

Displays Smart e Android em Ambientes Industriais:

Limitações Críticas e a Superioridade do Linux Embarcado

Revisão técnica 2025 | Baseado em experiência de campo com soluções de display industrial
Referências: IEC 62443, Android Enterprise Recommended, Yocto Project, MISRA, IPC-7711, Linux Foundation Embedded

1. Introdução

A demanda por interfaces gráficas em equipamentos industriais, máquinas agrícolas, sistemas de automação e IoT embarcado cresceu de forma expressiva ao longo da última década. Com isso, o mercado de displays TFT industriais se fragmentou em três grandes arquiteturas: displays com controladores simples (sem sistema operacional de propósito geral), displays chamados "Smart" baseados em SoCs proprietários com firmware fechado, e displays baseados em Android ou Linux embarcado com sistema operacional completo.

A aparente facilidade de desenvolvimento que os displays Smart e Android oferecem torna-os atraentes na fase de prototipagem. Porém, em produção industrial — especialmente em aplicações com ciclo de vida longo, requisitos de confiabilidade, conformidade regulatória e suporte de campo — essas plataformas apresentam limitações estruturais que comprometem projetos inteiros.

Este artigo documenta as limitações técnicas, comerciais e operacionais dos displays Smart e Android em aplicações industriais, e apresenta os critérios pelos quais soluções baseadas em Linux embarcado (Yocto, Buildroot, Debian/Armbian customizados) oferecem vantagens determinantes para engenheiros e arquitetos de produto.

2. Arquiteturas de Display Industrial: Classificação

Para contextualizar a análise, é necessário distinguir as três arquiteturas dominantes no mercado:

Arquitetura	Descrição	Plataformas Típicas	Uso Típico
Display Simples (MCU)	Controlador MCU dedicado (STM32, RA, PIC32). Sem SO. Firmware próprio.	LVGL, Segger emWin, TouchGFX	IHMs simples, painéis de máquina, indicadores
Display Smart / Proprietário	SoC dedicado com firmware fechado do fabricante. Interface via comandos seriais ou UART.	Nextion, STONE, DWIN (DGUS), TDO, EVE/BT817 (Bridgetek)	Prototipagem rápida, HMI de baixo custo

Display com Android	SoC (RK3288, RK3399, A40i) executando Android AOSP customizado. GUI nativa Android.	Rockchip, Allwinner, Qualcomm	Quiosques, painéis HMI médios, eletrônica de consumo industrial
Display com Linux Embarcado	SoC executando kernel Linux customizado (Yocto, Buildroot). Acesso total ao hardware.	i.MX8, RK3568, AM64x, RPi CM	Automação industrial, agtech, IoT crítico, sistemas com ciclo de vida longo

As seções a seguir analisam em profundidade as limitações das arquiteturas Smart e Android, que são as mais frequentemente escolhidas de forma equivocada em projetos industriais sérios.

3. Limitações dos Displays Smart (Firmware Fechado)

Displays Smart — como Nextion, STONE TFT, DWIN (série DGUS/T5L) e TDO — são atraentes pela facilidade de integração inicial: editor gráfico proprietário, comunicação UART simples e curva de aprendizado baixa para firmware de MCU host. Cada fabricante possui seu ecossistema próprio com características distintas, mas todos compartilham uma limitação arquitetural comum: o firmware interno é fechado, o hardware não é documentado em nível de registradores, e a lógica de interface é compilada para um SoC proprietário. Essa conveniência inicial encobre restrições sérias para produtos industriais.

3.1 Aprisionamento Tecnológico (Vendor Lock-in)

Toda a lógica de interface gráfica, transições, animações e lógica condicional reside em um arquivo binário compilado para execução no SoC proprietário de cada fabricante: .tft (Nextion), .hmi (STONE), .dgus (DWIN) ou formato equivalente (TDO). Os arquivos de projeto não são interoperáveis entre fabricantes. As consequências diretas são:

- **Impossibilidade de migração:** trocar de fornecedor de display exige reescrever 100% da interface gráfica — não existe padrão portátil.
- **Dependência de ferramentas proprietárias:** Nextion Editor, STONE Designer, DWIN DGUS Tool, TDO Tool — ferramentas fechadas, sem API pública, sem integração com pipelines CI/CD ou controle de versão Git.
- **Sem acesso a código-fonte:** depuração de falhas de renderização é feita por tentativa e erro, sem logs de sistema ou stack trace.
- **Sem suporte a padrões industriais:** OPC-UA, MQTT nativo, protocolos Modbus sobre TCP exigem intermediação obrigatória via MCU host.

