zephyr-env.sh
    - Runs user-supplied ~/.zephyrrc
  - source west-completion.bash
- Aliases to export variables for toolchain env?
  - E.g., espenv, xccenv
  - clearenv to unset all?
- Other commands to alias?
  - cd $ZEPHYR_BASE
  - sudo minicom –D /dev/tty<DEVICE>

```
$ cat .bashrc
export ZEPHYR_TOOLCHAIN_VARIANT=zephyr
export ZEPHYR_SDK_INSTALL_DIR=
$HOME/zephyr-sdk-0.14.2
export ZEPHYR_BASE=$HOME/zephyrproject/zephyr
source $ZEPHYR_BASE/zephyr-env.sh
source $ZEPHYR_BASE/scripts/west_commands/completion/
west-completion.bash

##### ARM #####
export ARMGCC_DIR=$HOME/gnu_arm_embedded
##### END ARM #####

##### ESP #####
alias espenv='export
ZEPHYR_TOOLCHAIN_VARIANT="espressif" && \
export ESPRESSIF_TOOLCHAIN_PATH=
"${HOME}/.espressif/tools/zephyr" && \
cd $ZEPHYR_BASE/.. && west espressif update && cd
$ZEPHYR_BASE'
##### END ESP #####
...
```

```
$ west build -b qemu_
qemu_arc_em          qemu_cortex_m3
 qemu_riscv32_xip     qemu_x86_nommuti
...
```

Python Environment

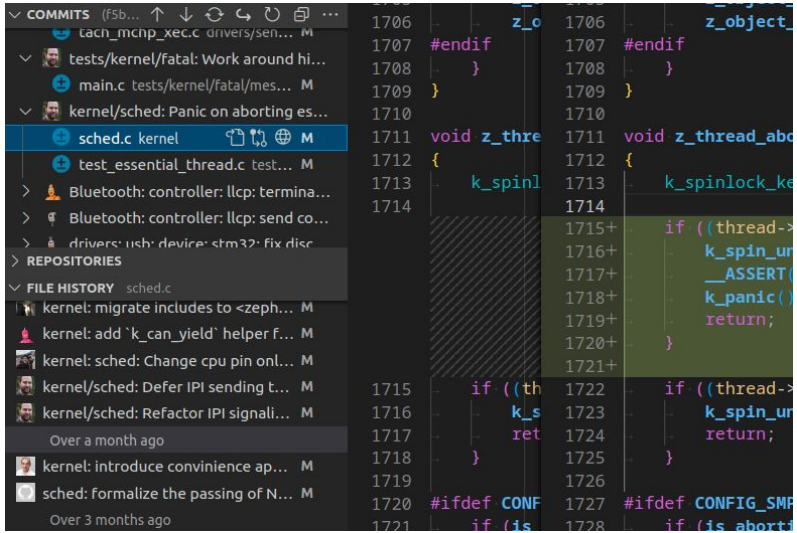
- Python 3.6 or higher
- Use a virtualenv in your $ZEPHYR_BASE*
- Install with --user*
- Do NOT install into Ubuntu's system Python if you are married and / or have children / pets

One of the many dangers of ~~dividing by zero~~
installing packages into your system Python on Ubuntu

DO AS I SAY, NOT AS I DO

Code: Extensions

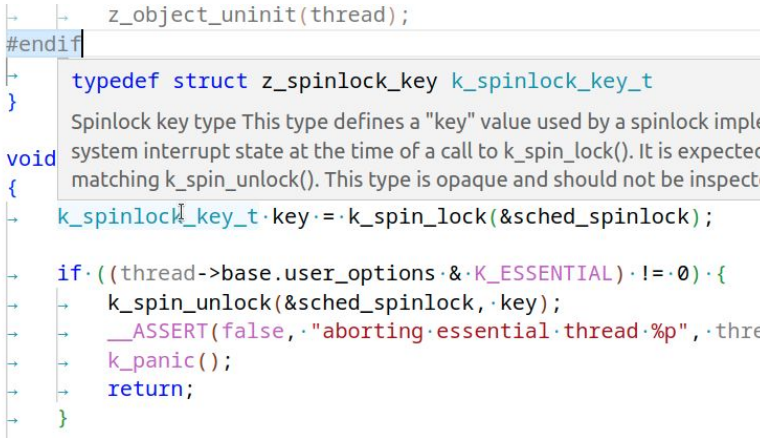
- (Not formally maintained or supported by Project)
- Zephyr-specific extensions
  - **DeviceTree for the Zephyr Project**
  - **Kconfig for the Zephyr Project**
- General extensions
  - **GitLens**
  - **C/C++**, CMake Tools
  - Python / Pylance
  - HexEditor
  - **One Dark Pro, vscode-icons**




```
1711  void z_thread_abort(struct k_thread *thread)
1712  {
1713      k_spinlock_key_t key = k_spin_lock(&sched_spinlock);
1714
1715      if ((thread->base.user_options & K_ESSENTIAL) != 0) {
1716          k_spin_unlock(&sched_spinlock, key);
1717          __ASSERT(false, "aborting essential thread %p", thread);
1718          k_panic();
1719          return;
1720      }
```

