



Testing in a Box: Streamlining Embedded Systems Testing

Speaking Today



TITLE

Software Engineer

TENURE

Joined Codethink in
2022

SPECIALTIES

Embedded Systems

EDUCATION

Mechatronics & Robotics
University of Leeds

Codethink



Speaking Today



TITLE

Senior Engineer

TENURE

Joined Codethink in
2017

SPECIALTIES

Linux systems integration

Custom Embedded:

- Custom Distributions
- Hardware testing

EDUCATION

University of Brunel

Aerospace Engineering

Codethink



Contents

- 01 Context – Testing
- 02 Project Requirements
- 03 TIAB Implementation



Introduction – Codethink

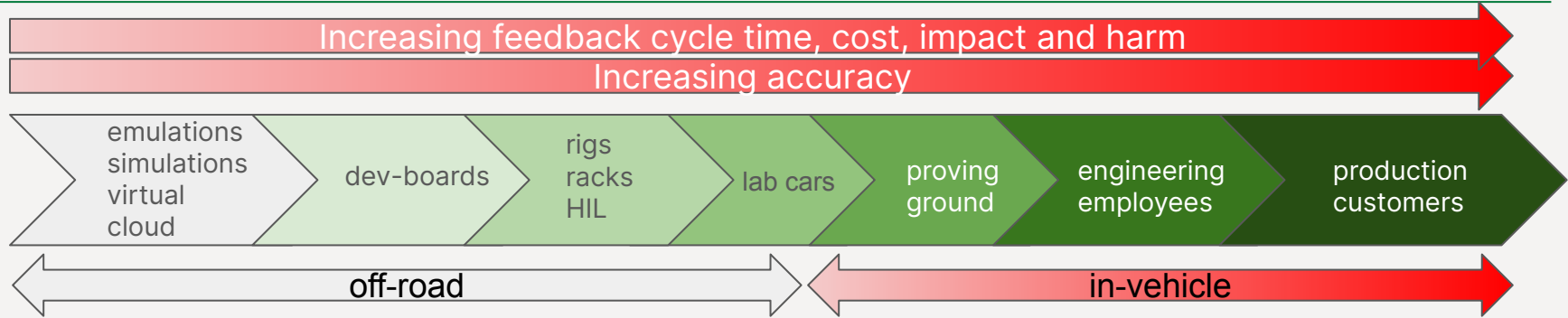


- Offering software services for over 15 years
- Mostly related to open source software
- Lots of Automotive and Financial services clients
- Specialize in build and integration of complex systems



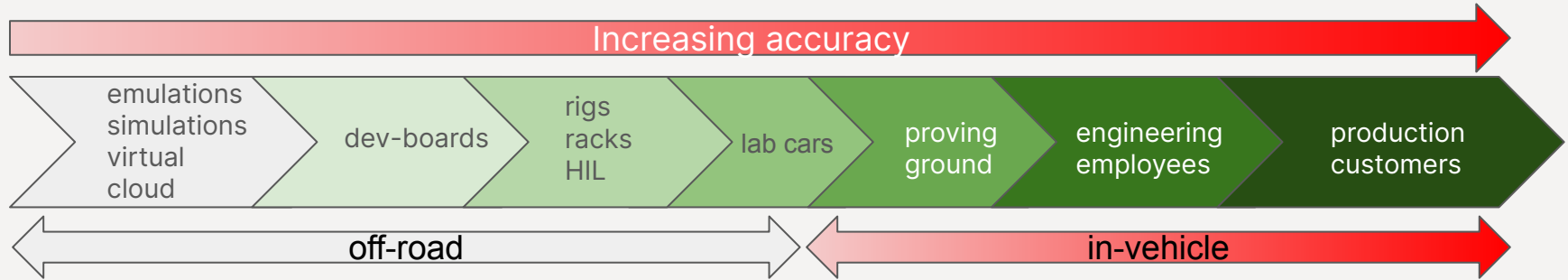
Context – Testing

Test early and often



- At Codethink we have testing as a key pillar of our approach
- We try to test as much and as often as possible
- We try to keep cycle times low
- But some things can only be tested on the final system
 - Rig/VM
 - Server
 - Car

