Automated Testing for Embedded development

Next generation Board Farming

Chris Fiege – cfi@pengutronix.de

About Pengutronix

Most active 5.14 employers

By changesets			By lines changed		
Huawei Technologies	1731	11.7%	AMD	293439	28.6%
Intel	1331	9.0%	Intel	135564	13.2%
(Unknown)	1003	6.8%	(Consultant)	50998	5.0%
AMD	879	6.0%	Broadcom	47742	4.7%
Red Hat	854	5.8%	Linaro	33652	3.3%
Google	756	5.1%	Red Hat	30978	3.0%
(None)	744	5.0%	Huawei Technologies	29704	2.9%
Linaro	654	4.4%	(Unknown)	29631	2.9%
SUSE	503	3.4%	Google	29387	2.9%
IBM	445	3.0%	NVIDIA	28415	2.8%
NVIDIA	319	2.2%	(None)	23154	2.3%
Oracle	290	2.0%	IBM	22541	2.2%
Canonical	278	1.9%	SUSE	19887	1.9%
NXP Semiconductors	276	1.9%	Marvell	17294	1.7%
Facebook	274	1.9%	Microchip Technology	14852	1.4%
Arm	255	1.7%	NXP Semiconductors	12200	1.2%
(Consultant)	229	1.6%	Arm	11831	1.2%
Renesas Electronics	203	1.4%	SoMainline	10599	1.0%
Linux Foundation	170	1.2%	Socionext Inc.	10526	1.0%
Pengutronix	151	1.0%	Code Aurora Forum	10050	1.0%

About Me

Chris Fiege
Senior Hardware Developer

cfi@pengutronix.de

SmithChart

SmithChart

Why talk about Board Farming?

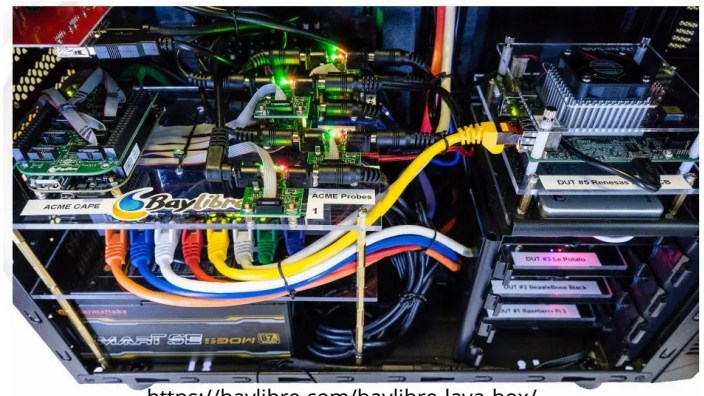
Fuego Logo created by Tim Bird, CC BY 4.0 US

https://docs.lavasoftware.org/lava/index.html

More at elinux.org:

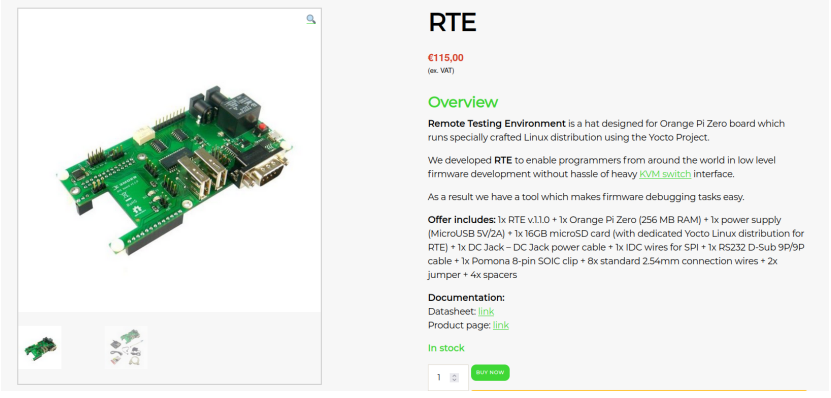
- *https://elinux.org/Board_Farm#Software*

Why talk about Board Farming?



https://baylibre.com/baylibre-lava-box/

https://www.youtube.com/watch?v=Hx9MEhR05cU



https://3mdeb.com/shop/

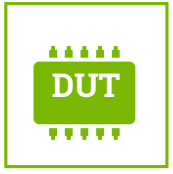
(Add your custom solution here.)

