THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
EUROPE

U-Boot in the Fast Lane: Developing a Safety Test Framework for Automotive Applications

Neha Malcom Francis

August 27th, 2025

About us: TI Processors and Open Source

Upstream FIRST!

- ✔ Upstream FIRST mentality!
- ✔ Decades of contribution and collaboration
- ✔ Focus on long term, sustainable and quality products
- ✔ Upstream & opensource ecosystem in device architecture
- ✔ Ingrained culture to give back to the community

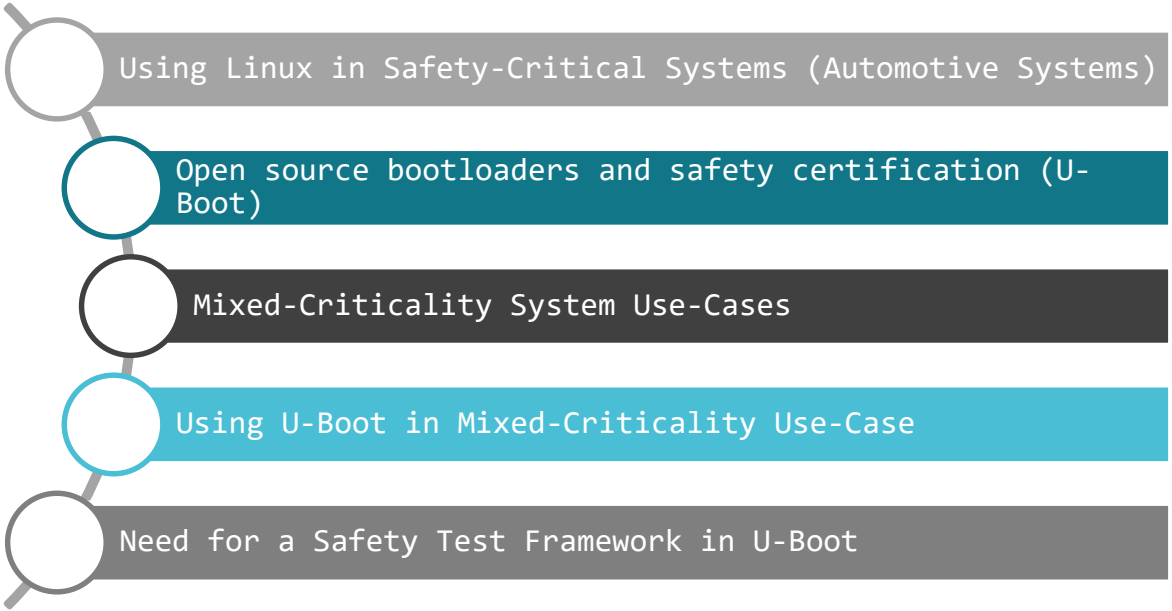
Speaker Introduction

```
Neha-Malcom-Francis {
    software-engineer;
    Texas-Instruments;
    location = “Bangalore, KA, India”;
    works-on = “u-boot”, “ddr”, “ufs”, “Jacinto”;
    hobbies = “dancing”, “travelling”, “painting”;
};
```

Disclaimers

- This is a technology presentation, not product-readiness or roadmap commitment
- Opinions presented here are that of the speakers and may not reflect that of Texas Instruments Inc.