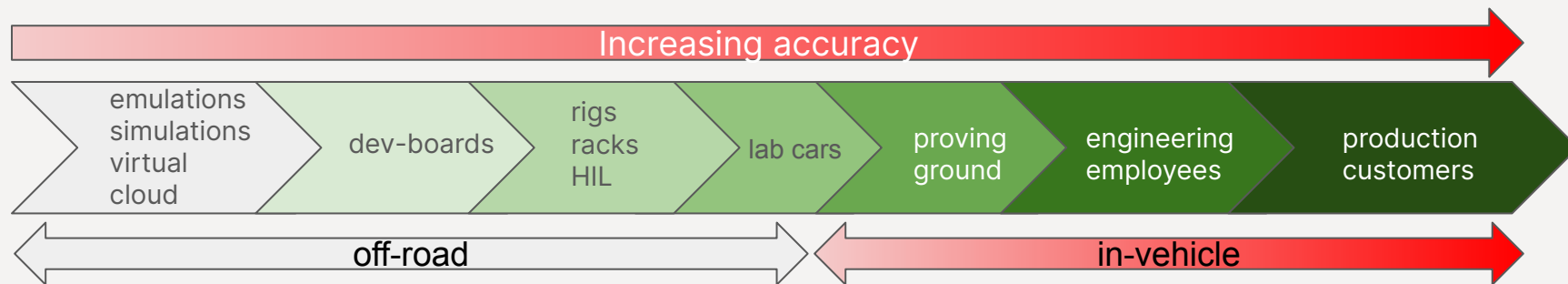
Test early and often on infrastructure



As we go to the right we run up against

- Slow Automated tests
- Manual tests
- Expensive tests
- Resource constrained tests

Test early and often for devs



Accurate

- Keep the tests as close to PROD as possible
 - Hardware
 - Peripherals
 - Network

Easy

- Tests should be quick
- Tests should be reproducible / repeatable
- Test should be easy for devs to trigger
- Test should be easy for devs to debug

Testing on hardware

Testing Rigs and Cars



Rig tests in many companies are very MANUAL.

- Things to automate
 - Full rig flash
 - OTA updates
 - UI interactions
 - CAN updates
 - Peripherals
 - Connection actions
 - Peripheral state dependent action



Final requirements



- CI only rigs
- No way to merge without automatic tests
- Test automation co-ordination
- UI tests
- Automated hardware control
 - Can transceivers
 - GPIO
 - Peripheral control

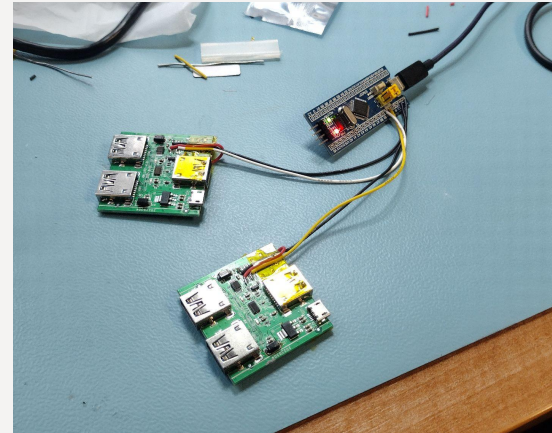


Testing infrastructure

Requirements for testing infrastructure



- Control hardware to run tests on device under test
 - A mini PC or a laptop
- Hardware to test/simulate production application
 - CAN dongle
 - JTAG
 - Serial
 - GPIO
 - Custom hardware for niche applications



