Adding support for Puya and WCH chips to Zephyr - 2025

Dhiru Kholia

About me

- Software security engineer by the day
- HW, amateur radio, RF enthusiast
- !!! Not a SME on today's topics !!!
- dhiru@openwall.com / https://github.com/kholia / https://notebook.rfcorner.in/ - my solar powered blog

Agenda

- Why Zephyr?
- Why CH32V003 is interesting?
- CH32V003 support in Zephyr
- Why PUYA PY32F is interesting?
- PY32F support in Zephyr

Why Zephyr? Why my interest in it?

- I heard about Zephyr from Amit, Anuj and Akshay at the Makerville ‘24 Conference in Pune
- Zephyr is the de-facto portable RTOS for resource-constrained devices
- The “Linux kernel” of the MCU world
- Write once, run on almost anything - lot of chips became unobtanium during Corona times!
- Strong, helpful open-source community. No-brainer choice these days!

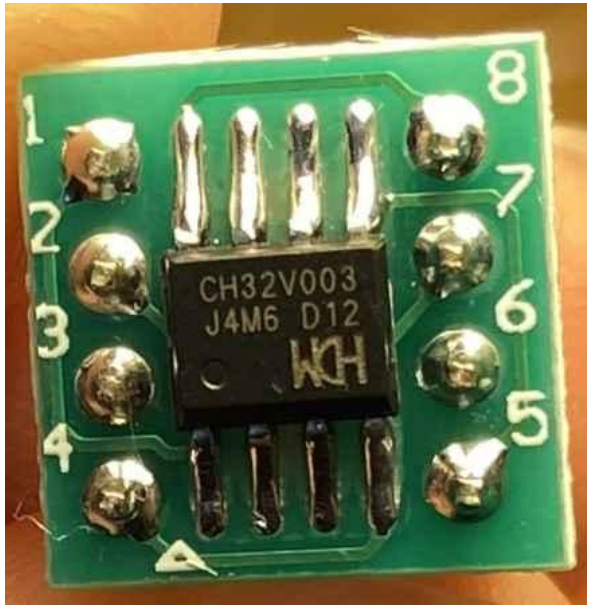
Why Zephyr? Why my interest in it?

- Already at this point, I wanted to experiment with Zephyr ecosystem!

- Around this time, I was playing around with WCH CH32V003 SOP-8 chips at home

- So the idea of adding support for CH32V003 to Zephyr was born!

CH32V003 - SOP-8 package



CH32V003 VS ATtiny85 - cost perspective

- ATtiny85 is a popular small MCU, often the traditional go to choice for small projects
- ATTINY85-20PU is ~2.5 USD (10 NOS). 199 INR on Robu.
- WCH CH32V003 is ~0.22 USD (5 NOS) from LCSC. 26 INR including GST in loose quantities in India (https://punoscho.in/).
- More than an order of magnitude difference in bulk LCSC.com pricing!

CH32V003 VS ATtiny85

- 20 MHz 8-bit AVR vs 48 MHz 32-bit RISC-V
- 4X more RAM, 2X more flash in CH32V003
- Better ADC, more peripherals in CH32V003
- 5V tolerant GPIOs, hand-solderable (003 operates with a wide voltage range 2.x to 5v)
- More well-behaving peripherals in CH32V003 (I2C bugs, anyone?)

CH32V003 Versus STM32G031J4M6

- STM32G031J4M6 - Costs 133 INR

- STM32G031 couldn’t run the EEPROM emulator program (essentially a I2C slave working at 100 to 400 kHz speed)! Spent too many days on it :-(

- With CH32V003 and ‘cnlohr/ch32fun’ software, I got the EEPROM emulation working in less than 30 minutes!

Zephyr Footprint

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

...