Overview

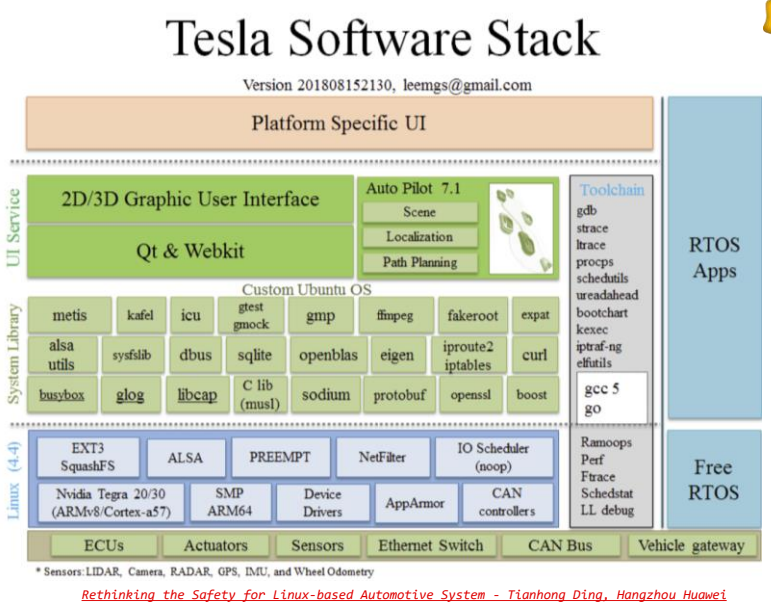


Focus of this presentation would be on **why** and **how** we need to develop a safety test framework in U-Boot, not implementation level details which may be trivial

Using Linux in Safety-Critical Systems (Automotive systems)

Linux in Automotive

- Advantages that make adoption tempting:
  - Heterogeneous computing
  - Rich HW platform support
  - Cheap development cost
  - Security capabilities
  - And everything else that comes with it being open source!



Rethinking the Safety for Linux-based Automotive System - Tianhong Ding, Hangzhou Huawei

Tesla makes use of the Linux ecosystem

Linux in Automotive

- Excuses for not picking up Linux?
  - RT
  - Monolithic
  - Insufficient testing
  - Safety-compliance

Linux in Automotive

- Excuses for not picking up Linux?
  - ~~RT~~
  - Monolithic
  - Insufficient testing
  - Safety-compliance

ZDNET

20 years later, real-time Linux makes it to the kernel - really

After 20 years, Real-Time Linux (PREEMPT_RT) is finally -- finally -- in the mainline kernel. Linus Torvalds blessed the code while he was at Open Source...

18 Sept 2024

Linux in Automotive

- Excuses for not picking up Linux?
  - ~~RT~~
  - ~~Monolithic~~ (still monolithic but we have ways around it)
  - Insufficient testing
  - Safety-compliance

ZDNET

20 years later, real-time Linux makes it to the kernel - really

After 20 years, Real-Time Linux (PREEMPT_RT) is finally -- finally -- in the mainline kernel. Linus Torvalds blessed the code while he was at Open Source...

18 Sept 2024

Xen Project Wiki
https://wiki.xenproject.org › wiki › Xen_Project_Softwar...

Xen Project Software Overview

13 Feb 2024 — **Linux distributions that are based on Linux kernels newer than Linux 3.0** are Xen Project-enabled and usually include packages that contain the ...

Linux in Automotive

- Excuses for not picking up Linux?
  - ~~RT~~
  - ~~Monolithic~~ (still monolithic but we have ways around it)
  - ~~Insufficient testing~~
  - Safety-compliance

ZDNET

20 years later, real-time Linux makes it to the kernel - really

After 20 years, Real-Time Linux (PREEMPT_RT) is finally -- finally -- in the mainline kernel. Linus Torvalds blessed the code while he was at Open Source...

18 Sept 2024

Xen Project Wiki
https://wiki.xenproject.org › wiki › Xen_Project_Softwar...

Xen Project Software Overview

13 Feb 2024 — **Linux distributions that are based on Linux kernels newer than Linux 3.0** are Xen Project-enabled and usually include packages that contain the ...

KernelCI

Linux in Automotive

- Excuses for not picking up Linux?
  - ~~RT~~
  - ~~Monolithic~~ (still monolithic but we have ways around it)
  - ~~Insufficient testing~~
  - **Safety-compliance?**

ZDNET

20 years later, real-time Linux makes it to the kernel - really

After 20 years, Real-Time Linux (PREEMPT_RT) is finally -- finally -- in the mainline kernel. Linus Torvalds blessed the code while he was at Open Source...