⚠ Risco de projeto: Em 2019, a Nextion alterou o protocolo de comunicação serial em uma atualização de firmware sem aviso prévio compatível. Equipamentos em campo com firmware antigo pararam de comunicar com o novo display de reposição. Este é um risco real e documentado de vendor lock-in em plataformas fechadas.

3.2 Capacidade Computacional Limitada

O modelo de execução dos displays Smart é fundamentalmente limitado: toda a lógica de negócio deve residir no microcontrolador host externo. O display executa apenas comandos de desenho e lógica condicional simples. Isso cria uma dependência estrutural e limitações claras:

- Sem suporte a múltiplas tarefas simultâneas (multithreading real) dentro do display

- Sem capacidade de processamento de dados localmente (análise de séries temporais, cálculos de processo, logging)
- Animações complexas e atualizações de alta frequência (>10 Hz) causam artefatos visuais e travamentos
- Memória de aplicação limitada a poucos MB — projetos com muitas telas ou assets gráficos atingem o limite rapidamente

3.3 Confiabilidade e Comportamento em Campo

Em ambientes industriais com variações de temperatura, vibração e transientes elétricos, os displays Smart apresentam comportamentos problemáticos que raramente aparecem em bancada:

- **Corrupção de memória flash interna:** atualizações de firmware interrompidas (falha de energia durante upload) resultam em displays inutilizáveis sem jig de recuperação.
- **Watchdog limitado:** o firmware fechado não expõe mecanismos de watchdog configuráveis. Travamentos do display não são recuperáveis remotamente.
- **Temperatura de operação reduzida:** a maioria dos módulos Smart é classificada para 0°C a 50°C — insuficiente para painéis externos em regiões tropicais ou ambientes de máquinas agrícolas (-10°C a +70°C).
- **Sem logs de sistema:** falhas de campo são não-diagnosticáveis remotamente. A única solução é recolher o equipamento ou enviar técnico.

3.4 Ciclo de Vida do Produto e Disponibilidade — Análise por Fabricante

A avaliação de ciclo de vida dos displays Smart exige diferenciação entre fabricantes. A generalização de que todo o segmento opera com histórico de descontinuação abrupta não é precisa — e ignorar essa distinção leva a decisões de projeto mal calibradas.

Fabricante	Fundação / Histórico	Gestão de EOL	Disponibilidade de modelos	Risco para projetos >7 anos
Nextion (ITEAD)	2014. Origem em maker/prototipagem.	Sem política formal	Modelos revisados sem aviso; protocolo alterado em 2019.	ALTO
STONE Technology	2013. Foco industrial desde o início.	Notificação informal	Linha mais estável que Nextion; alguns modelos ativos por 5+ anos.	MÉDIO
DWIN Technology	1999. Um dos mais antigos do segmento. Clientes industriais consolidados.	Melhor do segmento — notificação com antecedência; T5 serie ativa 10+ anos	Série T5L/DGUS estável. Compatibilidade de protocolo mantida entre gerações.	MÉDIO-BAIXO*
TDO	Fabricante com foco em segmento industrial/HMI. Portfólio de displays com host integrado.	Política não padronizada publicamente	Linha de produtos estável em aplicações específicas; verificar	MÉDIO

			disponibilidade de longo prazo com distribuidor.	
--	--	--	--	--

(*) O risco MÉDIO-BAIXO atribuído à DWIN reflete seu histórico superior de gestão de ciclo de vida dentro do segmento Smart. Contudo, a qualificação permanece condicionada a três limitações estruturais que nenhum fabricante de display Smart resolve:

- **Ausência de compromisso formal NCNR (Non-Cancellable Non-Returnable):** mesmo fabricantes com bom histórico não oferecem contratos de disponibilidade garantida por 10 a 15 anos equivalentes aos de componentes industriais certificados (ex.: Texas Instruments, NXP com product longevity programs formais).
- **Sem portabilidade de projeto entre gerações:** mesmo dentro da DWIN, uma migração de T5 para T5L exige revalidação parcial da interface — migrações entre séries exigem retrabalho significativo. Não existe abstração de portabilidade.
- **Risco de counterfeit em canais spot:** displays de alta demanda com marcas reconhecidas (Nextion, DWIN) têm histórico documentado de falsificação em distribuidores não autorizados — risco especialmente relevante para reposição em campo.