Visual Studio Code: C++ Intellisense

- Provided by Microsoft C/C++ extension
  - Need to make sure West is generating compile_commands.json
    - west config build.cmake-args -- -DCMAKE_EXPORT_COMPILE_COMMANDS=ON
  - Configure in .vscode/c_cpp_properties.json*
    - (Sample will be provided)
- Not 100% functional, but helpful
  - Most reliable for built code
  - For one specific compiler
- Ming (Intel) also has a [script] for Windows / Linux!



Development Workflow

West Workspace (zephyrproject/)

- (More than one way to skin a kite!)
- West manifest repo (overriding west.yml) version-controls Zephyr and modules
- Topologies
  - T1 Star (Default)
    - zephyr/ is manifest repo
  - T2 Star (User?)
    - Your app is manifest repo
  - T3 Forest (Support, e.g., Intel)
    - Standalone manifest repo

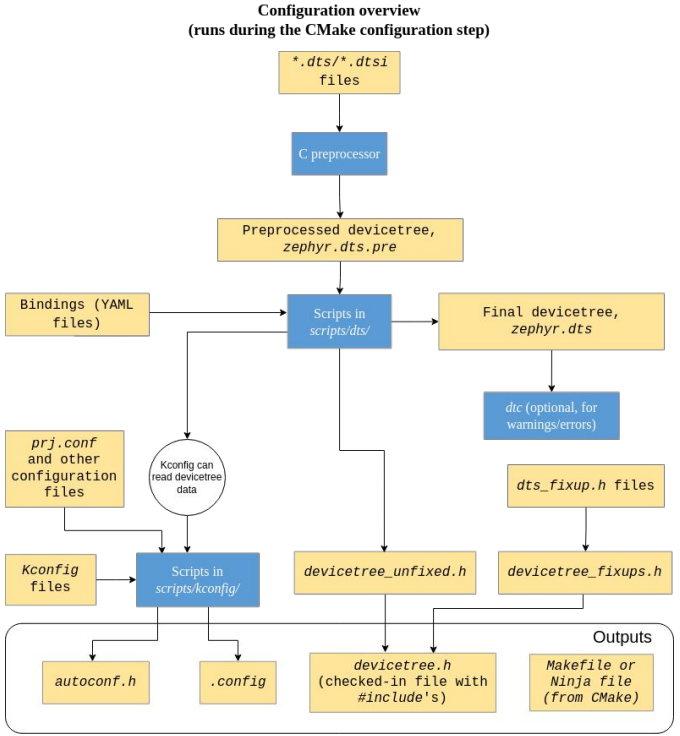
```yaml
manifest:
  defaults:
    remote: upstream

  remotes:
    - name: upstream
      url-base: https://github.com/zephyrproject-rtos

  #
  # Please add items below based on alphabetical order
  projects:
    - name: canopennode
      revision: 53d3415c14d60f8f4bfca54bfbc5d5a667d7e724
      path: modules/lib/canopennode
    - name: chre
      revision: 0edfe2c2ec656afb910cfab8ed59a5ffd59b87c8
      path: modules/lib/chre
    - name: civetweb
```