18 Sept 2024

Xen Project Wiki
https://wiki.xenproject.org › wiki › Xen_Project_Softwar...

Xen Project Software Overview

13 Feb 2024 — **Linux distributions that are based on Linux kernels newer than Linux 3.0** are Xen Project-enabled and usually include packages that contain the ...

KernelCI

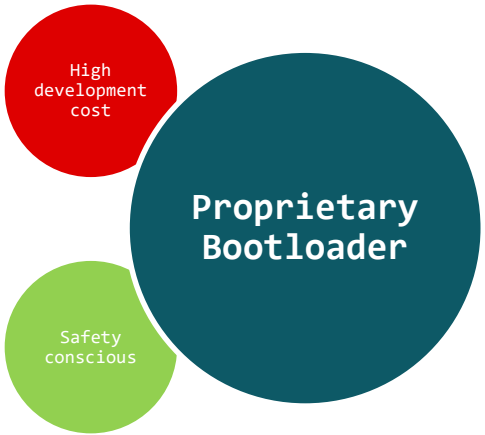
Strides in safety compliance by Linux

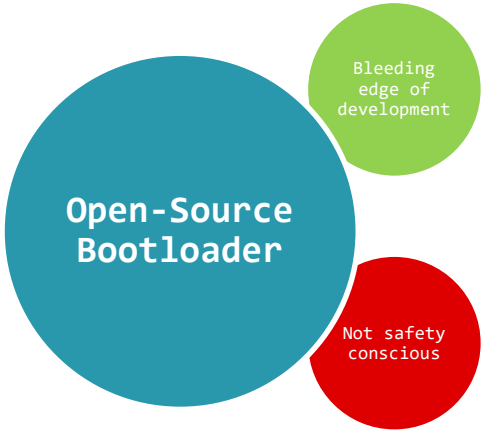
- OSADL SIL2LinuxMP Project
  - Methodology to qualify Linux on a multi-core embedded platform at SIL2
  - Open
- ELISA
  - Defining and maintaining processes that can safety certify Linux-based systems
  - Open
- Automotive Grade Linux
  - Actively developing a Linux-based, open software platform for automotive applications
  - Open
- RHIVOS
  - Targeting functional safety certification prioritizing automotive in Linux
  - AGL
  - Partially open
- OpenAMP
  - Making strides in standardizing heterogeneous communication
  - Not safety specific, but helps

Where are we with bootloaders?

Open Source Bootloaders

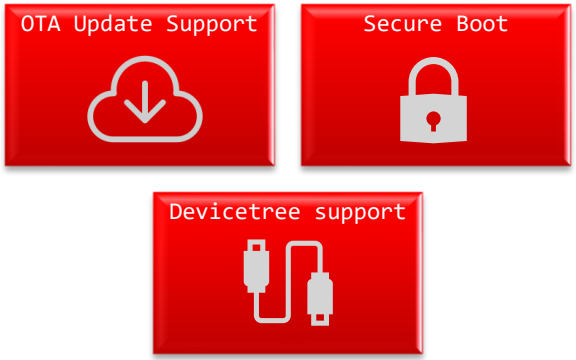
- Not much advancement in safety certification
- Onus on the SoC vendor to safety certify their proprietary bootloaders
- Usually the flow picked is: Proprietary bootloader → Linux