[114/114] Linking C executable zephyr/zephyr.elf

Memory region     Used Size  Region Size  %age Used

           ROM:      11684 B        16 KB    71.31%

           RAM:       1518 B         2 KB    74.12%
```

CH32V006 - the future is (almost) here now!

- More flash and RAM (4X)!
- 12-bit ADC
- 2 to 5V - extended voltage range
- Lot more goodies!
- https://www.cnx-software.com/2024/05/09/wch-ch32v006-risc-v-microcontroller-adds-more-i-os-memory-and-storage-compared-to-ch32v003/

WCH - other chips

- WCH has BLE MCUs too! Ridiculously low cost!
- WCH 5XX and others
- **YOUR** help is needed in adding support in Zephyr!

Why this Zephyr port was and is interesting?

- How to fit an RTOS on a small chip?
- How to make it work decently well?
- Neat, self-contained optimization problem
- A personal challenge to improve skills

Lot of things start with a Google search

- Initially I was quite lost about where to even start doing so (porting Zephyr to CH32V003).
- This is quite a common experience for engineers -> learning, searching and figuring out new things as needed ;)
- After a bit of searching, I learned about Michael Hope's (https://juju.nz/michaelh/) existing work.
- We got in touch and the joint upstreaming efforts started!

Technical challenges encountered - 1

- k_msleep(500) was NOT returning! Fixed by correctly setting up ‘mtvec’ (vector table related) and PFIC - via Michael

- Interim guerilla debugging -> Removed k_msleep(500). Added breakpoint on "gpio_ch32v00x_port_toggle_bits" function in GDB. This allowed us to test many things even without basic Zephyr code working! - Dhiru’s hack

