



Zephyr® Project
Developer Summit 2022

HOWTO: Get Your Zephyr Patches Merged

Carles Cufí, Martí Bolívar
carles.cufi@nordicsemi.no, marti.bolivar@nordicsemi.no

Nordic Semiconductor

7 June 2022

Outline

- Prerequisites
- What to upstream
- Commits
- Branches
- Responding to feedback
- Getting attention
- Resolving disputes
- Summary





- ▶ Both: v3.1 release managers, Nordic Semiconductor engineers
- ▶ Carles: Bluetooth, documentation, shell, west
- ▶ Martí: Devicetree, west, documentation

Prerequisites

Know how to code

Trying to get your patches into upstream Zephyr is not a recommended way to learn embedded programming or git.



Required git and GitHub skills

- ▶ Basic setup (installing git, configuring `user.name` and `user.email`)
- ▶ Creating branches and commits
- ▶ Rebasing, including interactive rebasing and conflict resolution
- ▶ Forking, pushing, pulling, opening pull requests



Good written English

- ▶ Fluency is **not** required!
- ▶ ... but we need to understand what you're saying

What to upstream



Things to upstream

- ▶ New hardware support
- ▶ Kernel and subsystem enhancements
- ▶ Bug fixes
- ▶ Documentation
- ▶ Helpful sample code



Things to ask about

- ▶ Integrations with third party code
- ▶ Anything involving binary blobs

Commits

Anatomy of a good commit

- ▶ Does “one” thing
- ▶ Helpful commit message

Doing one thing

Remember: a good commit does one thing.

- ▶ **Do** make each commit accomplish one specific task
- ▶ **Do** use as many separate commits as you need
- ▶ **Don't** mix style tweaks with bug fixes, etc.



Doing two things

Showing 1 changed file with 141 additions and 141 deletions.

Split Unified

```
559 559 * @param prop lowercase-and-underscores property name
560 560 * @return a representation of the property's value
561 561 */
562 - #define DT_PROP(node_id, prop) DT_CAT(node_id, _P_##prop)
562 + #define DT_PROP(node_id, prop) DT_CAT3(node_id, _P_, prop)
563 563
564 564 /**
565 565 * @brief Get a property's logical length
566 566
567 567 @@@ -606,8 +606,8 @@@
568 568
569 569 * @return the property's length or the given default value
570 570 */
571 571 #define DT_PROP_LEN_OR(node_id, prop, default_value) \
572 572 - COND_CODE_1(DT_NODE_HAS_PROP(node_id, prop), \
573 573 - (DT_PROP_LEN(node_id, prop)), (default_value))
574 574 + COND_CODE_1(DT_NODE_HAS_PROP(node_id, prop), \
575 575 + (DT_PROP_LEN(node_id, prop)), (default_value))
576 576
577 577 /**
578 578 * @brief Is index "idx" valid for an array type property?
579 579
580 580 @@@ -630,7 +630,7 @@@
581 581
582 582 * into the given property, and 0 otherwise.
583 583 */
584 584 #define DT_PROP_HAS_IDX(node_id, prop, idx) \
585 585 - IS_ENABLED(DT_CAT6(node_id, _P_, prop, _IDX_, idx, _EXISTS))
586 586 + IS_ENABLED(DT_CAT6(node_id, _P_, prop, _IDX_, idx, _EXISTS))
587 587
588 588
```



Doing two things: substance

Showing 1 changed file with 141 additions and 141 deletions.

Split

Unified

```

559 559 * @param prop lowercase-and-underscores property name
560 560 * @return a representation of the property's value
561
562 - #define DT_PROP(node_id, prop) DT_CAT(node_id, _P_##prop)
562 + #define DT_PROP(node_id, prop) DT_CAT3(node_id, _P_, prop)
563
564 /**
565 * @brief Get a property's logical length
566
567 ****
567 +
567 -
567 ****
568 606 * @return the property's length or the given default value
569 607 */
570 608 #define DT_PROP_LEN_OR(node_id, prop, default_value) \
571 609 - COND_CODE_1(DT_NODE_HAS_PROP(node_id, prop), \
572 610 - (DT_PROP_LEN(node_id, prop)), (default_value))
573 609 + COND_CODE_1(DT_NODE_HAS_PROP(node_id, prop), \
574 610 + (DT_PROP_LEN(node_id, prop)), (default_value))
575 611
576 612 /**
577 613 * @brief Is index "idx" valid for an array type property?
578
579 +
579 -
579 +
580 630 * into the given property, and 0 otherwise.
581 631 */
582 632 #define DT_PROP_HAS_IDX(node_id, prop, idx) \
583 633 - IS_ENABLED(DT_CAT6(node_id, _P_, prop, _IDX_, idx, _EXISTS))
584 633 + IS_ENABLED(DT_CAT6(node_id, _P_, prop, _IDX_, idx, _EXISTS))
585 634

```

Doing two things: style

Showing 1 changed file with 141 additions and 141 deletions.

