Porting Zephyr to the CH32V003

michaelh@juju.nz   juju.nz/michaelh/   @nzmichaelh

CH32V003 highlights

RV32E with 16 KiB, 2 KiB

UART, I2C, SPI

8 channel ADC

18 GPIO

Timer, advanced timer

Systick, watchdog

SWD

5V tolerant, hand-solderable

What’s in Zephyr

RV32E with 16 KiB, 2 KiB

UART, *I2C*, *SPI*

8 channel ADC

18 GPIO

Timer, advanced timer

Systick, *watchdog*

SWD

5V tolerant, hand-solderable

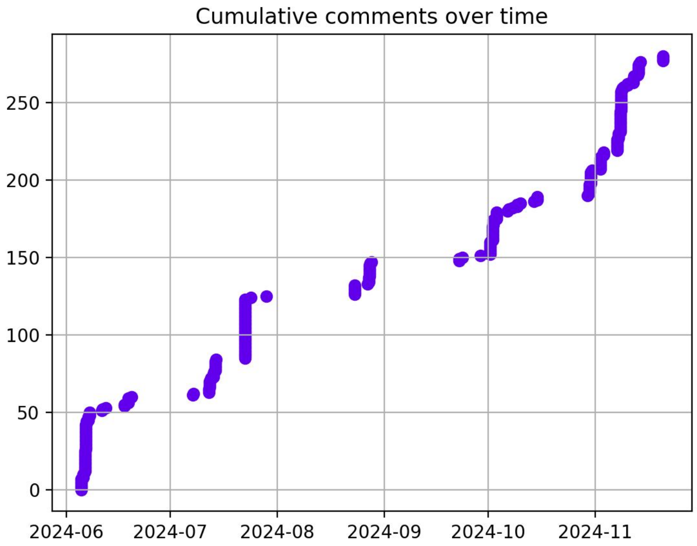
The review process

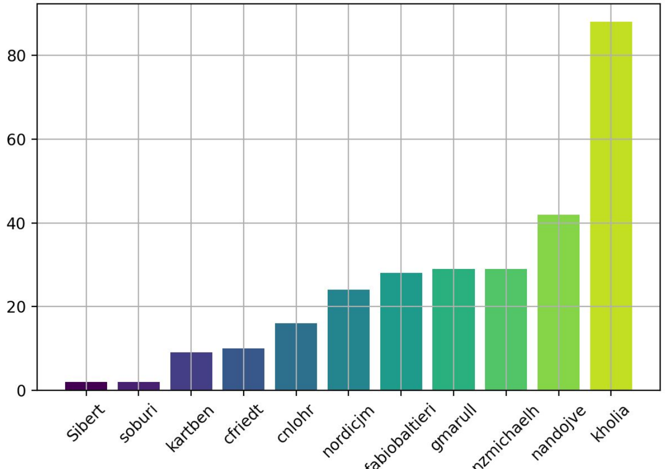
5 months so far, down to one outstanding

Where to put the HAL caused churn

Devicetree expectations could be clearer

Can we bring this down to two months?





Footprint

```
$ west build -p -b ch32v003evt samples/basic/blinky
```

```
[99/99] Linking C executable zephyr/zephyr.elf
Memory region         Used Size  Region Size  %age Used
             ROM:       10448 B        16 KB     63.77%
             RAM:        1524 B         2 KB     74.41%
```

Project ideas

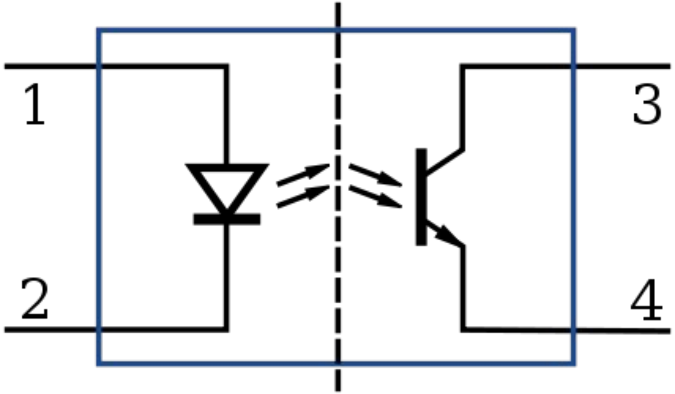
Isolated proportional solenoid driver

Position depends on current

Supplied by an unregulated battery on a different ground

Instead of sending the drive and feedback over the barrier…

send demand over a isolated UART



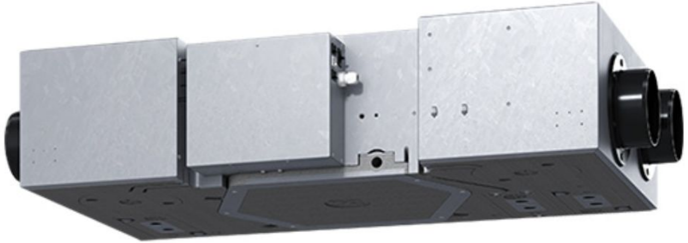
Telemetry display for PX4 autopilot

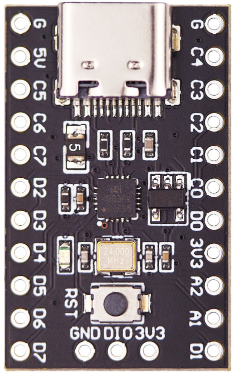
Mavlink decoder and display driver

4.5 km/h

Ventilator integration with Home Assistant

Passive I2C sniffer





What’s next

Land the PR!

Add the pending I2C, SPI, timer, ADC, and watchdog drivers

Add support for the CH32V002 (+4 KiB RAM) when released

Interested to chat about:

- Reducing Zephyr’s flash and RAM footprint, and keeping on top of regressions
- Improving the review experience