U-Boot in Automotive

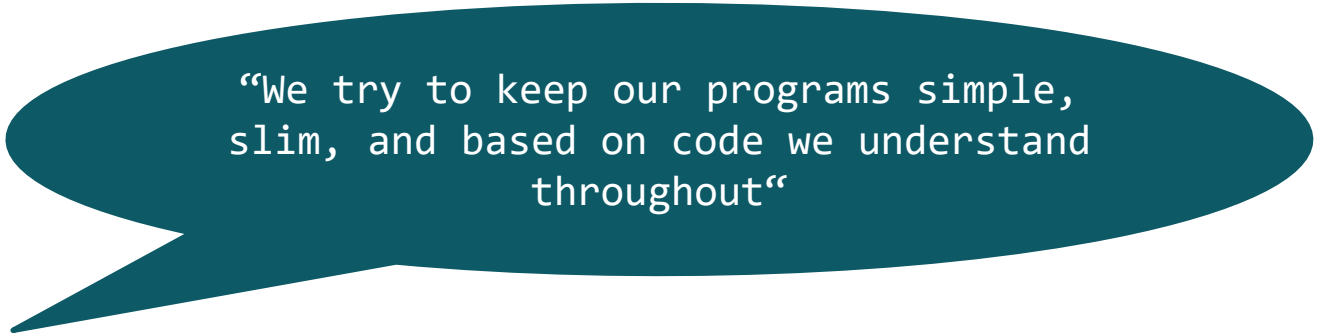
- U-Boot is quite rich
  - Peripheral drivers and stacks
  - Integration with the Linux kernel
  - Multiple arch support: ARM, x86, RISC-v etc.
- Opportunities within automotive for using U-Boot



It’s not like U-Boot hasn’t been leveraged anywhere …

- In fact, it’s more than we think

Project	Bootloader Used
EB Corbos	
RHIVOS	
SpaceX’s Falcon 9	
AGL Images	



Dietrick - SpaceX

Writing simpler, robust code is the underlying principle of U-Boot and all open-source software; and this inherently eases the safety risk

Proprietary bootloaders

- For valid reason, proprietary bootloaders do pretty well for automotive use-cases

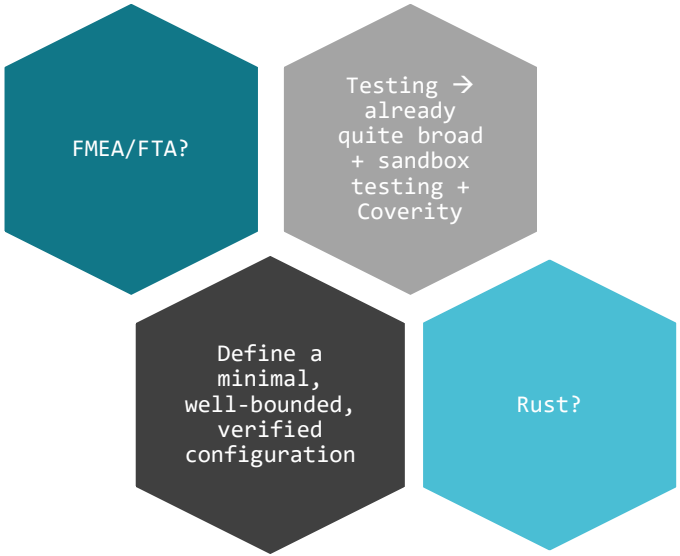
Project	Bootloader Used
Embidetel	Proprietary Flash Bootloader
Vector Informatik	Proprietary Flash Bootloader
Elektrobit	Proprietary ACO Bootloader
ALTEN	Proprietary Auto Bootloader
Texas Instruments	Proprietary SBL

And more …

Safety certifying U-Boot

Is safety certifying U-Boot possible?

- Quite a reach, but not impossible

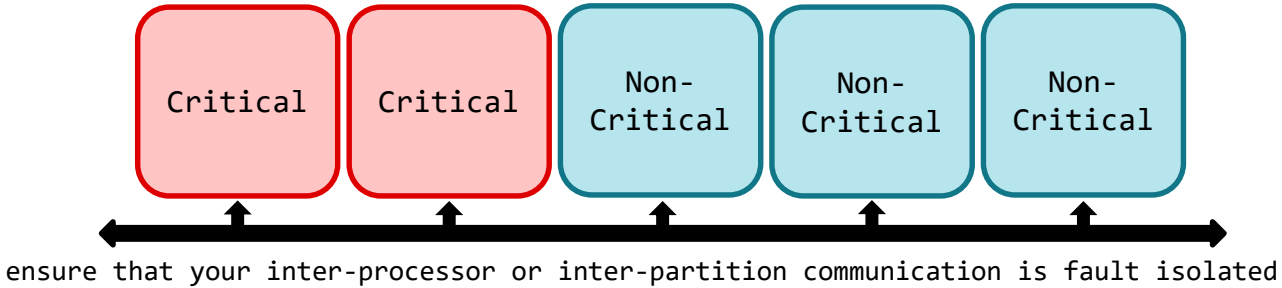


- We have a long way to go… what can we aim at now?

