



Zephyr™ Project

Developer Summit

June 8-10, 2021 • @ZephyrIoT

On-target testing with Twister and the process of results publishing

MACIEJ PERKOWSKI,

NORDIC SEMICONDUCTOR: SOFTWARE TEST DEVELOPER

- Test terminology
- Hardware maps
- Quarantine
- Types of reports
- Result publishing process

Test terminology

- Test suite with test cases are defined in `[test]/src/main.c`

```
65 void test_main(void)
66 {
67     ztest_test_suite(test_eventfd_basic,
68                     ztest_unit_test(test_eventfd),
69                     ztest_unit_test(test_eventfd_write_then_read)
70                     );
71     ztest_run_test_suite(test_eventfd_basic);
72 }
```

```
18 static void test_eventfd(void)
19 {
20     int fd = eventfd(0, 0);
21
22     zassert_true(fd >= 0, "fd == %d", fd);
23
24     zsock_close(fd);
25 }
```

- `testcase/sample.yaml` defines test scenarios

```
1 common:
2   arch_exclude: posix
3   min_ram: 32
4   tags: posix pthread eventfd
5 tests:
6   posix.eventfd_basic:
7     extra_configs:
8       - CONFIG_POSIX_API=n
9   posix.eventfd_basic.posix_api:
10    extra_configs:
11      - CONFIG_POSIX_API=y
```

- test scenarios + platform = test configuration

Test terminology

```
00:48 $ ./scripts/twister -T tests/posix/eventfd_basic/ -v -p nrf52dk_nrf52832 -p nrf52840dk_nrf52840
ZEPHYR_BASE unset, using "/home/maciej/zephyrproject2/zephyr"
Renaming output directory to /home/maciej/zephyrproject2/zephyr/twister-out.16
INFO - Zephyr version: zephyr-v2.6.0-90-gefccfcbfc915
INFO - JOBS: 4
INFO - Using 'zephyr' toolchain.
INFO - Building initial testcase list...
INFO - 2 test scenarios (4 configurations) selected, 0 configurations discarded due to filters.
INFO - Adding tasks to the queue...
INFO - Added initial list of jobs to queue
INFO - 1/4 nrf52dk_nrf52832 tests/posix/eventfd_basic/posix.eventfd_basic PASSED (build)
INFO - 2/4 nrf52dk_nrf52832 tests/posix/eventfd_basic/posix.eventfd_basic.posix_api PASSED (build)
INFO - 3/4 nrf52840dk_nrf52840 tests/posix/eventfd_basic/posix.eventfd_basic PASSED (build)
INFO - 4/4 nrf52840dk_nrf52840 tests/posix/eventfd_basic/posix.eventfd_basic.posix_api PASSED (build)

INFO - 4 of 4 test configurations passed (100.00%), 0 failed, 0 skipped with 0 warnings in 10.37 seconds
INFO - In total 8 test cases were executed, 0 skipped on 2 out of total 369 platforms (0.54%)
INFO - 0 test configurations executed on platforms, 4 test configurations were only built.
INFO - Saving reports...
INFO - Writing xunit report /home/maciej/zephyrproject2/zephyr/twister-out/twister.xml...
INFO - Writing xunit report /home/maciej/zephyrproject2/zephyr/twister-out/twister_report.xml...
INFO - Run completed
```

Running Tests on Hardware

To enable testing on a hardware: add `--device-testing` to args

- Single device: `-p platform_name --device-serial port`
 - `twister -T samples/hello_world/ -p nrf52dk_nrf52832 --device-testing --device-serial /dev/ttyACM0`
- Multiple devices in parallel: Hardware map is needed.
 - To create one: `twister --generate-hardware-map map_name.yaml`
 - Fill *unknown(s)* with the proper names of connected platforms
 - Hardware map is also required if one wants to use platforms with fixtures
 - To use an existing one: `--hardware-map map_name.yaml`
 - It is important to regenerate the hardware map before each test run:
 - Updates connection statuses and matching ports

```
-| connected: true
   id: 000683205578
   platform: nrf52840dk_nrf52840
   product: J-Link
   runner: nrfjprog
   serial: /dev/ttyACM1
- connected: true
  fixtures:
  - gpio_loopback
    id: 000682923426
    platform: nrf52dk_nrf52832
    product: J-Link
    runner: nrfjprog
    serial: /dev/ttyACM0
```