Split Unified

```
559 559 * @param prop lowercase-and-underscores property name
560 560 * @return a representation of the property's value
561
562 - #define DT_PROP(node_id, prop) DT_CAT(node_id, _P_##prop)
562 + #define DT_PROP(node_id, prop) DT_CAT3(node_id, _P_, prop)
563
564 564 /**
565 565 * @brief Get a property's logical length
566
566 @@ -606,8 +606,8 @@
567
568 606 * @return the property's length or the given default value
569
570 608 #define DT_PROP_LEN_OR(node_id, prop, default_value) \
571 609 - COND_CODE_1(DT_NODE_HAS_PROP(node_id, prop), \
572 610 - (DT_PROP_LEN(node_id, prop)), (default_value))
573 609 + COND_CODE_1(DT_NODE_HAS_PROP(node_id, prop), \
574 610 + (DT_PROP_LEN(node_id, prop)), (default_value))
575
576 611
577 612
578 613 * @brief Is index "idx" valid for an array type property?
579
580 @@ -630,7 +630,7 @@
581
582 630 * into the given property, and 0 otherwise.
583 631 */
584 632 #define DT_PROP_HAS_IDX(node_id, prop, idx) \
585 633 - IS_ENABLED(DT_CAT6(node_id, _P_, prop, _IDX_, idx, _EXISTS))
586 633 + IS_ENABLED(DT_CAT6(node_id, _P_, prop, _IDX_, idx, _EXISTS))
587
588 634
```



Doing one thing: substance

Showing 1 changed file with 1 addition and 1 deletion.

Split

Unified

```
include/zephyr/devicetree.h

@@ -559,7 +559,7 @@
559 559      * @param prop lowercase-and-underscores property name
560 560      * @return a representation of the property's value
561 561      */
562 - #define DT_PROP(node_id, prop) DT_CAT(node_id, _P_##prop)
562 + #define DT_PROP(node_id, prop) DT_CAT3(node_id, _P_, prop)
563 563
564 564      /**
565 565      * @brief Get a property's logical length
```

Doing one thing: style

Showing 1 changed file with 139 additions and 139 deletions.

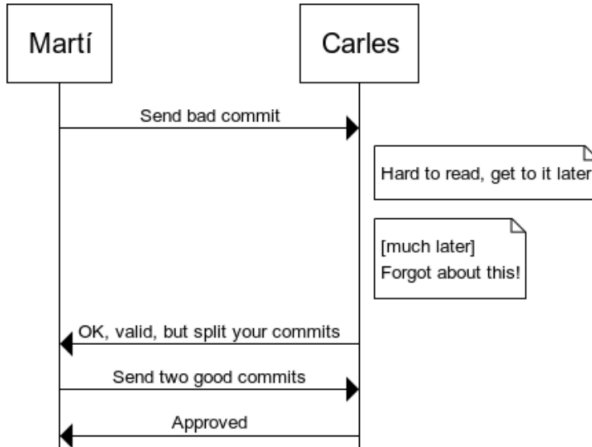
Split Unified

```
278 include/zephyr/devicetree.h

@@ -43,7 +43,7 @@
43 43 * _ENUM_TOKEN: property's value as a token into bindings enum (string
44 44 * enum values are identifiers) [deprecated, use _STRING_TOKEN]
45 45 * _ENUM_UPPER_TOKEN: like _ENUM_TOKEN, but uppercased [deprecated, use
46 - * _STRING_UPPER_TOKEN]
46 + * _STRING_UPPER_TOKEN]
47 47 * _EXISTS: property is defined
48 48 * _FOREACH_PROP_ELEM: helper for "iterating" over values in the property
49 49 * _FOREACH_PROP_ELEM_VARS: foreach functions with variable number of arguments

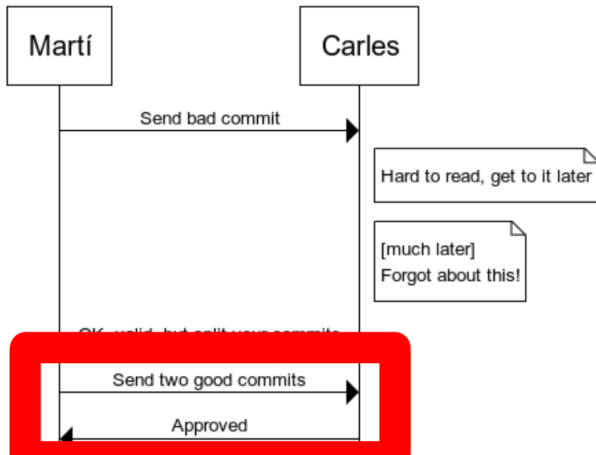
@@ -396,20 +396,20 @@
396 396 *
397 397 * Example devicetree fragment:
398 398 *
399 - * node-a {
400 - * compatible = "vnd,device";
401 - * status = "okay";
402 - * };
399 + * node-a {
400 + * compatible = "vnd,device";
401 + * status = "okay";
402 + * };
403 403 *
```