- UART took some debugging to work (pinmux setup related problem - fixed by Michael too)

Process challenges we encountered - 1

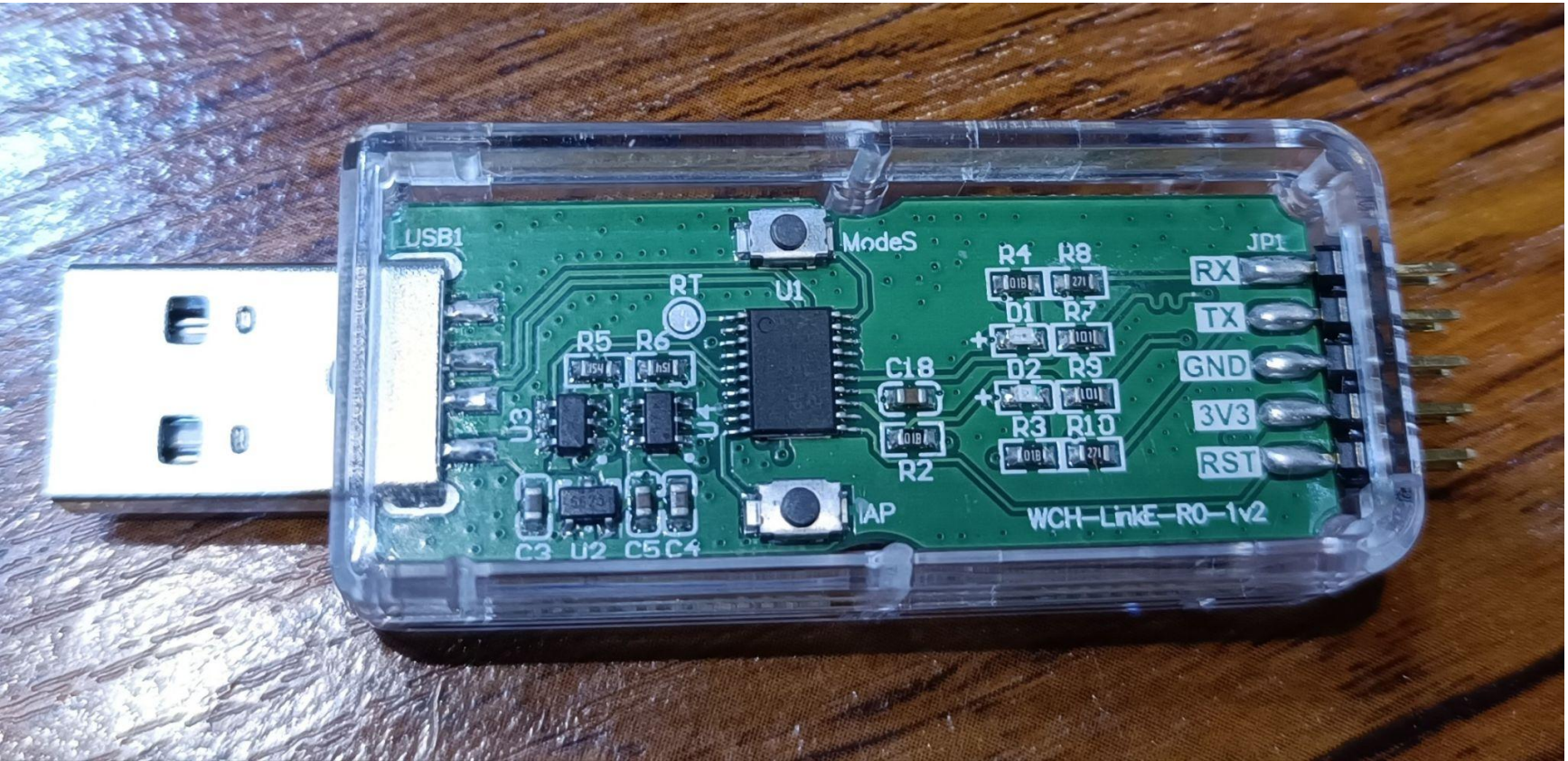
- Where to put the ‘HAL’ bits caused much code churn (“re-writes”)

- Licensing issues around ‘ch32fun headers’ took a while to resolve (Thank you **cnlohr - Charles and WCH - Patrick)**

Upstreaming can be a lengthy process

- It looks us around 6 months to get our upstream Zephyr PR merged!
- I was working on it very sporadically in my free time (some part of weekends).
- https://github.com/zephyrproject-rtos/zephyr/pull/73761

WCH-LinkE is NOT needed!



Bonus: CH32V003 programming via PicoRVD

- The proprietary WCH programmer is NO longer required!
- A general-purpose Raspberry Pi Pico works great as a programmer - thanks to the PicoRVD project (by https://github.com/aappleby).
- https://github.com/kholia/picorvd (customized and maintained fork)
- We will open an upstream Zephyr PR to add support for PicoRVD programmer soon

Next Zephyr target - Puya PY32F - Recon

- PY32F003 SOP-8 package costs only 0.23 USD. Much lesser in bulk quantities (0.13 USD). Available @ https://www.etstore.in in India

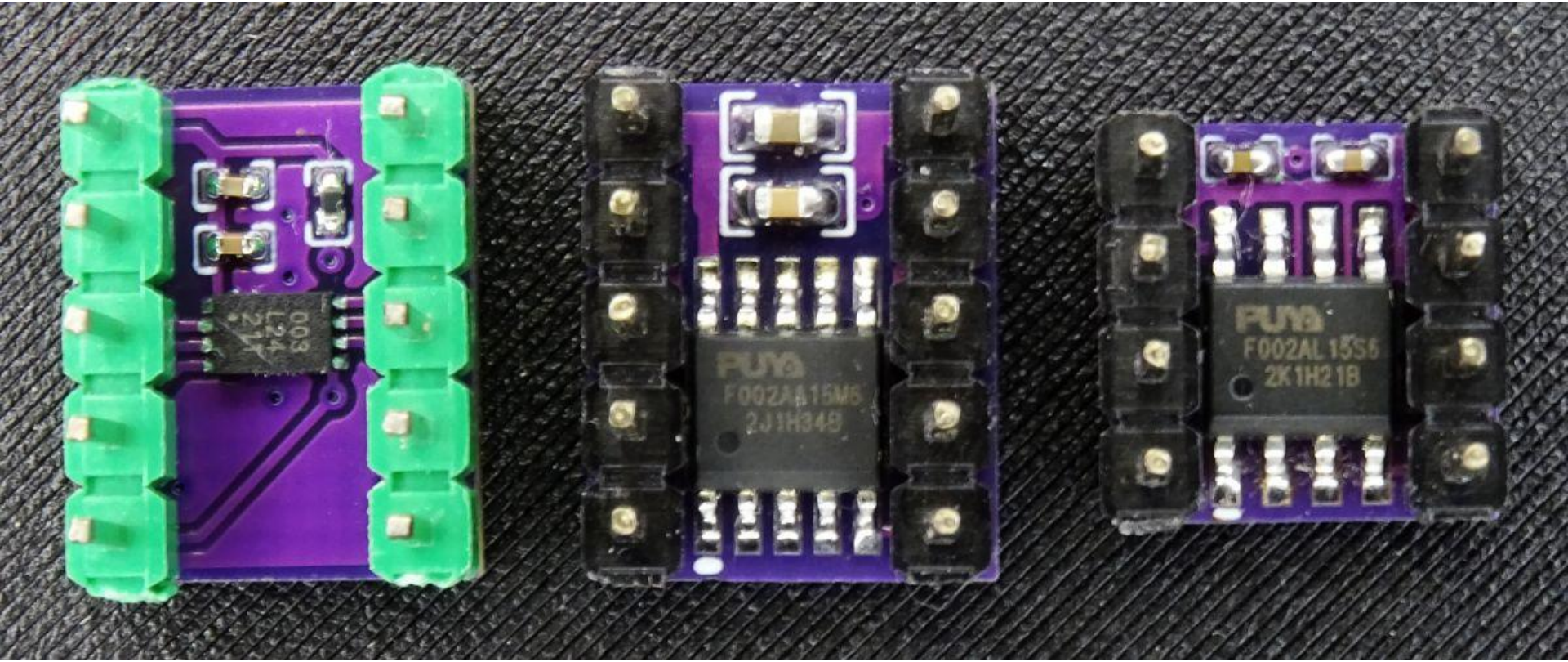
- Familiar ARM Cortex M0+ core! Cost-effective replacement for STM32's STM32G031J4M6 part.

- Has SWD interface - easy to program.

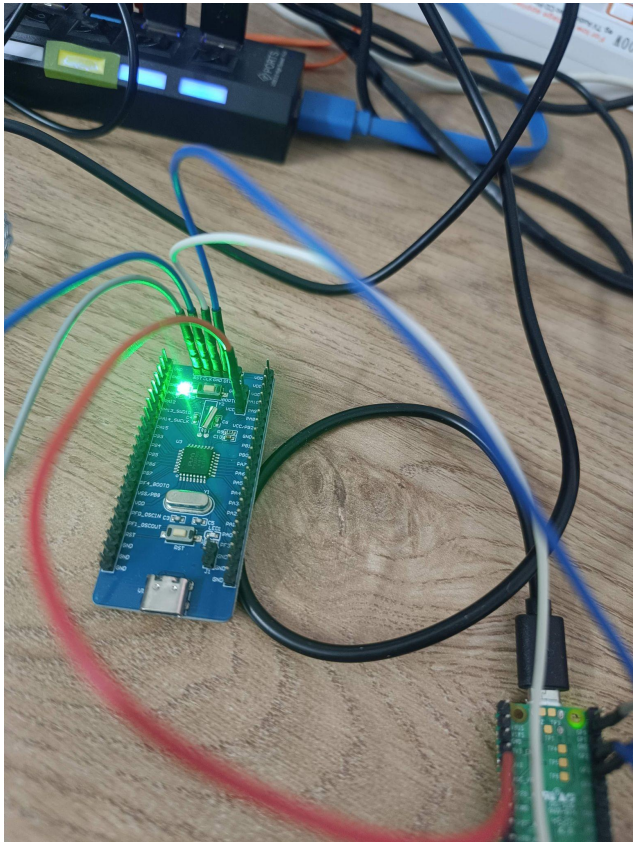
- It seems PY32F030 is some mashup of STM32F030 chips with "IOPORT" stuff borrowed from STMG0 family.

- I didn't have to write any ASM. PY32F030 boots with the STM32F030 low-level ASM bits just fine. **PHEW!**

PY32F packages



Next Zephyr target - Puya PY32F - Progress Report