Topics

- What is Board Farming and why you may want to do it?
- How Pengutronix does it
- Pros and Cons for the current approach
- Next generation Board Farm

- Discussion

Controlling an Embedded Linux Device?



Board farming?

- Automated control of devices
- Working from remote and board sharing
- Run tests
- Continuous testing

Board farming is an important foundation for quality assurance for all software on real hardware.

Pengutronix: Use cases

Interactive

- Work on a software component on real hardware.
- Run or develop a test suite.

Automated

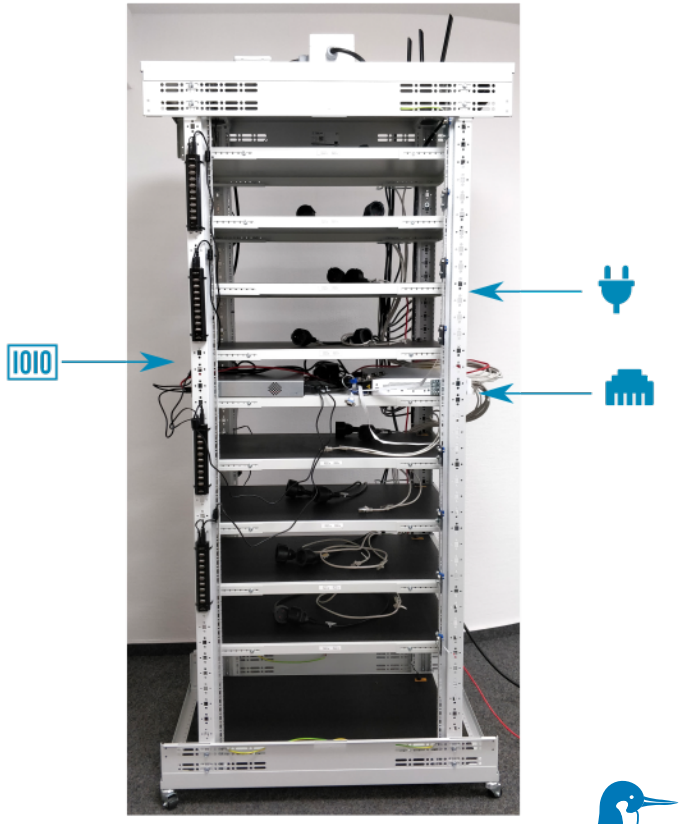
```python
import re

import pytest

from labgrid.driver import ExecutionError


def test_memory_memtester_short(shell):
    """Test RAM for errors"""
    try:
        shell.run_check('which memtester')
    except ExecutionError:
        pytest.skip("memtester missing")

    result = shell.run_check('memtester 128k 1 | tail -n 1')
    result = result[-1].strip()

    assert result == "Done."
```

- Testsuite executed by CI without human interaction.