- Failure/errors in one test can cause issues with other tests
 - E.g. a faulty test can leave a board in a corrupted state and cause consecutive tests to fail
- Automation-friendly solution (for on-target regression testing)
- Simple yaml files can be created and used to handle the quarantine
 - A sequence with combinations of scenarios/platform/comment mapping
- `--quarantine-list quarantine_list.yaml`
- Skipped tests will be marked in the report as “Quarantine: *user’s comment*”
- Can be verified by adding `--quarantine-verify`

```
- scenarios:
  - sample.basic.helloworld
platforms:
  - all
comment: "Link to the issue: https://github.com/zephyrproject-rtos/zephyr/pull/33287"

- scenarios:
  - kernel.common
  - kernel.common.misra
  - kernel.common.nano64
platforms:
  - qemu_cortex_m3
  - native_posix
```

Report types

Stored in 'zephyr/twister-out'. Modified with `-o / --report-dir path`

Extra args	Name	Format	Level of details
	twister.csv	CSV	scenarios
	twister.xml	JUnit XML	scenarios
	twister_report.xml	JUnit XML	test cases
--platform-reports	platform_name.xml	JUnit XML	test cases
--json-report	twister.json	JSON	test cases

- Names can be modified with `--report-suffix suffix`
- Platform reports are the ones which are submitted to the upstream
- ROM/RAM size available in CSV and JSON reports with `--show-footprint`

```

- <testsuites>
- <testsuite errors="0" failures="0" name="nrf52dk_nrf52832" skipped="10" tests="210"
time="37.976840">
- <properties>
  <property name="version" value="v2.6.0-rc2-110-gfe1de126fec9"/>
</properties>
<testcase classname="kernel.common" name="kernel.common.bootdelay" time="7.650268"/>
<testcase classname="kernel.common" name="kernel.common.irq_offload" time="7.650268"/>
<testcase classname="kernel.common" name="kernel.common.byteorder_memcpy_swap"
time="7.650268"/>
<testcase classname="kernel.common" name="kernel.common.byteorder_mem_swap"
time="7.650268"/>
<testcase classname="kernel.common" name="kernel.common.sys_get_be64" time="7.650268"/>
<testcase classname="kernel.common" name="kernel.common.sys_put_be64" time="7.650268"/>
<testcase classname="kernel.common" name="kernel.common.sys_get_be48" time="7.650268"/>
<testcase classname="kernel.common" name="kernel.common.sys_put_be48" time="7.650268"/>
<testcase classname="kernel.common" name="kernel.common.sys_get_be32" time="7.650268"/>
<testcase classname="kernel.common" name="kernel.common.sys_put_be32" time="7.650268"/>
<testcase classname="kernel.common" name="kernel.common.sys_get_be24" time="7.650268"/>
<testcase classname="kernel.common" name="kernel.common.sys_put_be24" time="7.650268"/>
<testcase classname="kernel.common" name="kernel.common.sys_get_be16" time="7.650268"/>
<testcase classname="kernel.common" name="kernel.common.sys_put_be16" time="7.650268"/>
<testcase classname="kernel.common" name="kernel.common.sys_get_le16" time="7.650268"/>
<testcase classname="kernel.common" name="kernel.common.sys_put_le16" time="7.650268"/>
<testcase classname="kernel.common" name="kernel.common.sys_get_le24" time="7.650268"/>
<testcase classname="kernel.common" name="kernel.common.sys_put_le24" time="7.650268"/>
<testcase classname="kernel.common" name="kernel.common.sys_get_le32" time="7.650268"/>
<testcase classname="kernel.common" name="kernel.common.sys_put_le32" time="7.650268"/>
<testcase classname="kernel.common" name="kernel.common.sys_get_le48" time="7.650268"/>
<testcase classname="kernel.common" name="kernel.common.sys_put_le48" time="7.650268"/>
<testcase classname="kernel.common" name="kernel.common.sys_get_le64" time="7.650268"/>
<testcase classname="kernel.common" name="kernel.common.sys_put_le64" time="7.650268"/>
<testcase classname="kernel.common" name="kernel.common.atomic" time="7.650268"/>
<testcase classname="kernel.common" name="kernel.common.threads_access_atomic"
time="7.650268"/>
- <testcase classname="kernel.common" name="kernel.common.bitfield" time="7.650268">
  <skipped message="Unknown" type="skipped"/>
</testcase>
<testcase classname="kernel.common" name="kernel.common.bitarray_declare" time="7.650268"/>
<testcase classname="kernel.common" name="kernel.common.bitarray_set_clear" time="7.650268"/>
<testcase classname="kernel.common" name="kernel.common.bitarray_alloc_free"
time="7.650268"/>
<testcase classname="kernel.common" name="kernel.common.bitarray_region_set_clear"
time="7.650268"/>
<testcase classname="kernel.common" name="kernel.common.printk" time="7.650268"/>
<testcase classname="kernel.common" name="kernel.common.timeout_order" time="7.650268"/>
<testcase classname="kernel.common" name="kernel.common.clock_uptime" time="7.650268"/>
<testcase classname="kernel.common" name="kernel.common.clock_cycle" time="7.650268"/>
<testcase classname="kernel.common" name="kernel.common.version" time="7.650268"/>