Mixed-Criticality Systems (MCS)

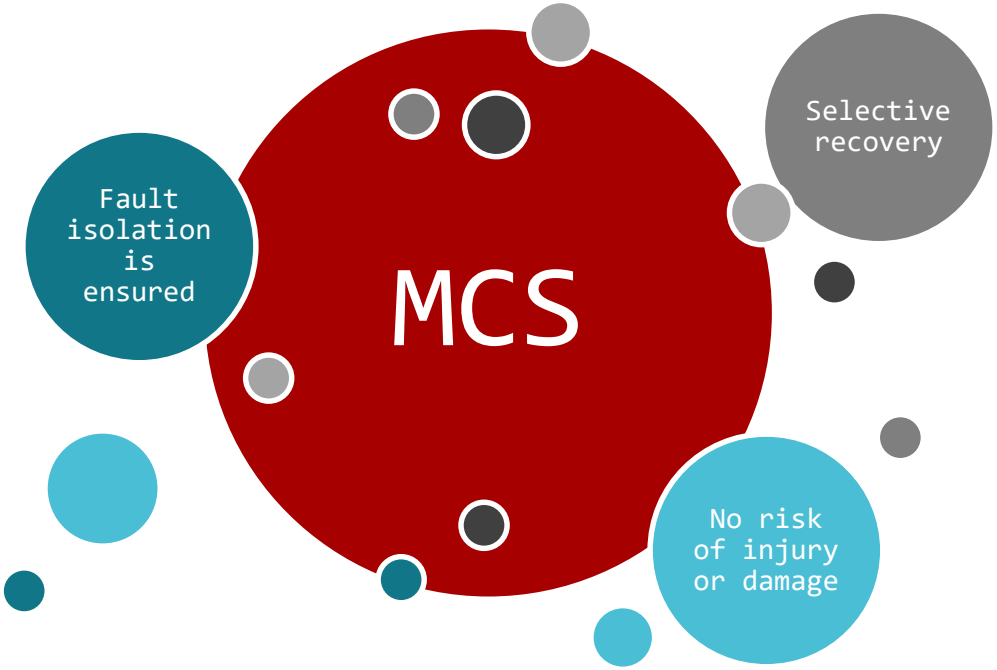
Mixed-criticality use-cases of safety

- MCS allows varying levels of safety requirements to coexist on the same hardware platform



- e.g. IVI systems require VirtIO and PV devices
  - higher complexity + lower safety levels

Mixed-criticality use-cases of safety



Red Hat Reaches Key Milestone in Push to Functional Safety Certification for Red Hat In-Vehicle Operating System

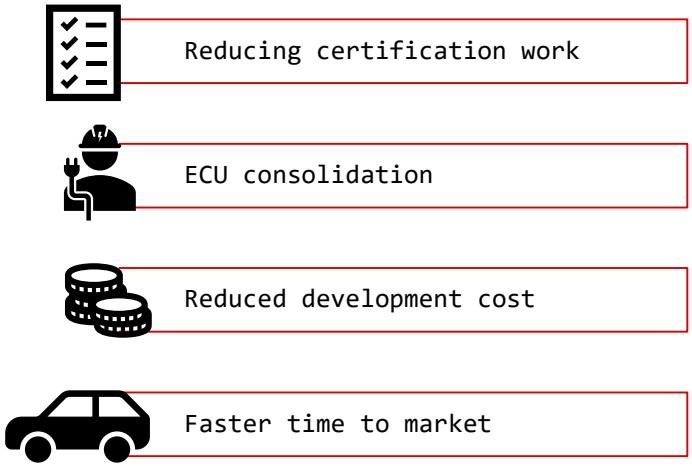
BlackBerry
Why Mixed-Criticality Is the Future of Automotive Architectures
AUTOMOTIVE & IOT / 09.14.22 / Randy Martin

wipro
Three Steps to Implementing Mixed Criticality...
Implement Mixed Criticality Systems for Automotive Safety