Pengutronix Board Farm: Hardware

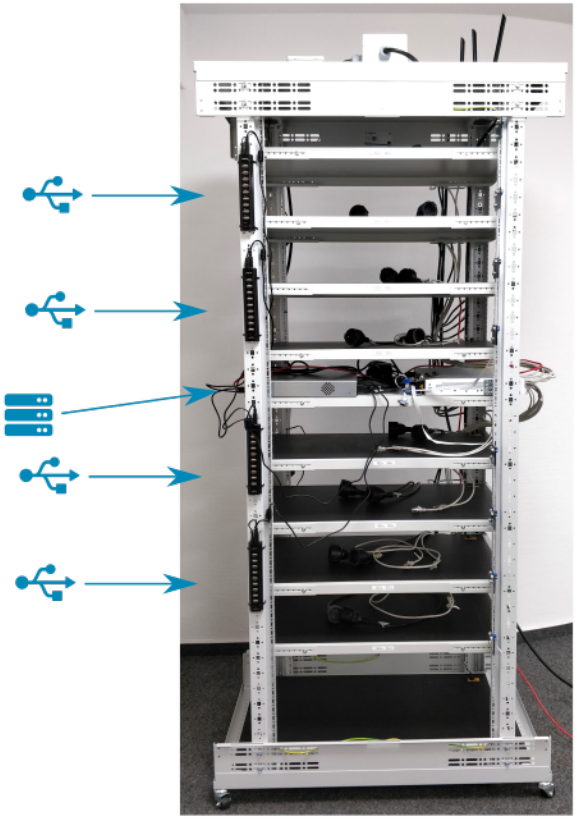
Pengutronix Board Farm: Hardware

- 16x or 24x Power Switch
- 24x Ethernet
- RS232 Serial Server



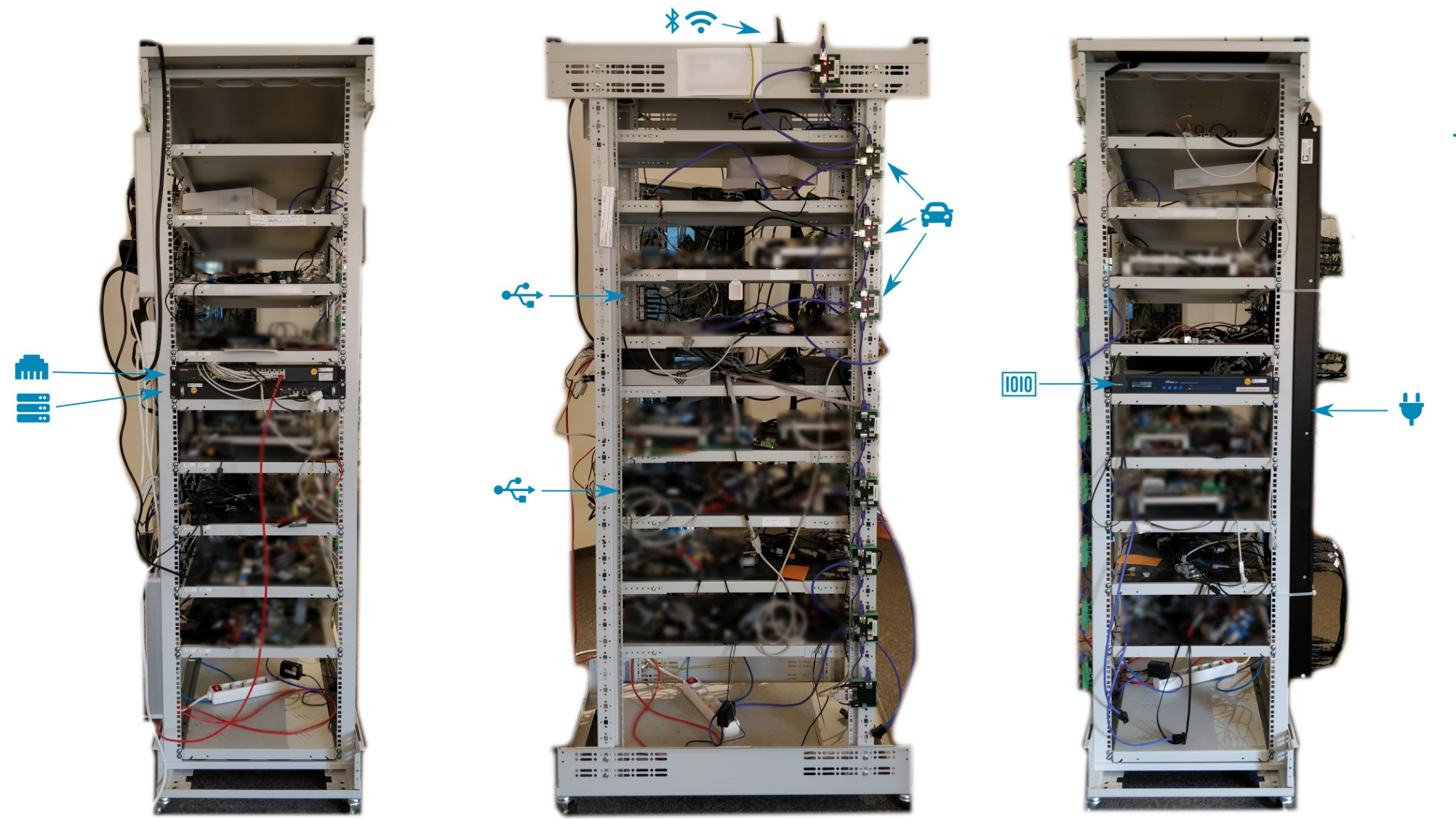
Pengutronix Board Farm: Hardware

- Test-Server
  - USB Host
    - 1-Wire via USB
      - GPIO
    - CAN
    - 20x USB
  - USB Host
    - 20x USB



Pengutronix Board Farm: Hardware

- Wifi-AP
- Bluetooth Beacon


1010

Pengutronix Board Farm: Hardware

- 8 racks with 128 slots (total) in our regular lab
- 10+ labs with 4 .. 16 slots each on desks

Pengutronix Board Farm: Interactive

https://docs.pytest.org/

Pengutronix Board Farm: Continuous Testing

https://jenkins.io

LAB GRID

GitLab

https://about.gitlab.com

pytest

https://docs.pytest.org/

Labgrid

- Documentation:
- https://labgrid.readthedocs.io/en/latest/

- **„About the joy and tears of testing Embedded Devices“**
  https://fosdem.org/2021/schedule/event/testing_embedded_devices/

What works well?

Single hardware pool for interactive and CI/CT

⊕ Only a single board is needed

⊕ Easier debugging of failing tests

⊕ Every developer can access every board in the lab

What works well?

Single hardware pool for interactive and CI/CT

⊕ Only one set of hardware is needed

