Zephyr in Action

Real-World Product Development - An Interactive Workshop

Jonas Remmert

September 8, 2023

Phytec Messtechnik GmbH - in Collaboration with PHYTEC Embedded Pvt. Ltd.

Table of contents

Getting Started

What is Zephyr?

An RTOS for IoT

- multiple supported architectures (ARM, RISC-V, x86...)
- Multi-threading
- Power Management

and much more!

- Open Source Bluetooth Low Energy Stack
- Networking, USB, Filesystems, Cryptography
- Shell, Logging, Sensors, Display, Audio

Ideal to build IoT products

- Well supported for a wide range of hardware
- Vendor neutral steering by Linux Foundation

How to get started?

Documentation

- Documentation: `https://docs.zephyrproject.org/latest/`
- Getting Started Guide: `https://docs.zephyrproject.org/latest/develop/getting_started/`
- Supported Boards: `https://docs.zephyrproject.org/latest/boards/`
- Samples and Demos: `https://docs.zephyrproject.org/latest/samples/`

Source

- Zephyr source (GitHub): `https://github.com/zephyrproject-rtos/zephyr`
- Out-of-tree application example: `https://github.com/zephyrproject-rtos/zephyr`

Setting up the Development Environment

Types of Requirements

- Linux, macOS or Windows 10/11
- Installation requirements (CMake, Python3..)
- Python requirements (west, pyocd..)
- Zephyr SDK that provides Toolchains (gcc, gdb, newlib..)

Everything is described in the Zephyr Getting Started Guide.

Samples, Applications and Products

Samples in Zephyr

Samples

- Zephyr provides a wide range of samples
- Samples are located in `zephyr/samples/`
- Isolated functionality or feature

Tests

- Tests are located in `zephyr/tests/`
- Isolated test cases for a feature or hardware
- Useful to test e.g. a device driver

Applications

- ZSWatch - Open Source Smart Watch: `https://github.com/jakkra/ZSWatch`

Examples

- Overview: `https://www.zephyrproject.org/products-running-zephyr/`
- Wildlife Tracking and Protection (OpenCollar)
- Wind Turbines (Vestas)
- Hearing Aid (Oticon)
- Wastewater Pump Monitoring (BeST Sensor, German Railways - DB)

Do not miss the talk tomorrow at 6 PM: ”Why BeST uses OpenSource and Zephyr RTOS”

Hands-on Examples

Examples for this Workshop

Five different examples that show a (small) subset of Zephyr features.

Examples

1. Hello World
2. Logging
3. Workqueues (and runtime context)
4. Shell
5. Sensor
6. Bluetooth Low Energy

Prerequisities to run the Samples

- Samples located in the Zephyr repository: jremmert-phytec-iot/zephyr-workshop
- Initialize the repository with west or clone into your existing workspace

Initialize without existing workspace

```
west init -m https://github.com/jremmert-phytec-iot/zephyr-workshop --mr main zephyrproject
# update Zephyr modules
cd zephyrproject
west update
```

Add Repository to existing workspace

```
cd zephyrproject
git clone https://github.com/jremmert-phytec-iot/zephyr-workshop
# Change west config manifest file location to the zephyr-workshop repository
west config manifest.path zephyr-workshop
west update
```

01_HELLO_WORLD

- Structure of a Zephyr Application
- Zephyr Repository: `zephyr/samples/hello_world`

Build

- west build -b qemu_cortex_m0 samples/01_hello_world -p

Run

- west build -t run

Expected terminal output

```
*** Booting Zephyr OS build zephyr-v3.4.0 ***
Hello World! qemu_cortex_m0
```

02_LOGGING

- Logging API
- Zephyr Repository: `zephyr/samples/subsys/logging`

Build

- west build -b qemu_cortex_m0 samples/02_logging

Run

- west build -t run

Expected terminal output

```
*** Booting Zephyr OS build zephyr-v3.4.0 ***
Hello World! qemu_cortex_m0
[00:00.001,570] <err> hello_world: error str
[00:00.001,593] <dbg> hello_world: main: debug str
[00:00.001,605] <inf> hello_world: info str
[..]
```

03_WORKQUEUES

- Work items getting called from a queue
- Offload work from interrupt context

Build

- west build -b qemu_cortex_m0 samples/03_workqueues

Run

- west build -t run

Expected terminal output

```
*** Booting Zephyr OS build zephyr-v3.4.0 ***
Work Item Executed - runtime context:
 Thread Name: main
 Thread Priority: 0

Work Item Executed - runtime context:
 Thread Name: sysworkq
 Thread Priority: -1

Work Item Executed - runtime context:
 Thread Name: my_work_q_thread
 Thread Priority: 5

Timer Expired!!
Work Item Executed - runtime context:
 ISR Context!

Work Item Executed - runtime context:
 Thread Name: sysworkq
 Thread Priority: -1
```