Doing two things: review process



Doing one thing: review process

Only the last round trip was necessary!



Commit messages

Each commit log *must* contain a one-line shortlog, explanation for why it is needed, and at least one Signed-off-by: line:

```
[subsystem]: [summary of change]
```

```
[Explanation for why the change is being made,  
how you tested it, any tricky parts, etc..]
```

```
Signed-off-by: [Your full name] <your.email@address>
```



Commit messages: example 1

Message for zephyr commit bfff5012.

```
drivers: bme280: factor out bus I/O routines
```

```
Move the SPI and I2C bus I/O bits into their own files. This makes  
this driver more similar to other sensor drivers.
```

```
Signed-off-by: Martí Bolívar <marti.bolivar@nordicsemi.no>
```

Remember: *why*, not *what*, and sign off.

- ▶ **Don't just** summarize *what* the commit does!
- ▶ **Do** explain *why* the change is worth doing
- ▶ **Do** include a Signed-off-by: line matching your `user.email`

Bad commit log

bluetooth fix

This fixes the bug with GATT

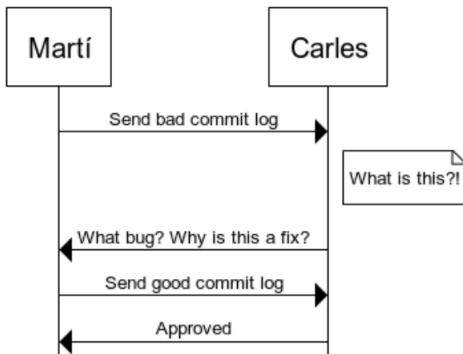
Bluetooth: Mesh: fix gatt network id advertising

Per prf spec 7.2.2.2.1, "A node that does not support the Proxy feature or has the Proxy feature disabled shall not advertise with Network ID".

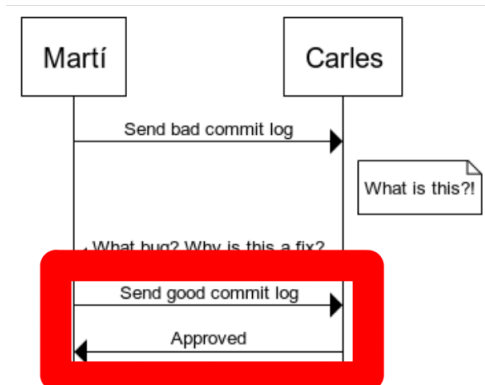
The PR aligns implementation with the specification.

