



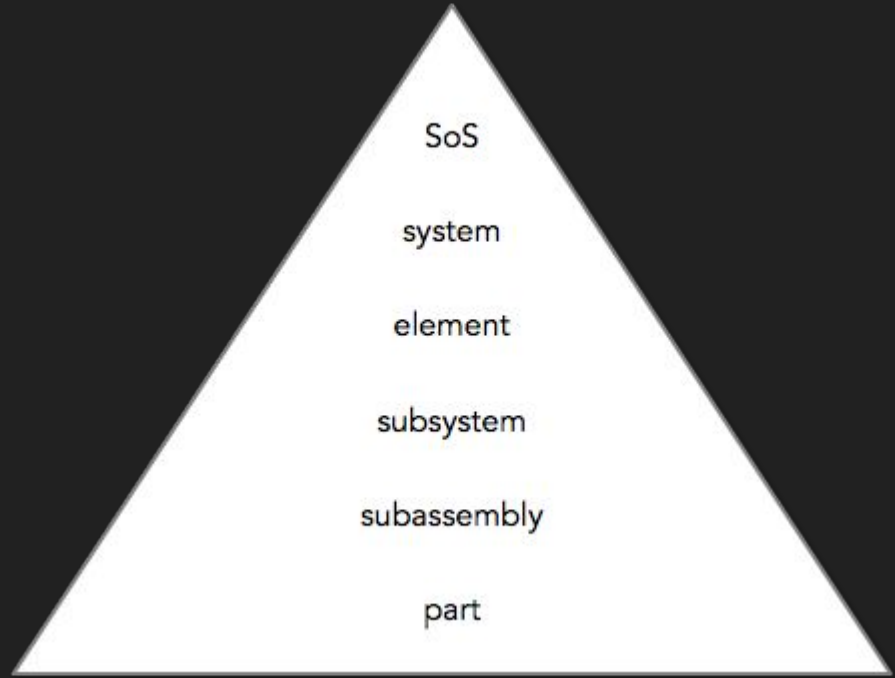
Test Like You Fly: Implementing On-Target Testing for the Libuavcan Project.

Scott A Dixon :: UAVCAN :: libuavcan

First, the “why”

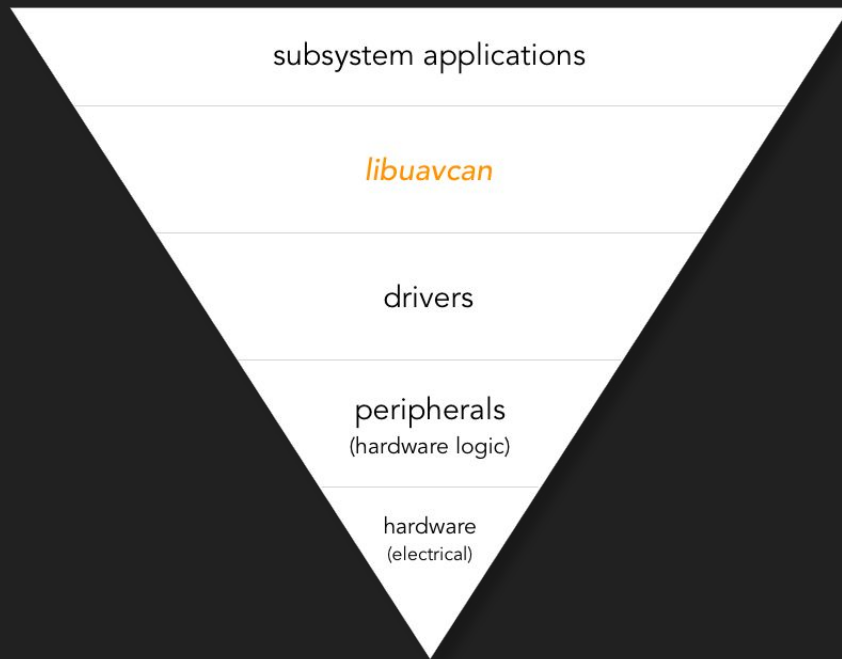
“The system should never experience expected operations, environments, stresses or their combinations for the first time during the mission

The Progression of Integration Level Tests



(Adapted from: White, Julie; Tilney, Lindsay “Applying the “Test Like You Fly” (TLYF) Process To Flight Software Testing” Workshop of Spacecraft Flight Software 2013,
http://flightsoftware.jhuapl.edu/files/2013/talks/FSW-13-TALKS/TLYF_Apply2FSW_Dec2013r1.pdf)

What's below libuavcan?