```

file:///home/maciej/zephyrproject2/zephyr/twister-out/twister.json

JSON Raw Data Headers

Save Copy Collapse All Expand All Filter JSON

```

{
  "environment": {
    "os": "posix",
    "zephyr_version": "v2.6.0-rc2-110-gfe1de126fec9",
    "toolchain": "zephyr"
  },
  "testsuites": {
    "0": {
      "testcases": {
        "0": {
          "testcase": "kernel.common.bootdelay",
          "arch": "arm",
          "platform": "nrf52dk_nrf52832",
          "ram_size": 30275,
          "rom_size": 118502,
          "status": "passed",
          "execution_time": 7.650268316268921
        },
        "1": {
          "testcase": "kernel.common.irq_offload",
          "arch": "arm",
          "platform": "nrf52dk_nrf52832",
          "ram_size": 30275,
          "rom_size": 118502,
          "status": "passed",
          "execution_time": 7.650268316268921
        },
        "2": {
          "testcase": "kernel.common.byteorder_memcpy_swap",
          "arch": "arm",
          "platform": "nrf52dk_nrf52832",
          "ram_size": 30275,
          "rom_size": 118502,
          "status": "passed",
          "execution_time": 7.650268316268921
        },
        "3": {
          "testcase": "kernel.common.byteorder_mem_swap",
          "arch": "arm",
          "platform": "nrf52dk_nrf52832",
          "ram_size": 30275,
          "rom_size": 118502,
          "status": "passed",
          "execution_time": 7.650268316268921
        },
        "4": {
          "testcase": "kernel.common.sys_get_be64",
          "arch": "arm",
          "platform": "nrf52dk_nrf52832",
          "ram_size": 30275,
          "rom_size": 118502,
          "status": "passed",
          "execution_time": 7.650268316268921
        }
      }
    }
  }
}

```


Publishing results

- Synchronization based on:
https://testing.zephyrproject.org/daily_tests/versions.json
- Checkout the zephyr's version to be tested
- Run twister with `--platform-reports` and using your hardware map
- Fork https://github.com/zephyrproject-rtos/test_results
- Copy/move *platform.xml* results into: ``test_results/results/version`` (create the directory if not existing)
- Create a github branch, add reports, commit and push
- Create a PR, add @nashif as a reviewer

```
1  import urllib.request
2  import json
3
4  ver_url = 'https://testing.zephyrproject.org/daily_tests/versions.json'
5
6  def get_version():
7      try:
8          webURL = urllib.request.urlopen(ver_url)
9          data = webURL.read()
10         ver_data = json.loads(data)
11         version = ver_data[-1]['version']
12     except:
13         version = 'master'
14     return version
15
16
17  if __name__ == "__main__":
18      print(get_version())
19
20
21
```

Syncing...

- The PR is verified by compliance script and some smoke tests
 - if zephyr's version the PR targets exists in the list of daily versions and if the version in the report is the same as the targeted one
 - file size \leq max size (5 MB),
 - platform name in the report == name of the report
 - #errors/#failures < threshold (50)
- Github action [process_results.yml](#) is triggered by each push to master and by cron once per day:
 - Converts Junit XML into HTML
 - Upload results to AWS S3
- Test results can be seen here: https://testing.zephyrproject.org/daily_tests/index.html
- The most results are published for weekly versions

- Terminology
 - Test **cases** and test **suites** are defined in .c files.
 - **Test scenarios** are defined in testcase/sample.yaml.
 - **Configuration = scenario + platform**. Each configuration has separate **binary**.
- Hardware maps
 - On-target setups require hardware maps.
 - HW map should be manually filled before the first run.
 - It is important to regenerate hw map before each run
- Quarantine
 - Useful for large test suites for on-target setups
 - DON'T ABUSE THIS FEATURE
- Reports:
 - Platform reports are the ones to be published to the upstream
- Publishing reports:
 - Tested version must come from the published list https://testing.zephyrproject.org/daily_tests/versions.json
 - To publish one has to raise a PR with the results to zephyrproject-rtos/test_results repo.
 - https://testing.zephyrproject.org/daily_tests/index.htm The most platforms are in weekly versions.
 - CONTRIBUTORS ARE MORE THAN WELCOM TO PUBLISH THEIR REPORTS. IMPORTANT FOR QA.



Zephyr™ Project
Developer Summit

Thank you



ZephyrTM Project

Developer Summit

June 8-10, 2021 ▪ @ZephyrIoT