04_shell

- Interactive Shell with user-defined commands
- Zephyr Repository: `zephyr/samples/subsys/shell/shell_module`

Build

- west build -b qemu_cortex_m0 samples/04_shell -p

Run

- west build -t run
- -> press ”tab” to show commands

Expected terminal output

```
uart:~$
  bypass              clear
  date                demo
  device              devmem
  dynamic             help
  history             kernel
  log                 log_test
  nrf_clock_control   resize
  retval              section_cmd
  shell               shell_uart_release
  stats               version
```

List of threads running on the system (shortened)

```
uart:~$ kernel threads
Threads:
 0x200008a0 sysworkq
options: 0x0, priority: -1 timeout: 0
Total execution cycles: 137 (0 %)
stack size 1024, unused 856, usage 168 / 1024 (16 %)

*0x20000390 shell_uart
options: 0x0, priority: 14 timeout: 0
Total execution cycles: 144111 (0 %)
stack size 2048, unused 896, usage 1152 / 2048 (56 %)

 0x200006b0 idle
options: 0x1, priority: 15 timeout: 0
Total execution cycles: 588177458 (99 %)
stack size 256, unused 164, usage 92 / 256 (35 %)
```

05_sensor

- TI HDC1010: I2C Temperature and Humidity Sensor
- Zephyr repository: `zephyr/samples/sensor/ti_hdc/`

Build

- west build -b reel_board samples/05_sensor -p

Flash

- west flash

Show Terminal output

- reel board is connected via USB-Serial
- Check serial device (tty dev/com port)
- connect to board via terminal (minicom, tio,..)

Expected terminal output (shortened)

```
*** Booting Zephyr OS build zephyr-v3.4.0 ***
Running on arm!
Dev 0x8584 name ti_hdc@43 is ready!
Fetching...
Temp = 26.356506 C, RH = 59.747314 %
Fetching...
Temp = 26.406860 C, RH = 59.149169 %
```

06_ble

- BLE Peripheral device, temperature monitor
- Zephyr repository: `zephyr/samples/bluetooth/peripheral_ht`
- App to connect: nRF Connect for Mobile (Android, iOS)

Build

- west build -b reel_board samples/06_ble -p

Flash

- west flash

Show Terminal output

- reel board is connected via USB-Serial
- Check serial device (tty dev/com port)
- connect to board via terminal (minicom, tio,..)

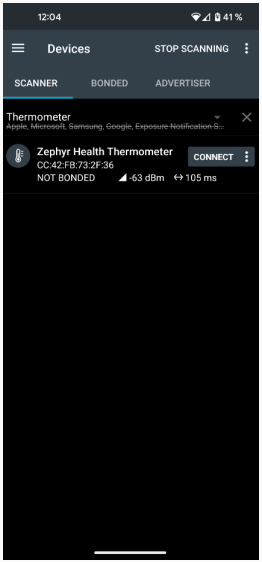
Expected terminal output (shortened)

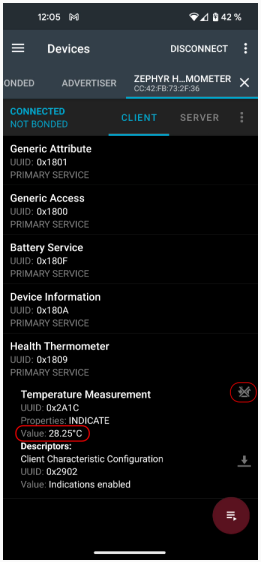
```
*** Booting Zephyr OS build zephyr-v3.4.0 ***
bt_hci_core: HW Platform: Nordic Semiconductor
bt_hci_core: HW Variant: nRF52x
bt_hci_core: Firmware: Standard BT controller 3.4
bt_hci_core: Identity: CC:42:FB:73:2F:36 (random)
bt_hci_core: HCI: version 5.4 rev 0x0000, mfg 0x05f1
bt_hci_core: LMP: version 5.4 subver 0xffff
Bluetooth initialized
temp device is 0x26500, name is temp@4000c000
Advertising successfully started
```

06_ble - Connect to device via 'nRF Connect for Mobile'

Terminal output

```
[..]
Connected
temperature is 28C
Indication success
Indication complete
```





Now.. let’s have some fun with Zephyr!