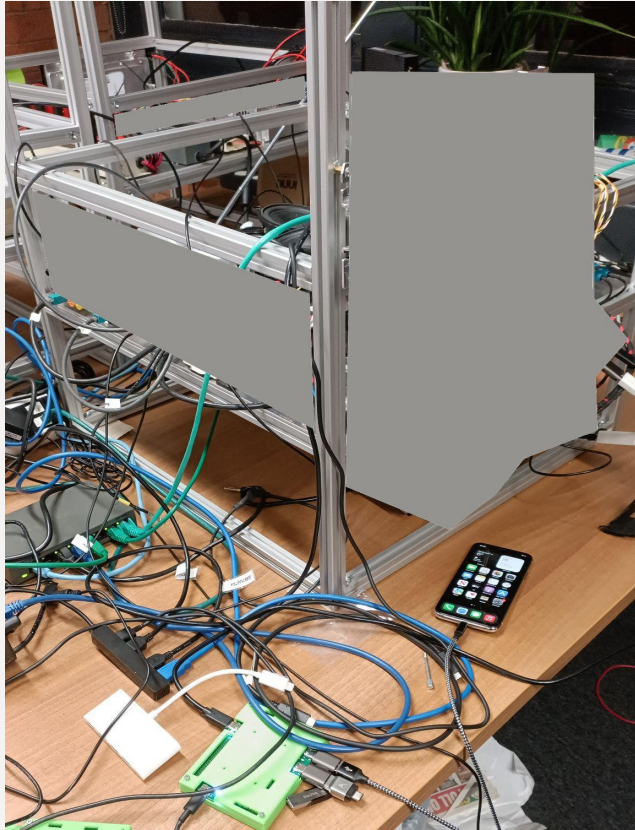
USB switch



- Bi-directional USB-C switch
- Allow 1 host to switch between 2 peripherals and vice-versa, programmatically
- Open source hardware, firmware and case
- Supports USB Super Speed
- EMC tested and certified
- <https://gitlab.com/CodethinkLabs/usb-switch/>



Resulting test setup



- Lots of capability
- Bit of a uncontrolled mess
- Very easy for things to get upset
- Not easy to replicate

Considerations for robust testing infrastructure

- Supply chain
 - Can you buy more of this IO hardware in next 5 years?
- Ease of setup
 - How long does it take to setup testing infrastructure for a rig?
- Consistency
 - Will different hardware claiming to do the same thing actually do it consistently?



TIAB Implementation

Hardware overview



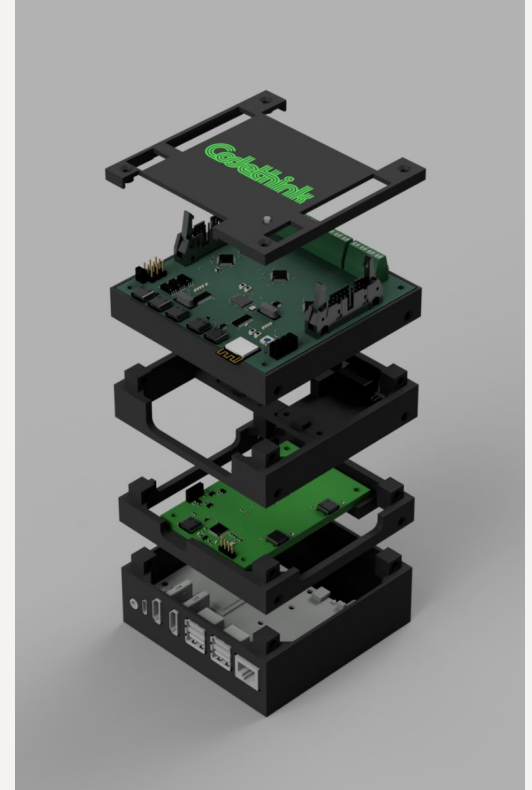
- Multitool for testing
- IO supported
 - Serial
 - Optocouplers
 - GPIO
 - SPI and I2C
 - HID emulation
 - USB hub
- Modular design for multiple layer stackability and expandability



Hardware overview



- Arm-based SBC as control unit
- USB switch
- CAN modules
- Open source IO board hardware, case, and config



Testing In A Box



- Coordination GitLab + openQA worker
- Rig control
 - Runner, companion computer to the, System Under Test, SUT.
 - QAD
 - QAnvas
 - USB switch
 - GPIO
 - CAN
- System under test