⊕ Easier debugging of failing tests

⊕ Every developer can access every board in the lab

⊕Provisioning of DUT from scratch => deterministic results

What works well?

Labs are built to be versatile

⊕ Every hardware can be placed in every lab

Problems

Devices are connected with many cables

⊖ Chance of accidental disconnection when working on an adjacent device

⊖ Moving a device into another lab is error prone

Problems

Infrastructure devices are a black box

⊖ e.g. serial-server missing characters

⊖ e.g. power switches not providing IPv6

⊖ e.g. power switches not responding

Expensive Problems

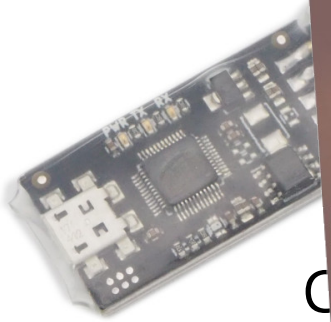
USB

Expensive Problems

USB

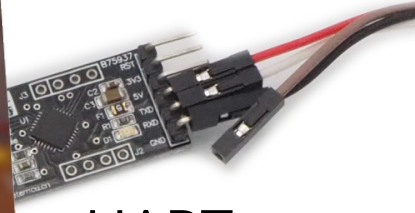
⊕ easy to use, widely available

Expensive Problems

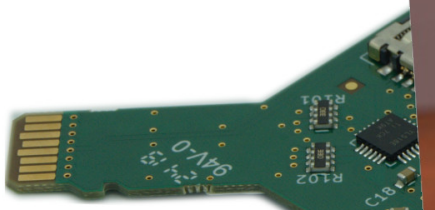


C

RS232



UART



USB-SD-Mux

USB-Mux

Expensive Problems

USB is a bad idea

⊕ easy to use, widely available

⊖ stability issues

⊖ consumer USB devices have bugs

⊖ hard to debug

Expensive Problems

USB is a bad idea

⊕ easy to use, widely available

⊖ stability issues

⊖ consumer USB devices have bugs

⊖ hard to debug

⊖ often needs „remote hands“ to recover

⊖ problems often affect neighbouring slots

⊖ costs time/money and nerves!