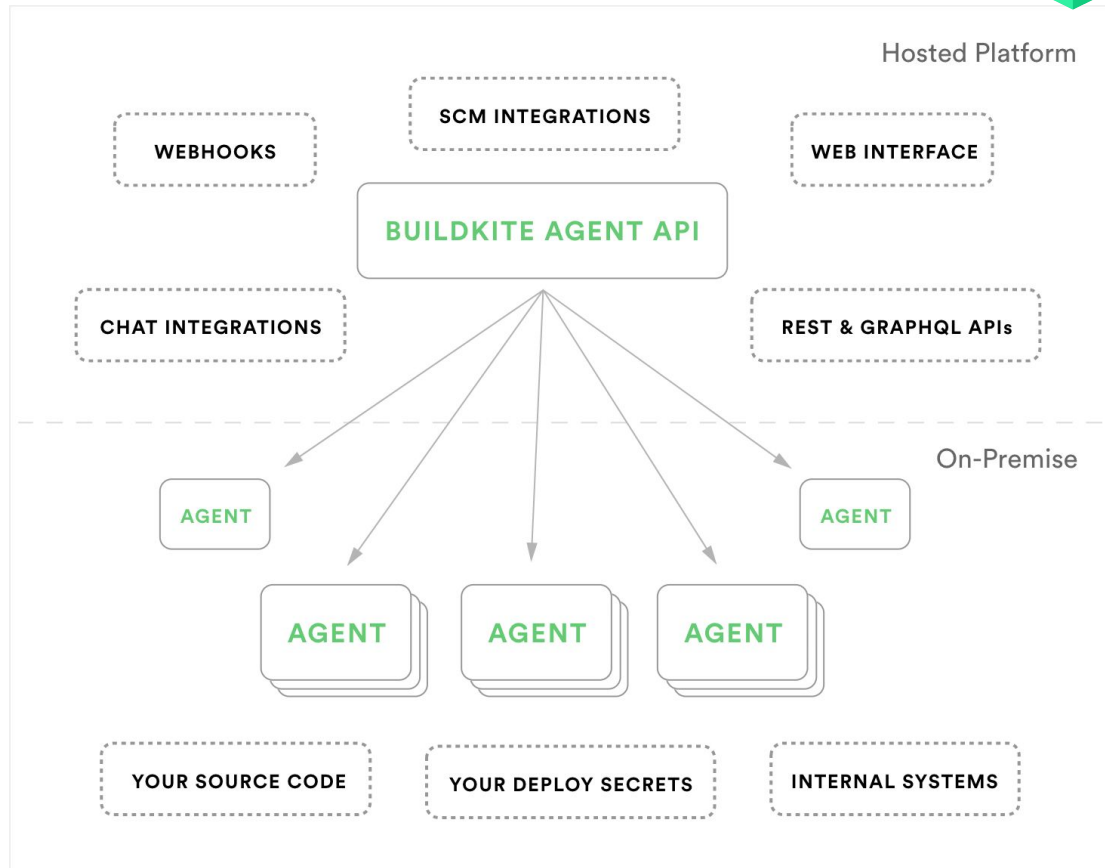
Test Like You Fly

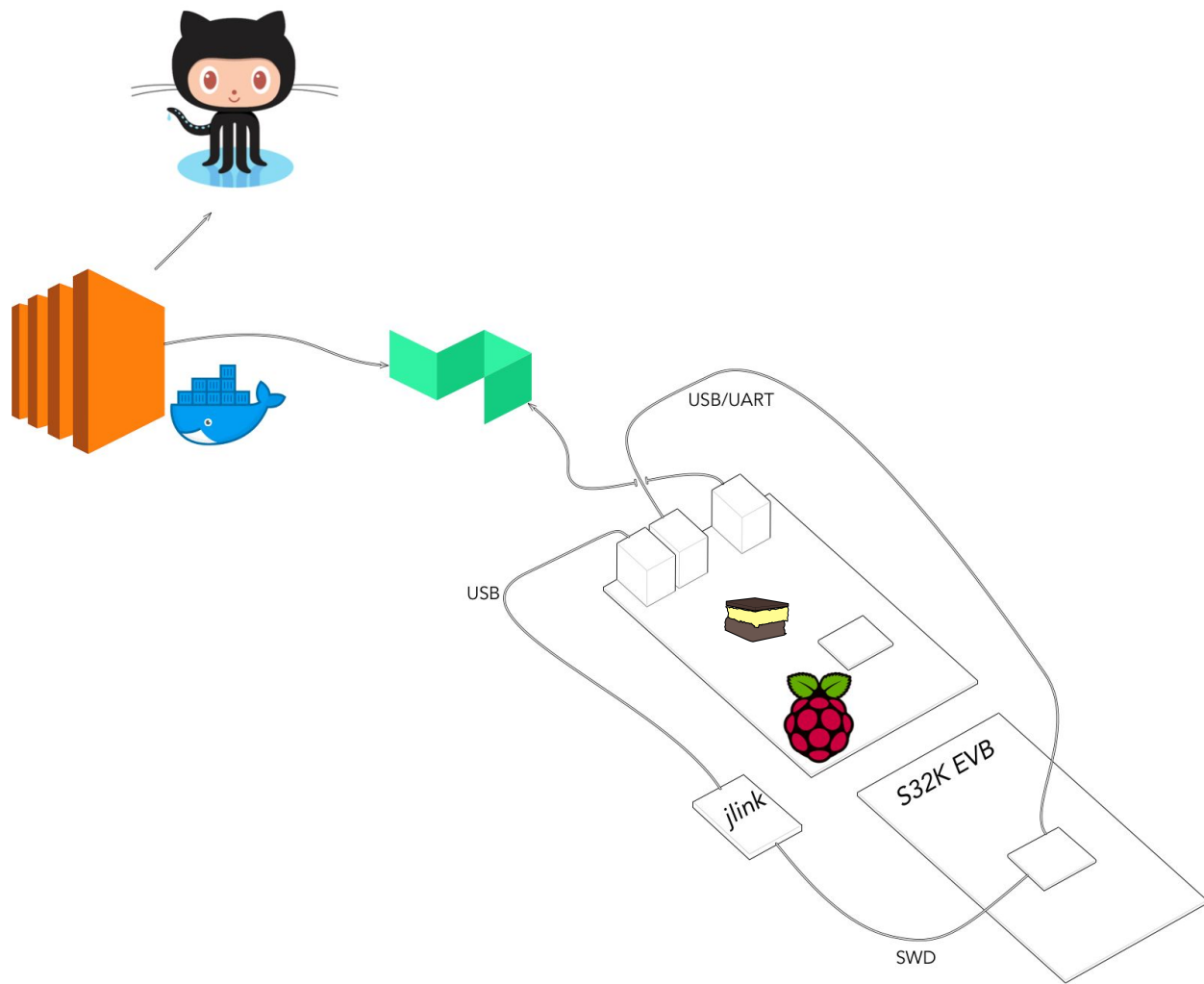


Open Source Hardware and Services



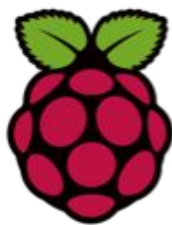
Buildkite