SYSGO
EMBEDDING INNOVATIONS
Tackling Mixed-Criticality for Automotive
Jun 14, 2021 – PikeOS, Automotive, Safety

MCS

- A lot of SDVs are adopting this already:



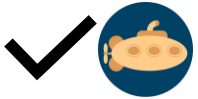
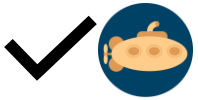
- Linux is a clear choice for enabling a part of such systems
- But why should U-Boot be the choice as a bootloader?

Gearing up a bootloader to work in an MCS

Gearing up a bootloader to work in an MCS

- Besides the obvious, i.e. initializing the hardware
- In an automotive system, the bootloader needs to also
  + Support OTA updates
  + Support secure boot
  + Support **diagnostic testing**

Gearing up a bootloader to work in an MCS

- Besides the obvious, i.e. initializing the hardware
- In an automotive system, the bootloader needs to also
  - + Support OTA updates 
  - + Support secure boot 
  - + Support **diagnostic testing**

U-Boot in MCS

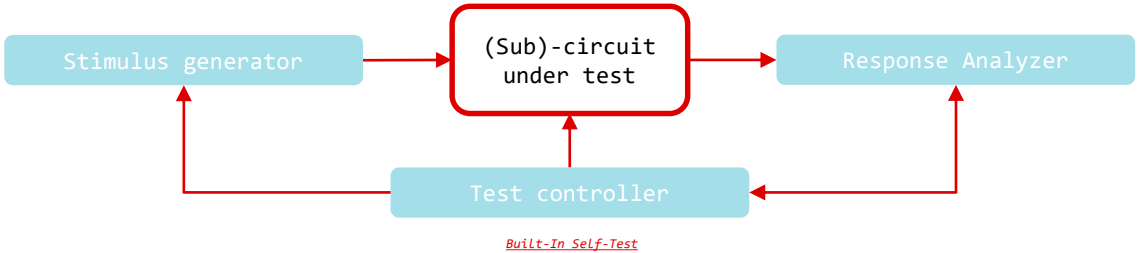
U-Boot in MCS?

- Increase the safety chapter of U-Boot → double edged sword
- In MCS, the bootloader can execute fault diagnostic tests prior to running safe applications
  - Enable ECC on memory if any

  - Check for CPU faults if any → BIST (Built In Self Test)

Using U-Boot in MCS

- Safety IP: BIST (Built-In Self Test)
  - Safety diagnostic block embedded in hardware that test memory areas and logic circuitry in an IC at boot time

