

Beyond Android

Minimalistic Yocto/Linux Image for the LTSCT SAC20



As connected products evolve; system designers are increasingly rethinking the role of the operating system. While Android offers a rich ecosystem and rapid application development, it is not always the optimal choice for embedded and IoT deployments where deterministic behaviour, low power consumption, faster boot times, and long-term maintainability are critical.

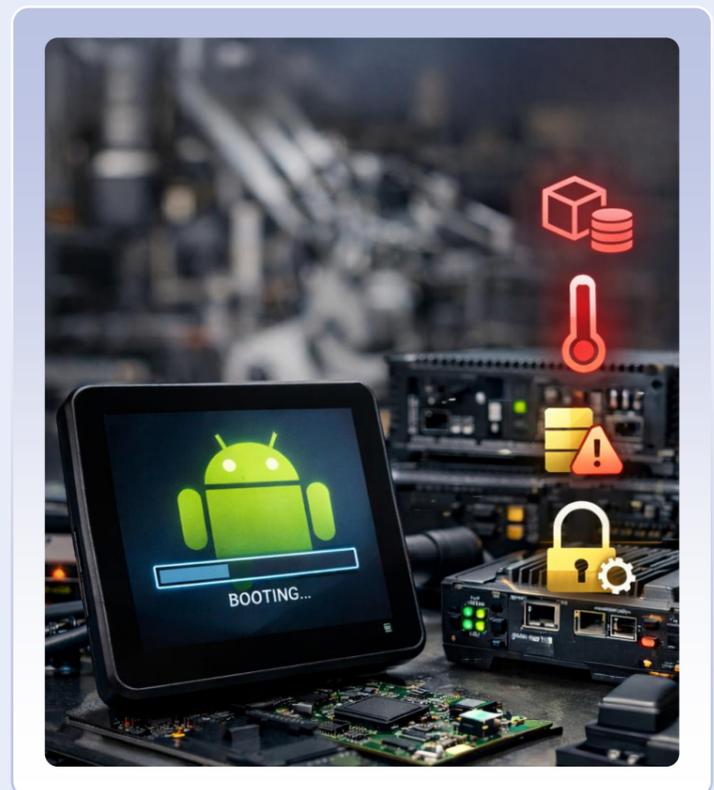
For such use cases, a minimalistic Yocto/Linux image on the LTSCT SAC20 Smart Module provides a compelling alternative delivering precisely what is required, and nothing more.

Why Look Beyond Android?

Android is well-suited for user-centric devices with complex graphical interfaces and application-layer flexibility. However, in many industrial and commercial deployments, it introduces overhead that directly impacts system efficiency.

Common challenges with Android-based systems include:

- Higher memory and storage footprint
- Longer boot times
- Increased power consumption
- Limited control over kernel and system services
- Complexity in long-term maintenance and customization
- For headless devices, industrial controllers, gateways, telematics units, kiosks, and asset trackers, these trade-offs often outweigh the benefits



The Case for Yocto-Based Linux on SAC20

Yocto is not a Linux distribution it is a build framework that allows developers to create highly customized Linux images tailored to a specific hardware platform and application.

On the **LTSCT SAC20**, Yocto enables:



A lean, purpose-built OS image



Full control over kernel configuration and system services



Optimized power, memory, and storage usage



Long-term stability and reproducibility

yocto.
PROJECT



This approach aligns perfectly with SAC20's modular hardware and low-power architecture.

1. Reduced Footprint



A Yocto-based image can be stripped down to only essential components:

- Custom kernel with required drivers only
- Lightweight unit systems
- Minimal root filesystem

Result:

Lower RAM and flash usage, enabling cost-optimized hardware designs.

2. Faster Boot Time



By eliminating unnecessary services and background processes, a minimal Linux image significantly reduces boot latency.

Ideal for:

- Automotive and telematics systems
- Industrial equipment with power cycling
- Devices requiring rapid readiness after wake-up



3. Improved Power Efficiency

With fewer daemons, optimized kernel settings, and tighter control over peripherals, Yocto/Linux complements SAC20's low-power states.

Key advantages:

- Better idle and sleep power behaviour
- Deterministic wake-up paths
- Application-driven power management



4. Deterministic and Reliable Operation

Unlike Android's complex service stack, a minimal Linux system offers:

- Predictable scheduling
- Easier debugging and profiling
- Greater system transparency

This is particularly valuable in mission-critical and industrial deployments.

Developer Control and Customization

Yocto provides fine-grained control across the entire software stack:

- Kernel configuration and patches
- Bootloader customization
- Device tree optimization
- Application and service inclusion
- Complete control over packages, init system, and services
- Easier Licensing and Compliance
- Faster BSP Bring-up & Debugging

For SAC20 developers, this means:

- Faster bring-up for custom products
- Easier integration of proprietary middleware
- Simplified certification and compliance workflows



Typical Use Cases for Yocto/Linux on SAC20

A minimal Yocto/Linux image on SAC20 is well suited for:

- ✓ Industrial IoT gateways
- ✓ Point of Sale devices
- ✓ Asset tracking and logistics systems
- ✓ Kiosks and Vending machines

In these scenarios, stability, efficiency, and longevity matter more than consumer-grade UI richness.

Android vs. Minimal Yocto/Linux: A Strategic Choice

| Aspect | Android | Minimal Yocto/Linux |
|-----------------------|-----------------|-------------------------------|
| Footprint | High | Very low |
| Boot Time | Longer | Fast |
| Power Efficiency | Moderate | High |
| Customization | Limited | Extensive |
| Long-Term Maintenance | Complex | Predictable |
| Ideal Use Case | UI-rich devices | Embedded & industrial systems |

ANDROID

MINIMAL YOCTO/LINUX

BEYOND ANDROID

Large Footprint Slow Boot

Small Footprint Fast Boot
Low Power

The choice is not about replacing Android universally, but about selecting the right OS for the right product.

What This Means for SAC20-Based Products

For Customers:

A Yocto/Linux approach reduces BOM pressure, extends battery life, and lowers long-term operational costs.

For Developers:

It delivers a clean, controllable, and scalable software foundation free from unnecessary abstractions and overhead.

Conclusion

Going beyond Android with a minimalistic Yocto/Linux image on the LTSCT SAC20 empowers to build efficient, reliable, and purpose-driven connected products. By aligning the operating system with the application's real requirements, SAC20-based solutions can achieve faster boot times, lower power consumption, and greater system longevity without compromising on performance or connectivity.

In modern embedded design, **simplicity is not a limitation; it is a strategic advantage.**

Future. Made Together.

Ready to accelerate your connectivity roadmap?

Partner with LTSCT Modules to accelerate certification, reduce cost and launch globally.

Book a technical discussion /
sample evaluation request:

Email: jrn@ltsct.com | asha.kulkarni@ltsct.com

Follow us on



www.ltsct.com