

Does System Ready IR "Just Work"?

Jon Humphreys

Texas Instruments



Author

Jon Humphreys, *Eng Manager – Open Source Product Technology*
Texas Instruments

Jon has been working with TI for over 30 years and is currently part of TI's Embedded Processing organization leading our Open Source Product Technology team. He spent the first ~20 years at TI working on embedded and DSP compiler technology and development tools. Before OSPT, he managed the embedded Linux teams delivering upstream and SDK support for TI's embedded SoCs.



About us: TI Processors and Open source



Decades of contribution and collaboration

Ingrained culture to give back to the community



Upstream FIRST!



Focus on long term, sustainable and quality products



Upstream and opensource ecosystem in device architecture

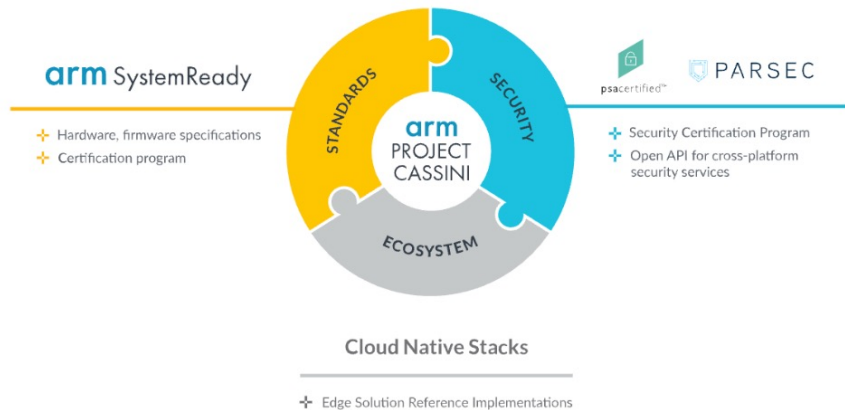


Open
Source

Upstream FIRST mentality!



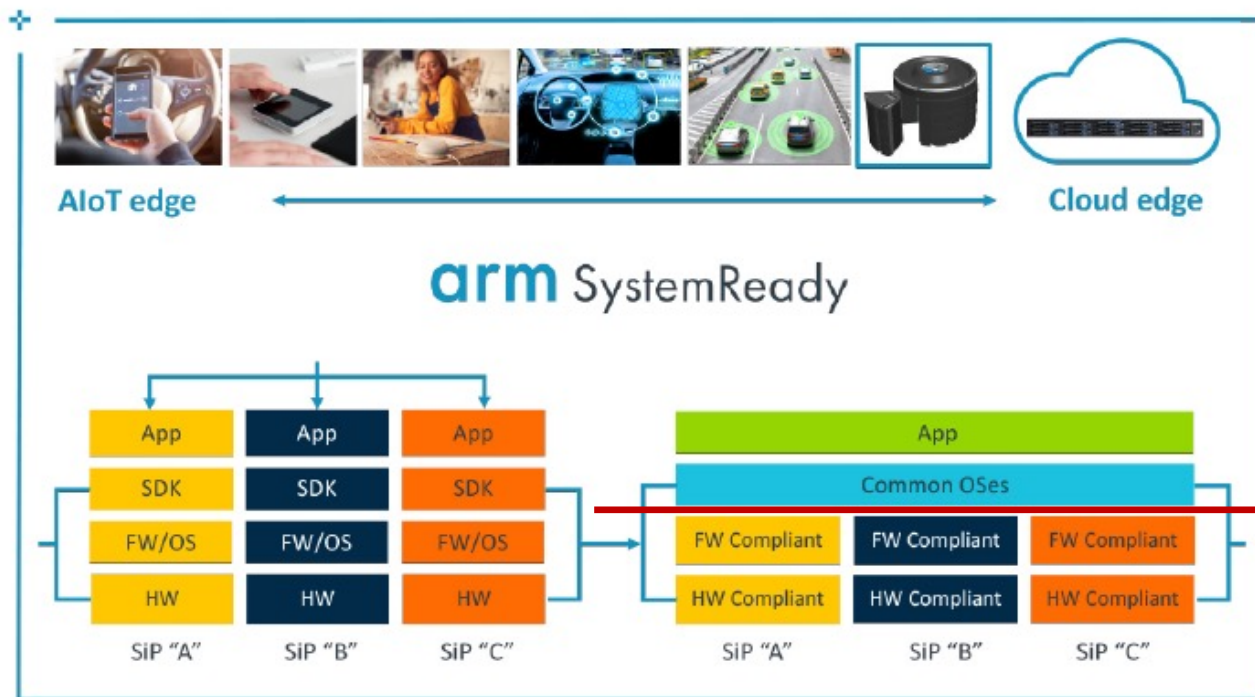
Project Cassini



"just works"