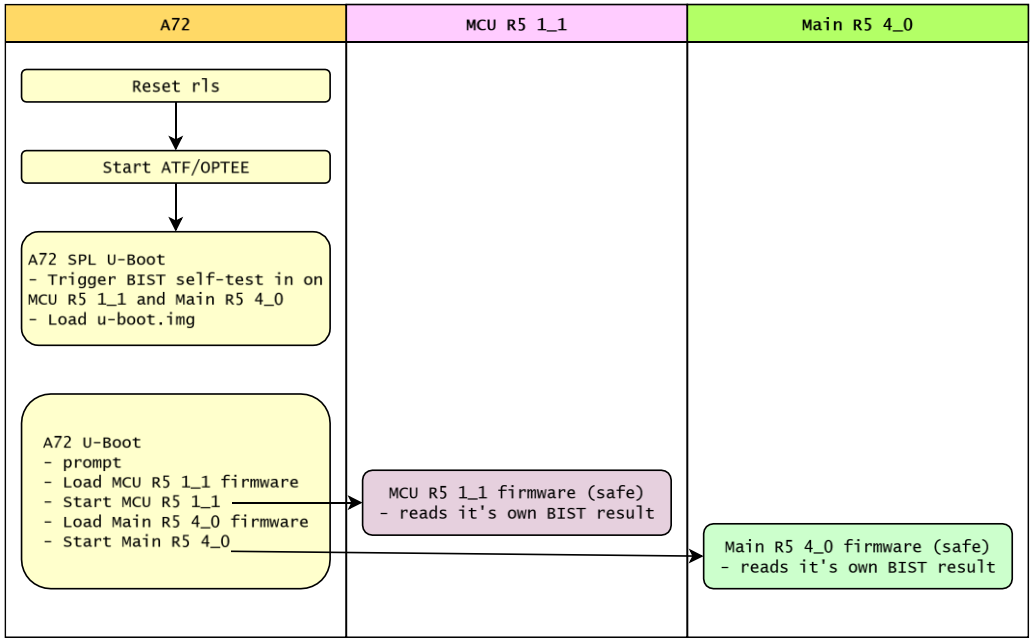


- Enhances system reliability
- Expected to be run once at boot time
- Destructive in nature, cannot be performed by applications at runtime without saving/restoring context

Using U-Boot in MCS: an example

- Customer Request: trigger BIST from non-safe U-Boot on the safety channels running safety certified firmware
- Upstreamed the BIST driver under misc/ in U-Boot
- Link to the patch series

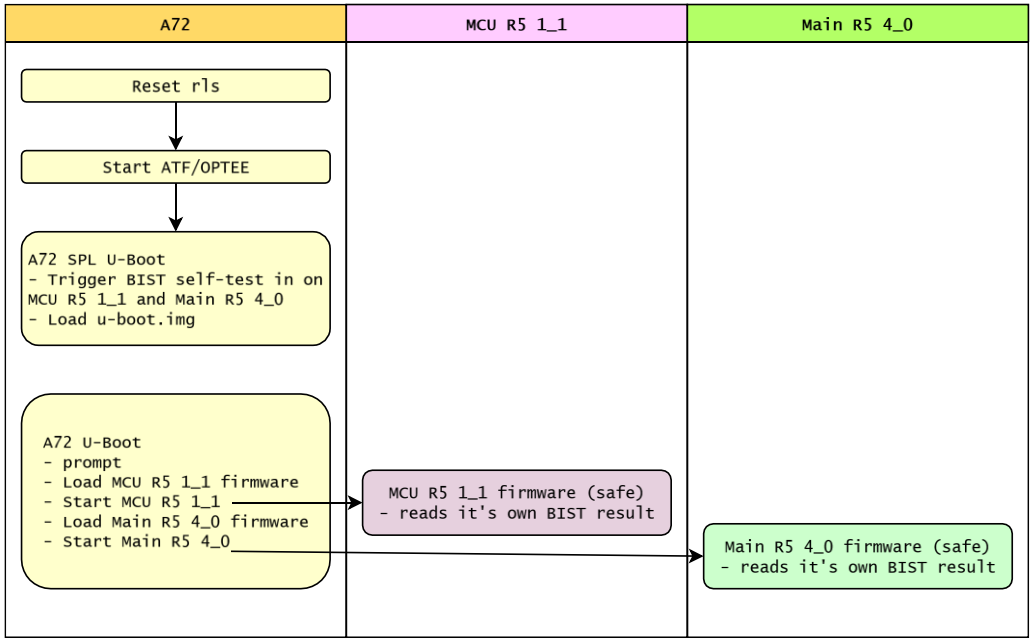
Using U-Boot in MCS: an example



Flow adopted by customer with System Level CAT3 certification using the BIST driver for a mixed-criticality safety use-case