Signed-off-by: Aleksandr Khromykh <Aleksandr.Khromykh@nordicsemi.no>

Bad commit log: review process



Good commit log: review process



Branches



Anatomy of a good branch

- ▶ Linear history
- ▶ No “fixup” commits
- ▶ Relevant changes only



Linear history

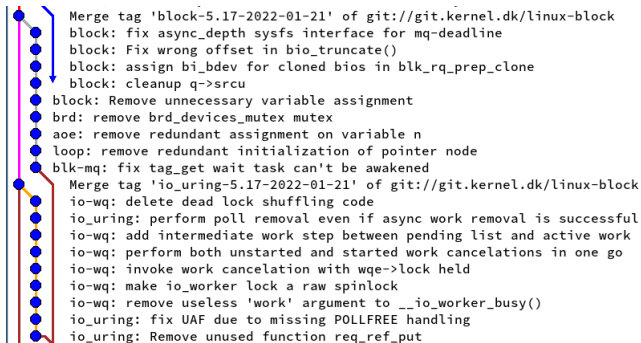
gitk on your branch should look like this:

```
Bluetooth: audio: Fix call control client build
soc: stm32wb: Move MB_MEM2 linker section to SRAM1
net: ipv4: Accept unspecified src address if dst is broadcast
lwm2m: software management: Fix URI value not being written
net: ethernet: Add Kconfig option to forward unrecognized EtherType frames
net: sockets: packet: Remove obsolete TODO
tests: net: sockets: Update packet socket tests
net: conn: Add proper processing of DGRAM packet sockets
net: pkt: Add flag indicating if pkt has already been processed by L2
net: context: Improve packet socket interface selection on sendto()
samples/soc_flash_nrf: Fix printf usage with off_t values
tests/c_lib: Set _GNU_SOURCE for qsort_r test
tests/mem_alloc: Eliminate alloc warnings. Define _BSD_SOURCE
tests/kernel/common: Include errno_private.h
tests/spi_loopback: Fix buffer sizes to avoid buffer overflow
subsys/img_mgmt: Add zephyr/toolchain.h for STRINGIFY [v2]
```



Non-linear history

gitk on your branch should **not** look like this:



Linear history: why

Remember: review bandwidth is scarce.

- ▶ **Do** use linear commits to tell a story
- ▶ **Do** tell the story clearly



Typical structure of a bigger PR

- ▶ Preparation: style, refactoring, etc.
- ▶ Main sequence: new features, main fixes
- ▶ Cleanup: removal or deprecation of old code



“Fixup” commits

drivers: fix build for my cool driver



mbolivar-nordic committed 27 seconds ago

drivers: add my cool driver



mbolivar-nordic committed 38 seconds ago



“Fixup” commits: don’t!

Remember: each commit accomplishes a task.

x drivers: fix build for my cool driver



mbolivar-nordic committed 27 seconds ago

drivers: add my cool driver



mbolivar-nordic committed 38 seconds ago

- ▶ **Don’t** submit fixup commits in the same PR
- ▶ **Do** squash or edit fixes into the right commits

This also applies to **style changes** and **responses to feedback**.



No “fixup” commits: why

- ▶ Readability and ease of review
- ▶ Fixup commits break `git bisect`



“Fixup” commits: exceptions

- ▶ “Draft” PRs
- ▶ PRs with [RFC] (request for comments) in the title



Responding to feedback

Await feedback

- ▶ Address review feedback you receive on your PR
- ▶ Reviewers may request changes
- ▶ Continuous integration (CI) may report problems



Keep your pull request clean

- ▶ **Do** squash changes as needed
- ▶ **Do** rebase your branch onto `main` as needed
- ▶ **Do** fix CI problems
- ▶ **Do** force push your changes to your branch on GitHub
- ▶ **Don't** merge `main` into your branch
- ▶ **Don't** close the pull request and open a new one every time



Getting attention



This can take time

Remember: review is optimized for *throughput*, not *latency*.

- ▶ Please be patient; we really do want to help you
- ▶ Some maintainers are volunteers
- ▶ Paid maintainers have their own code to write, too
- ▶ Review is hard, especially if we don't know you



Do:

- ▶ chat on Discord (invite: chat.zephyrproject.org)
- ▶ join a relevant weekly meeting
- ▶ email devel@lists.zephyrproject.org
- ▶ talk to us at conferences
- ▶ send RFC or draft pull requests
- ▶ talk to your company's project representatives (if any)



Getting attention

Don't:

- ▶ ping incessantly
- ▶ be rude

Resolving disputes



Dealing with maintainers

Remember: maintainers have the right to ask for changes, but maintainers are not always right.

- ▶ **Do** respect these rights
- ▶ **Do** let us know when we're wrong
- ▶ **Do** try to resolve disputes amicably
- ▶ **Do** hold us accountable for bad behavior (see [zephyr/CODE_OF_CONDUCT.md](#))



Where to discuss your work

- ▶ GitHub
- ▶ Discord
- ▶ Email lists
- ▶ At a meeting (e.g. dev-review or API)
- ▶ At conferences



TSC: the final decider

Remember: the TSC can vote to break roadblocks.

- ▶ **Do** keep this in mind
- ▶ **Don't** escalate to the TSC prematurely



Summary

- ▶ Thanks for submitting to Zephyr!
- ▶ We are drowning in GitHub notifications
- ▶ Following this process saves time overall



**THANK YOU FOR YOUR
PATCHES!**