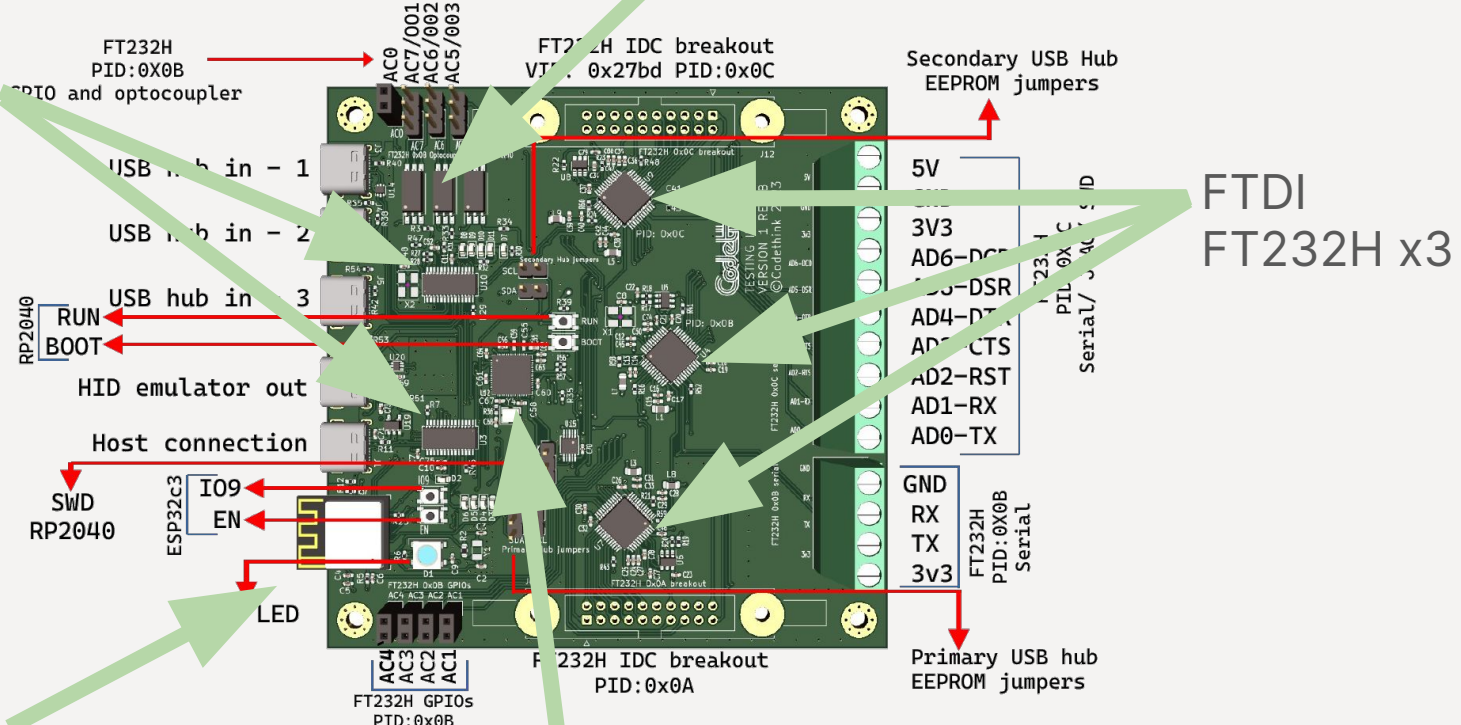


GPIO board

Opto couplers
TLP2748



FE1.1s(rev-b)