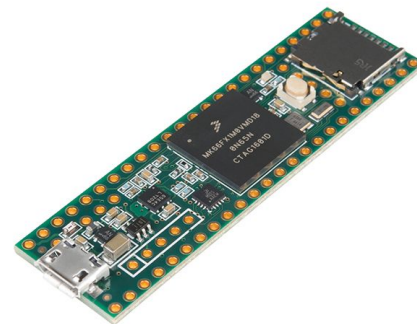
\$10-100 (USD) per month
(pay more for faster builds)



\$40 (USD) each



\$25 (USD) each



\$20-200 (USD) each

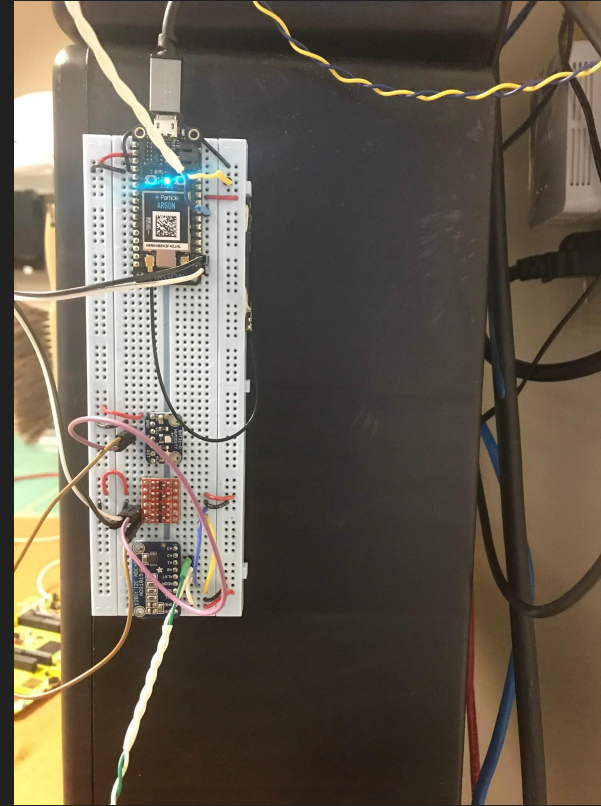
~\$20MM (USD)