West Build

- Pristine: –p <OPTIONS>
  - Paranoid: rm –rf build/ && west build <OPTIONS>
- Verbose with log: –vvv <OPTIONS> > build.log 2>&1
- Setting macro for one build: –D<MACRO> e.g. -DCONF_FILE=<ALT>.conf
- Save intermediate files: -DEXTRA_CFLAGS="-save-temps"
- Recommended reading: West building and flashing

Configuration Phase: Important Files

- Kconfig, prj.conf -> .config
  - cd build && ninja menuconfig
  - ninja hardenconfig
- .config ~ zephyr/include/autoconf.h
- zephyr/zephyr.dts
  - zephyr/include/devicetree_unfixed.h
- build.ninja
- CMakeFiles/rules.ninja



Build Phase: Important Files

- west.yml
  - Do 'west update' after pulling latest Zephyr
- zephyr/zephyr.map
- zephyr/linker.cmd
- Recommended reading: Build system documentation

Regular Debugging

- Live with GDB
  - west debug (start GDB on board)
  - west debugserver (start GDB server listening on network port; QEMU / IDE)
- Tracing
- Post-mortem with coredump
- Sometimes, you just gotta printk...

INCOMING DEMO

Visual Debugging on Emulators

- (Sample launch.json will be provided)

```
{
    "name": "gdb xtensa",
    "type": "cppdbg",
    "request": "launch",
    "program": "${workspaceFolder}/build/zephyr/zephyr.elf",
    "miDebuggerServerAddress": "localhost:1234",
    "miDebuggerPath": "/home/herbert/zephyr-sdk-0.14.0/xtensa-sample_controller_zephyr-elf/bin/
    xtensa-sample_controller_zephyr-elf-gdb",
    "args": [],
    "stopAtEntry": true,
    "cwd": "${workspaceFolder}",
    "environment": [],
    "externalConsole": true,
    "MIMode": "gdb"
},
```

INCOMING DEMO

Limitations of Visual Debugging on Hardware

- VS Code doesn't support target extended-remote (yet), meaning that you can't restart the program after the board is flashed by west debugserver
- Workaround is to add an idle loop of 5-10 seconds in main()
  - Gives you time to start GDB, set breakpoint and connect to server
- Code also defaults to setting breakpoints as software
  - Force all breakpoints to be hardware in launch.json OR
  - Set breakpoints in console

Test

- Attend Aastha's Twister talk! (Today @ 1:40pm - 2:10pm)

Regression testing is crucial for system validation as Zephyr developers contribute to the project. With the use case in mind, this presentation talks about the various (common & uncommon but important) options and features that Twister Test Runner offers which developers can leverage to validate their changes and integrate TestCases. Additionally, this presentation explains the scope of twister runs and its integration in CI. This talk also touches on the differences between TestCases and samples in Zephyr and how twister itself is tested. Finally, it also outlines how a platform developer might setup the test environment for their own hardware.

Speakers

Aastha Grover

Software Developer, Intel Corporation

Aastha Grover is a Software developer with Intel working on Zephyr OS project since 2019. She graduated from Northeastern University, Boston with majors in Information Systems with her undergrad in Computer Sciences. She developed the TestSuite for Test Runner Twister and has been... Read More →

Run upstream CI checks locally

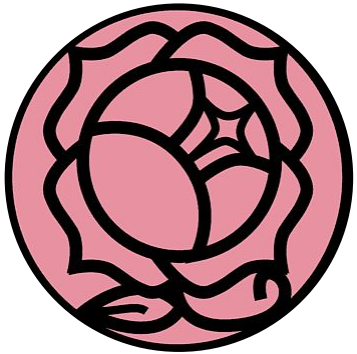
- Pre-check or validate GitHub CI failure
- Reverse-engineer .github/workflows/
- E.g., check compliance

  alias check_compliance='$ZEPHYR_BASE/scripts/ci/check_compliance.py -m Codeowners -m Devicetree -m Gitlint -m Identity -m Nits -m pylint -m checkpatch -m Kconfig'

  - Copies *some* of .github/workflows/compliance.yml
- Checking compliance and building docs most useful to run first locally
- Recommended reading: Contribution Guidelines

Questions?
Comments?
Suggestions?

(GitHub repo with PPT and .vscode/ here)



Lauren Murphy
laurenmurphyx64

Intel contributor to the Zephyr Project.
CS BS & MS UTD '21. Go Comets!