Potential Solution: Less Black Boxes

- 16x or 24x Power Switch
- 24x Ethernet
- RS232 Serial Server
- Test-Server
  - USB Host
    - 1-Wire via USB
      - GPIO
    - CAN
    - 20x USB
  - USB Host
    - 20x USB
- Wifi-AP
- Bluetooth Beacon

Potential Solution: Smaller USB Bus-Size

- 16x or 24x Power Switch
- 24x Ethernet
- RS232 Serial Server
- Test-Server
  - USB Host
    - 1-Wire via USB
      - GPIO
    - CAN
    - 20x USB
  - USB Host
    - 20x USB
- Wifi-AP
- Bluetooth Beacon

Potential Solution: Less Cables

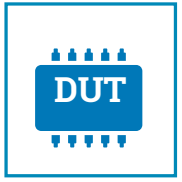
- 16x or 24x Power Switch
- 24x Ethernet
- RS232 Serial Server
- Test-Server
  - USB Host
    - 1-Wire via USB
      - GPIO
    - CAN
    - 20x USB
  - USB Host
    - 20x USB

- Wifi-AP
- Bluetooth Beacon

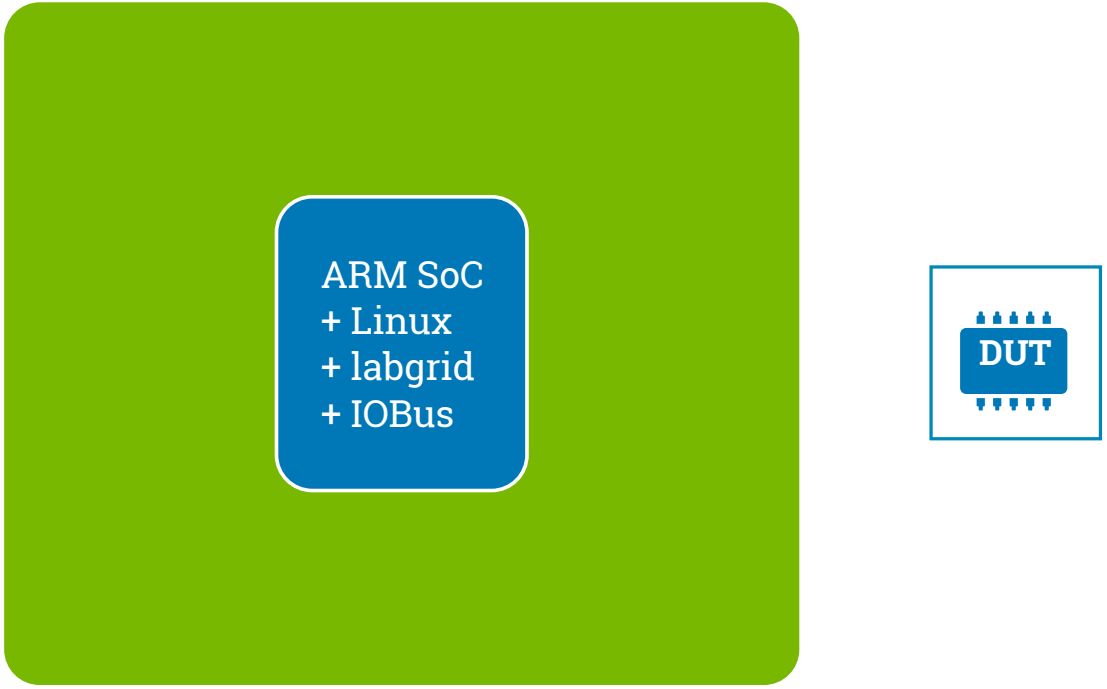
Test Automation Controller

Concept:

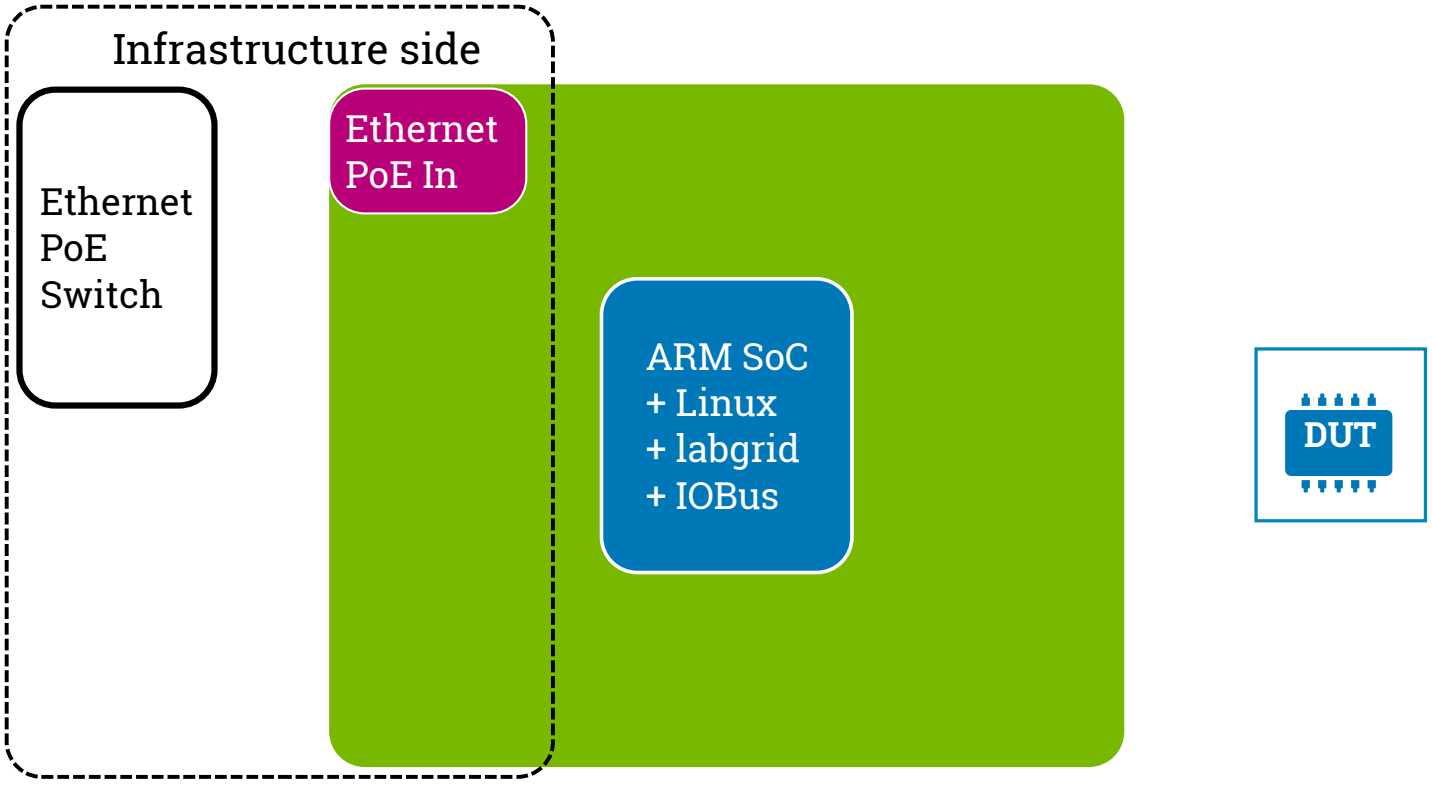
- Cover 80% of the use cases in everyday use
- More robust than current setup
- Built for everyday use