For now
Espressif
ESP32-s3

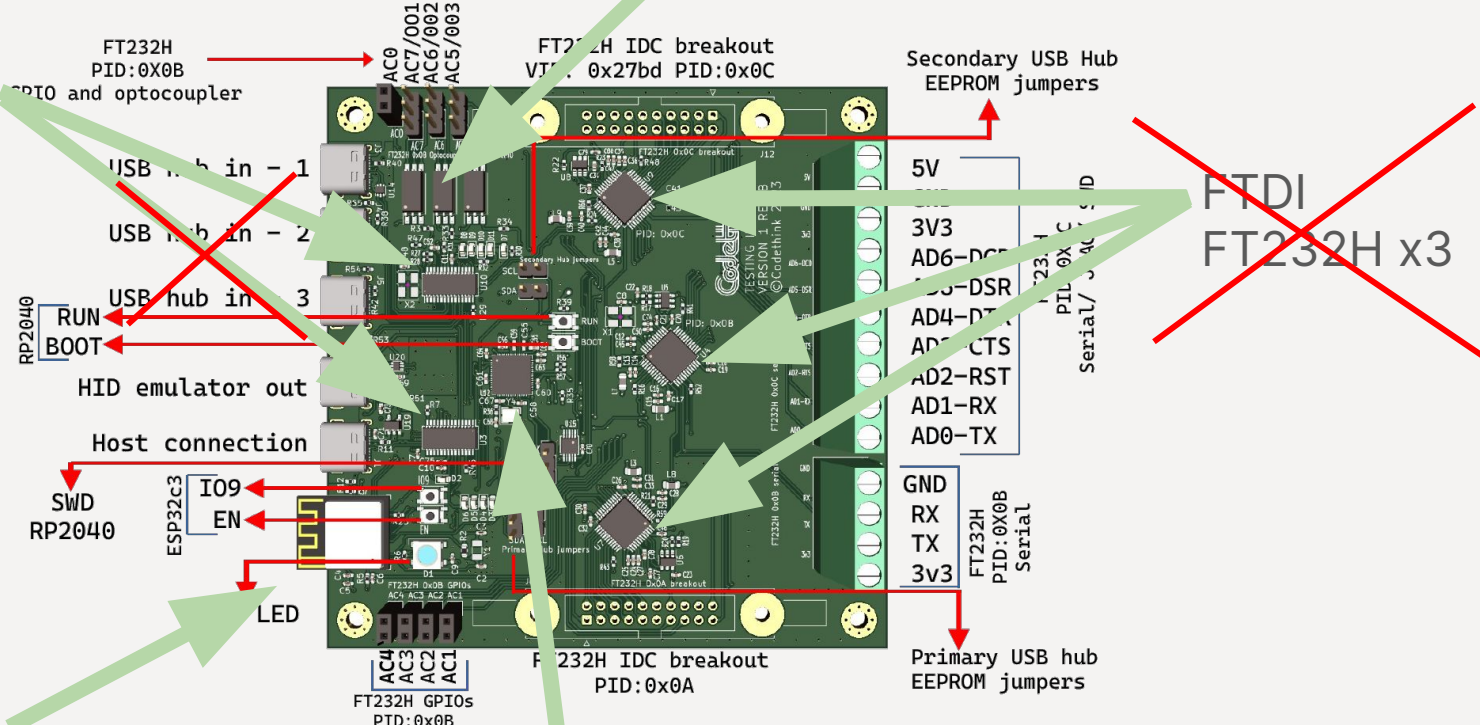
Raspberry Pi
RP2040

GPIO board

Opto couplers
TLP2748



FE1.1s(rev-b)



~~For now
Espressif
ESP32-s3~~

Raspberry Pi
RP2040

GPIO in future



- FTDI FT232H x3 → FT423-2H x1
- ESP32s3 → the RP2040 does 90% of what we actually used the ESP32 for
- Simpler and cheaper
- A lot less firmware to manage!



Example



Testing in a box software:

- Test Coordinator
- Test Runner
- UI tests Capture

Testing in a box:

- GPIO
- Full Flash
- Peripherals

What else you do get in package?



- Ansible scripts for setting up the control machine:
 - GitLab runner
 - udev rules
 - openQA worker
 - CI templates
- Ansible to setup a server with pre-installed tests for AGL on a Pi but can be adapted to any system
 - UI tests
 - CAN + UI tests
 - Networking

What's next



- Version 2 for TIAB IO board
- CAN FD expansion board for TIAB
- Much more to come

Get involved:

TIAB group:

<https://gitlab.com/CodethinkLabs/testing-in-a-box/>





Thank You.

Codethink Ltd.

3rd Floor Dale House,
35 Dale Street,
MANCHESTER,
M1 2HF,
United Kingdom.