- address fragmentation
 - consistent deployment model for end application
 - simpler to scale solutions across a diverse set of devices - that just works
- provide compute scalability
- allows migration of workloads from cloud to the edge

System Ready



System Ready Bands

arm SystemReady

	LS	IR	ES	SR
				
Firmware Spec	ACPI + SMBIOS	UEFI + <u>Devicetree</u>	UEFI + ACPI + SMBIOS	UEFI + ACPI + SMBIOS
Platform Hardware	64bit Arm	32bit/64bit Arm	64bit Arm	64bit Arm
	Can support UEFI <u>SecureBoot</u> and Secure Firmware Update via UEFI Capsule Service across (BBSR)			
OS/Hypervisor	Linux	Linux, etc.	Generic, off-the-shelf w/ exceptions: RAS, I/O virtualization, etc.	Generic, off-the-shelf
OS Distro (examples)	Linux	Fedora, openSUSE, SLES, Debian, Ubuntu, <u>Yocto</u> Under investigation: <u>OpenWRT</u> , QNX, VxWorks, Integrity, Wind River, Mentor	Windows IoT Enterprise, VMware <u>ESXi</u> , RHEL, SLES, Ubuntu, CentOS, Fedora, openSUSE, Debian, CBL-Mariner, FreeBSD, NetBSD, OpenBSD	VMware <u>ESXi</u> , Windows Client/Server, RHEL, SLES, Ubuntu, CentOS, Fedora, openSUSE, Debian, CBL-Mariner, FreeBSD, NetBSD, OpenBSD
Hardware Compliance Levels	BSA+SBSA Levels 3 through 6	BSA + No BSA requirements for 32-bit + BSA test reporting only initially	BSA + waivers for existing HW initially	BSA+SBSA Levels 3 through 6
BBR Recipe	LBBR	EBBR	SBBR	SBBR
Certification	Arm SystemReady LS + System Compatibility List	Arm SystemReady IR + System Certification List	Arm SystemReady ES + System Certification List	Arm SystemReady SR + System Certification List

System Ready IR Standards

Base System Architecture (BSA)	hardware requirements for an operating system to boot successfully
Base Boot Requirements (BBR)	firmware interface requirements
Embedded Base Boot Requirements (EBBR)	market segment-specific BSA supplement
Base Boot Security Requirements (BBSR)	Security

“Just Works” Definition

Just Works is an ecosystem of components that can be installed and correctly function together out of the box, without:

- customization or modification of the software
- expert knowledge
- device specific solutions

The result is:

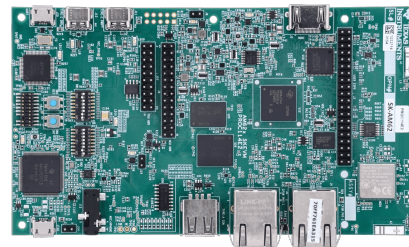
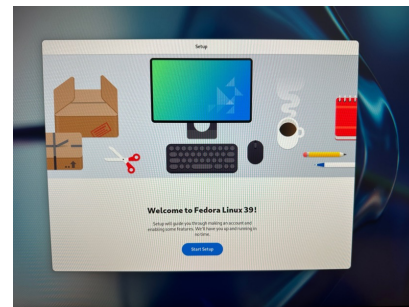
- components have a high degree of interoperability and compatibility
- The experience is ‘easy’



Focus of “just works” is on system software (firmware + OS)

The “simple test”

- Boot a generic image of a standard distro
 - Without modifications
 - Without being an expert of the device, distro, or embedded linux, EFI, DTs, ...
- All boards used are either claimed to by System Ready IR certified or known by me to be compliant (in case of the TI based boards)
 - I relied upon the instructions in the certification errata
- For results, all board references are anonymous