<https://buildkite.com/uavcan/libuavcan-v1/builds?branch=uavcan-v1.0>

[illegible]

Humble Beginnings

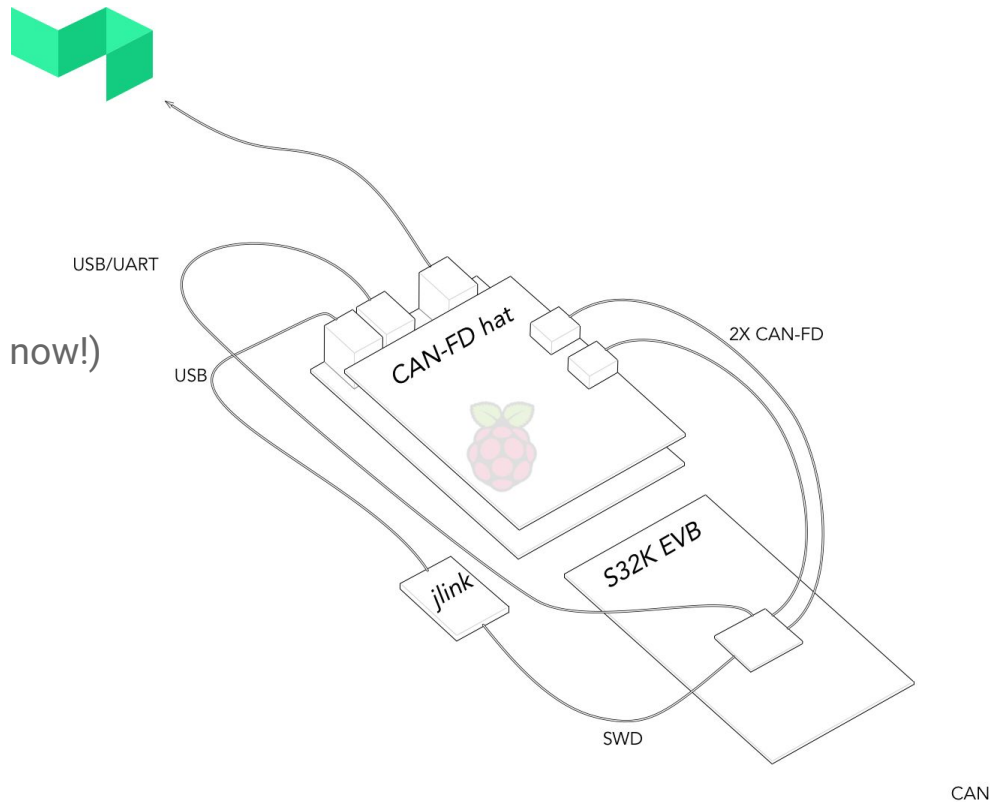


Building googletest with no system.

- <https://github.com/UAVCAN/libuavcan/blob/uavcan-v1.0/cmake/modules/Findgtest.cmake>
- <https://github.com/UAVCAN/libuavcan/blob/uavcan-v1.0/test/ontarget/S32K148EVB/src/main.cpp>
- https://github.com/UAVCAN/libuavcan/blob/uavcan-v1.0/test/ontarget/S32K148EVB/src/test_sys.c

Roadmap

1. Real-CAN bus
2. Non-googletest (using 116K heap right now!)
3. SWO
4. ETM
5. Resource constrained (M0)
6. Performance
7. Pixhawk?



Enabling SWO After Reset

- <https://www.segger.com/downloads/jlink/UM08001>

And you can too! (for great justice)

1. Get a pi (or whatever).
2. Get some hardware you want to test.
3. Implement an on-target test driver for libuavcan.
4. Get a token



Questions?

<https://32bits.io/blog/2019/5/26/libuavcan-hil-test-automation-using-buildkite-and-raspberry-pi>