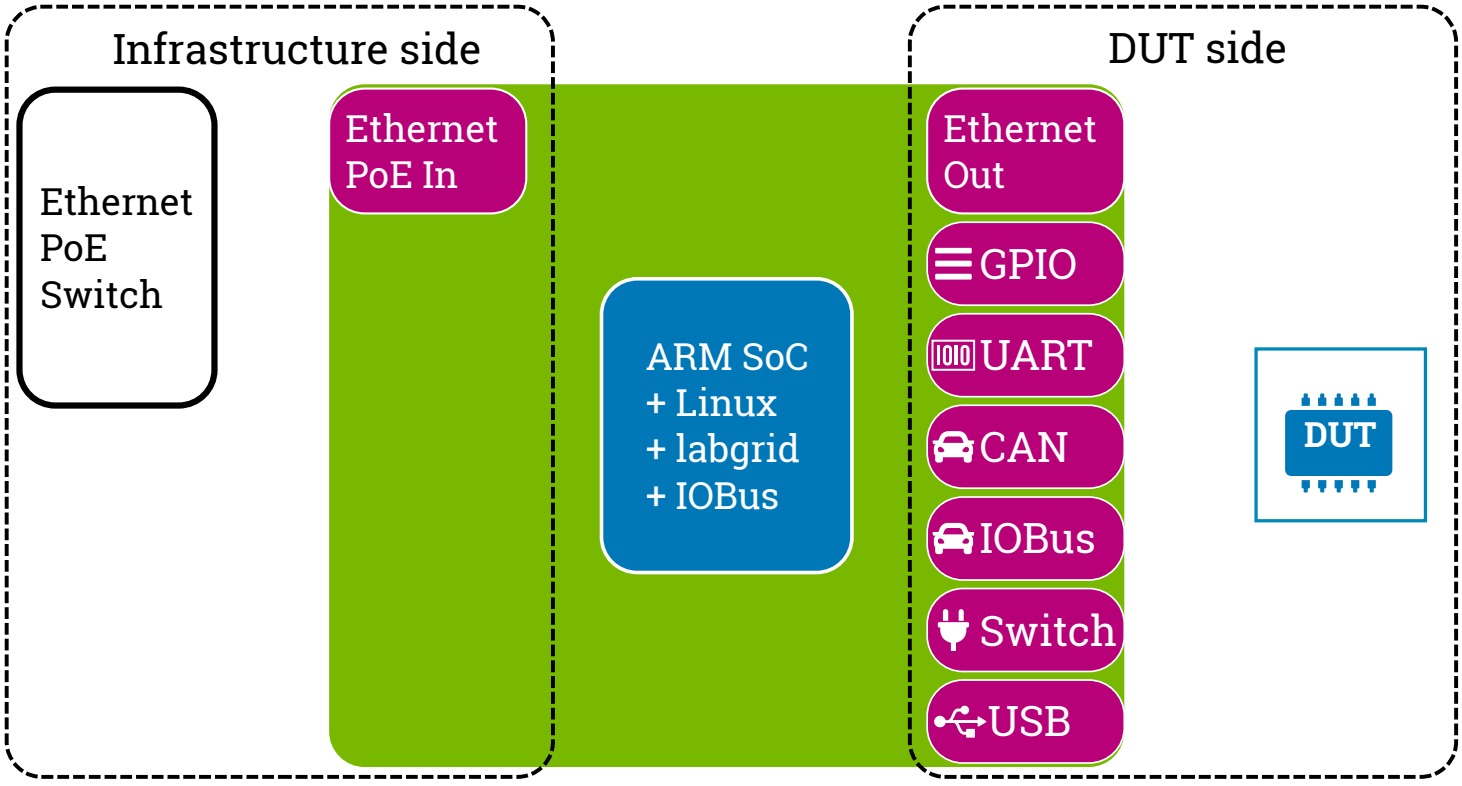
Test Automation Controller

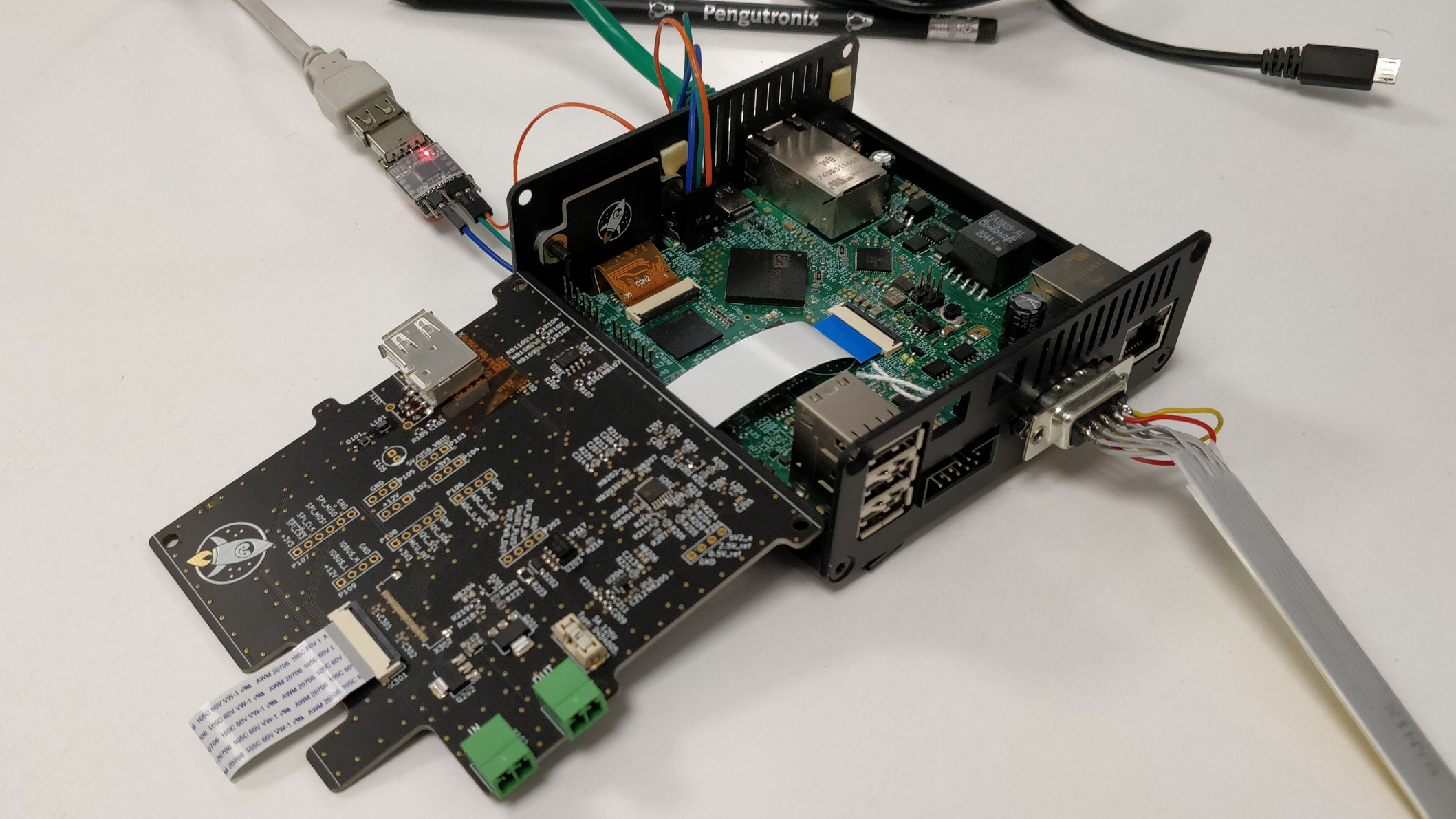


Test Automation Controller



Test Automation Controller




Pengutronix

Test Automation Controller

Current state:

- Hardware: V1
- Software: In active development

Next step:

- Concept validation

New problems:

- FIXME

Thank you!

Questions and Discussion!