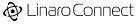


U-Boot for SystemReady-IR

Status and struggles

SystemReady-IR

- A band of the <u>SystemReady</u> certification program
- Tailored for embedded devices
- Based on
 - Embedded Base Boot Requirements (EBBR)
 - EBBR recipe of the Arm Base Boot Requirements (BBR) specification
 - Device Tree specification
- Defines a minimum set of hardware and firmware features and interfaces that are needed to deploy OS images in a standard way

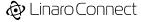


U-Boot UEFI pre-2019

• Basic UEFI support was added ~2016

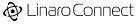
In 2019 U-Boot supported

- UEFI variables stored as environmental variables
- UEFI Boot Manager was just merged with only basic features
- Missing functionality
 - UEFI secure boot
 - UEFI/TCG2 measured boot
 - UEFI HTTP(s) Boot support
 - Capsule updates
 - Other useful protocols e.g EFI_RNG
 - A complete EFI BootMgr



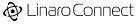
Feature history (1/2)

- UEFI variables non-volatile storage
 - In a Replay protected memory block (RPMB) May 2020
 - In a file <u>April 2020</u>
- UEFI Secure Boot <u>April 2020</u>
- UEFI/TCG2 Measured Boot added <u>November 2020</u>
- Capsule update support <u>November 2020</u>
 - Capsule authentication was added a bit later
- SetVariable at Runtime
 - Variables stored in <u>file</u> Authenticated variables not supported
 - Variables stored in <u>RPMB</u>



Feature history (2/2)

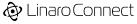
- Multi bank firmware updates <u>October 2022</u>
 - Brick/rollback protection for capsule updates
- UEFI BootManager enhancements <u>March 2021</u>
- UEFI HTTP Boot (HTTPs is WIP) <u>November 2023</u>
- UEFI RNG protocol <u>December 2019</u>
- And many more ...



SystemReady-IR requirements

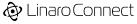
- A UEFI-compliant bootloader using DeviceTree
- Perform UEFI sanity tests in the EFI Shell
- Perform capsule update manually
- Run the automated Architecture Compliance Suite (for IR)
 - \circ ~ Tests all the UEFI protocols mandated by the certification
- Install three Linux distributions and perform OS tests manually
- Optionally test for Security Interface Extension
 - \circ ~ Tests for UEFI Secure and measured Boot
- Create logs including the information above
- Verify your test results using Arm provided scripts

A complete guide can be found <u>here</u>



U-Boot for SystemReady compliance

- The first (stable) compliant version of U-Boot was released in 2021
 - 2021.04 <u>release</u>
- Arm keeps a list of officially <u>certified</u> boards
- The majority of the certification was done against SystemReady-IR 1.x
- This is now deprecated and 2.x is required. There are some extra requirements compared to 1.x
 - Authenticated capsule updates
 - Device Tree conformance
 - Block device boot source checking
 - Ethernet checks in Linux
 - Optional Security Interfaces Extension for UEFI Secure boot and TPM support



Certification – Common Issues

Certification

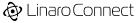
- Capsule updates Code missing from firmware
 - \circ ~ Each board must define its own UUIDs and updatable partitions
 - dfu_alt_info not configured, breaking capsule updates
- Device tree validation Using downstream DTs
 - \circ ~ Devicetree nodes not described by a YAML schema
 - CONFIG_OF_UPSTREAM helps?
- Distro installers Board specific Kconfig options missing from the arm64 distro kernel config
 - Contact distros and ask for board support before starting the certification?



Certification – Common Issues

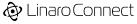
Certification

- Reset from U-Boot doesn't work properly
- Missing ESP Leads to broken UEFI variables persistence
- bootargs set on UEFI boot path SHIM doesn't work properly
- U-Boot environment stored in OS medium, necessitating protective partition
- U-Boot compiled for UEFI variables in OP-TEE/StMM, but no StMM, RPMB, OP-TEE support
- USB issues in U-Boot: instabilities preventing (long) ACS runs,
 - flaky mass storage enumeration
 - USB 3.0 not supported...



SystemReady-IR adoption

- U-Boot is stable for quite some time regarding EFI (famous last words...)
- Is there adoption from OEMs/ODMs? If not, any feedback why?
 - Is it because they found the transition from traditional booting to UEFI hard?
 - Does UEFI break any of the existing customer use cases?
 - Are there missing features? What are those?
- Thoughts from SIPs including instructions for SystemReady-IR compliant builds on the BSPs?
- Any known products built on top of SystemReady?



Further discussion

For further discussions join us <u>MAD24-319</u> <u>Findings from the last SystemReady IoT workshop:</u> <u>What a reference stack for Rich IoT should look like</u>

Thursday, 16 May 13:45 - 14:10 Room: Session 1 | Las Palmas I





Thank you