Using U-Boot in MCS: an example

```
bist_main14: bist@33c0000 {
        compatible = "ti,j784s4-bist”;
        reg = <0x00 0x033c0000 0x00 0x400>,
              <0x00 0x0010c1a0 0x00 0x01c>;
        reg-names = "cfg", "ctrl_mmr";
        clocks = <&k3_clks 237 7>;
        power-domains = <&k3_pds 237 TI_SCI_PD_EXCLUSIVE>;
        bootph-pre-ram;
        ti,sci-dev-id = <234>;
};
```



Flow adopted by customer with System Level CAT3 certification using the BIST driver for a mixed-criticality safety use-case

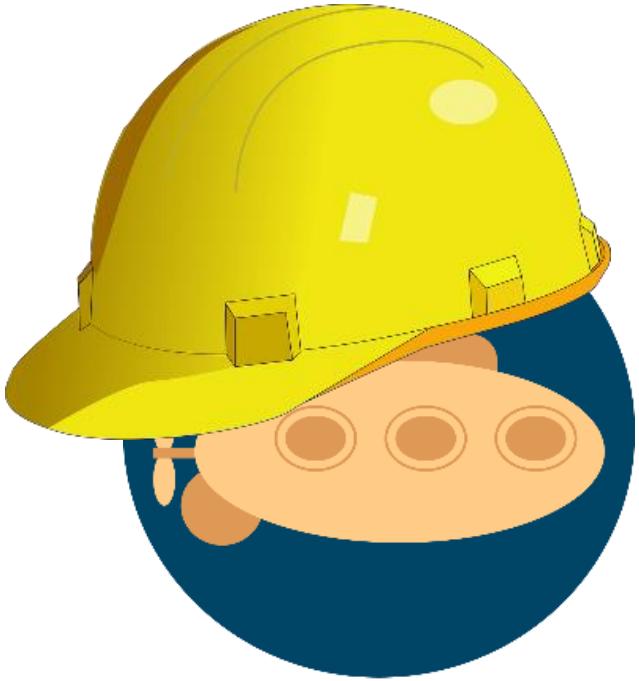
Safety IPs in U-Boot

- misc/ good enough place to house BIST driver?
- However possibilities of other safety IPs; such as JTAG, SDL, STL etc. can come up
  - JTAG: not just for debug/programmability → boundary scan tests
  - SDL/STL: ARM libraries
    - Runtime (non-destructive)
    - Boot time (includes destructive)
- Move from misc/ → safety-test/ or safety/

Conclusion

- Large scope for introducing safety IPs within U-Boot → safety-test framework
- The BIST driver in what is hopefully a series of more such drivers that can increase the use of U-Boot in mixed-criticality automotive use-cases

All part of an effort to ensure U-Boot remains the bootloader of choice in Automotive Systems!



References

- Linux in Safety Critical Systems - Kate Stewart
- Rethinking the Safety for Linux-based Automotive System - Tianhong Ding, Hangzhou Huawei
- From Space to Scrum – 3back
- TI Processor SDK Linux J784S4 - PBIST Guide
- TI Processor SDK Linux J784S4 - LBIST Guide

Credits and Acknowledgement

- Texas Instruments Inc.
- The Linux Foundation
- Kevin Peng, Texas Instruments Inc.
- Beleswar Padhi, Texas Instruments Inc.
- SDL Team, Texas Instruments Inc.
- U-Boot community

Q&A

- Contact Information:
  Neha Malcom Francis
  <n-francis@ti.com>
- Also on IRC @ libera.chat
  #linux-ti

Learn more about TI products

- https://www.ti.com/linux
- https://www.ti.com/processors
- https://www.ti.com/edgeai

Why choose TI MCUs and processors?

Scalability

Our products offer scalable performance that can adapt and grow as the needs of your customers evolve.

Efficiency

We design products that extend battery life, maximize performance for every watt expended, and unlock the highest levels of system efficiency.

Affordability

We strive to make innovation accessible to all by creating cost-effective products that feature state-of-the-art technology and package designs.

Availability

Our investment in internal manufacturing capacity provides greater assurance of supply, supporting your growth for decades to come.