Findings that spoil “just works”

Boards are certified against specific firmware versions

- Being paranoid, I want to use that specific version
 - Good luck finding `UBoot 2021.04+g<abcxyz> ...`
 - Or the firmware used is not publicly available
 - (some certification reports include a URL, which is nice!)
- But it would be best if I could just get the ‘latest’
 - Tried a few but without luck

Recommendations:

- Development boards are shipped with SystemReadyIR compliant f/w
- f/w updates are via SystemReadyIR standards (eg, EFI capsules)
- Certification requires using original f/w shipped with the board, or has only been updated via SystemReadyIR standards

Findings that spoil "just works"

Certification Errata

- Most boards certified have an Errata Document
- Some are simple, some not, none are "just works"
- Usually involves:
 - Flashing firmware
 - Formatting/Partitioning storage
 - Creating an ESP partition
 - OS installer workarounds
- Often links are broken

Recommendations

- Preinstalled firmware (see prior recommendation)
- Preformatted on-board storage
- ESP creation should be handled by the OS installer
- Certification archives software artifacts

Goal is NO errata!

Findings that spoil "just works"

OS provided DT do not get used

- EBBR requires DT to be provided by the firmware [0]
- DT evolves and kernel builds include corresponding DTB
- System Ready IR booted system almost guaranteed out-of-date DT

Options

- Do nothing, and hope that if you wait long enough, DT will stabilize and it won't matter
- OS provides capsule update with latest DTB
- Use UKI (Unified Kernel Image) images
- Standardize on location f/w would look for a DTB
- OS loader chooses DTB
- OS loader can choose more recent DTB

[0] <https://arm-software.github.io/ebbr/> Sec 1.2

Summary of Recommendations

System Ready IR, and "just works", are noble goals worth pursuing

Recommendations

- ☐ Certification requires using original f/w shipped with the board, or has only been updated via SystemReadyIR standards
- ☐ Adoption of non-certification, continuous testing services such as Linaro's ONElab
- ☐ Certification errata artifacts are archived
- ☐ Vendor's reduce certification errata as much as possible
- ☐ Standardize method of OS provided DTs

Q&A

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.



Linaro Connect
MADRID 2024 | MAY 12-17 2024

Thank you



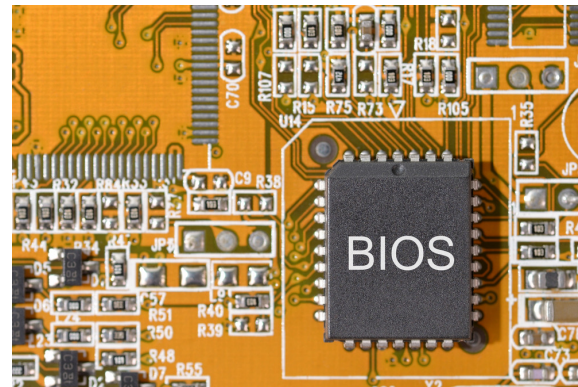
“Just Works” Implications

If system software cannot be device specific as a solution, but inherently has device specific components, it must be divided into firmware and OS

- this is what BBR/EBBR aims to define and delivered independently
- Otherwise we lose the ecosystem of interoperability

Thus there are 2 steps with installation:

- Flash firmware (device specific)
- Flash OS image or run OS installer (device agnostic)



“Just Works” Firmware

Some aspects of the firmware are standardized

- Clearly the firmware / OS interface
- Also other aspects important to “just works”
 - Firmware storage layout to prevent firmware / OS from corrupting each other

But many aspects are not standardized

- by necessity, they relate to the specifics of the board

This lack of standardization can be a big spoiler for “just works”

Findings that spoil “just works”

OS installer missing drivers

- Dreaded “No device for installation media was detected” ...
- OS installer uses a minimal kernel that may not have the driver needed to access the OS installer media

Recommendation

- Work with major OS vendors to ensure minimum storage drivers included in their installer kernel
- Board vendors provide loadable driver modules

Findings that spoil “just works”

Corrupted Firmware

- Firmware updates rely upon an existing working firmware
- If the current firmware isn't functional (or just can't process capsules), replacing it with a working firmware requires non-standard steps

Recommendation

- Standard A/B update mechanisms or ostree type solns exist
- SystemReady IR certification should require this