Next Zephyr target - Puya PY32F - Progress Report

- https://github.com/kholia/zephyr/ has a branch which adds hacky PY32F support!
  https://github.com/kholia/hal_py32/ contains the required ‘HAL glue’ (more like LL glue).

- Standing on the shoulder of giants - https://github.com/IOsetting/py32f0-template

- !!! Zephyr support for PY32F is VERY HACKY at the moment !!!

- Needs intensive clean-up efforts before upstreaming can start!

Learnings - (Obvious) porting tips for small MCUs

- Reduce Zephyr's flash and RAM footprint, if needed

  MAIN_STACK_SIZE 512

  IDLE_STACK_SIZE 256

  ISR_STACK_SIZE 256

- Get blinky working first (implement delay using large for-loops, if necessary)

- Try getting UART up next (UART is required for minimal upstream PR)

- More peripherals can come later in follow-up pull requests

Future work

- Start upstreaming efforts for adding support for Puya PY32F chips in Zephyr

- Port my EEPROM emulation (I2C slave) code to Zephyr and publish it. Does it still work fast enough?

Project ideas

- Port https://github.com/marcan/sigmafix to WCH / Puya SOP-8 chips
- Build an EEPROM emulator for security research
- “AirTag” / Google Tag for under 128 INR, anyone?
- Mouse jiggler - https://github.com/wagiminator/CH32V003-Mouse-Wiggler
- Custom IR remotes - https://github.com/wagiminator/CH32V003-IR-Remote

References

- https://sirinsoftware.com/blog/rtos-wars-freertos-vs-zephyr-a-decision-you-cant-afford-to-get-wrong

- https://www.zephyrproject.org/wp-content/uploads/2024/12/Porting-Zephyr-to-the-CH32V003.pdf

- https://zephyrproject.org/2-boards-you-can-finally-use-with-zephyr/

Thanks!

- Thank you for your time and attention today!