Recomendação prática: Para projetos com ciclo de vida superior a 5 anos que, por restrições de custo ou prazo, utilizem displays Smart, a DWIN (série T5L) oferece o melhor equilíbrio de estabilidade de plataforma dentro do segmento. Exija do distribuidor: (1) carta formal de disponibilidade mínima, (2) estoque de segurança contratual (buffer stock agreement), e (3) confirmação de que a série específica está em fase de crescimento ou maturidade — não em sunset. A HTMG pode auxiliar nessa avaliação para os fabricantes do portfólio.

4. Limitações dos Displays Android para Aplicações Industriais

O Android tem atratividade real: ecossistema de desenvolvimento maduro, suporte a Java/Kotlin, ampla biblioteca de widgets, e desenvolvedores facilmente disponíveis no mercado. Para aplicações de consumo, quiosques e painéis de baixa criticidade, pode ser adequado. Para aplicações industriais sérias, apresenta limitações arquiteturais difíceis de contornar.

4.1 Obsolescência do Sistema Operacional e Fragmentação

O Android foi projetado para ciclos de vida de 2 a 3 anos — o mesmo ciclo de smartphones. A indústria opera em ciclos de 7 a 15 anos. Essa incompatibilidade fundamental gera:

- **Suporte oficial limitado:** o Google oferece patches de segurança para versões do Android por no máximo 3 anos. Displays industriais com Android 9 (lançado em 2018) estão sem patches de segurança desde novembro de 2021.
- **Fragmentação de versões:** fabricantes de módulos Android embarcados (Rockchip, Allwinner) fornecem BSPs customizados que ficam 2 a 4 versões atrás do Android mainline, impedindo atualizações de segurança independentes.
- **APIs depreciadas:** aplicativos industriais desenvolvidos para Android 8 ou 9 frequentemente utilizam APIs depreciadas que deixam de funcionar em versões posteriores — forçando reescrita total em cada atualização major.

⚠ Risco de conformidade: A partir de 2024, a diretiva de Cibersegurança da UE (CRA — Cyber Resilience Act) exige que produtos com elementos digitais recebam atualizações de segurança durante todo o ciclo de vida esperado do produto. Displays Android sem suporte ativo de patches violam esse requisito para produtos destinados ao mercado europeu.

4.2 Tempo de Boot e Latência de Inicialização

O Android, mesmo em versões otimizadas, carrega um stack completo de serviços que é irrelevante para aplicações industriais. O impacto é mensurável:

Plataforma	Boot até UI operacional	Boot até I/O disponível	RAM consumida em idle
Android 11 (RK3568, otimizado)	18–35 segundos	12–20 segundos	~600–900 MB
Android (AOSP mínimo, sem GMS)	12–22 segundos	8–15 segundos	~350–500 MB
Linux Embarcado (Yocto mínimo)	3–8 segundos	2–5 segundos	~50–180 MB
Linux Embarcado (boot otimizado, SSD)	<2 segundos	<1 segundo	~30–80 MB

Em aplicações onde o painel de controle faz parte de um sistema de segurança ou deve exibir informações críticas imediatamente após energização (máquinas agrícolas, painéis de SCADA em campo), 30 segundos de boot é inaceitável.

4.3 Acesso ao Hardware e Determinismo de I/O

O Android interpõe múltiplas camadas de abstração entre a aplicação e o hardware físico. Isso é intencional para segurança em smartphones, mas problemático em sistemas industriais:

- **GPIO e I/O físico:** o acesso direto a GPIO, SPI, I²C, UART é feito via NDK (C/C++) com permissões root ou via HAL (Hardware Abstraction Layer), sem garantias de latência. Em Android puro, não existe acesso de GPIO no SDK padrão.
- **Sem garantias de tempo real:** o scheduler do Android (baseado em CFS do Linux, mas com modificações do AOSP) não é preemptivo determinístico. Tarefas do sistema (GC do Java, serviços do Android) podem causar jitter de centenas de milissegundos.
- **Impossibilidade de PREEMPT_RT:** o Android não suporta o patch PREEMPT_RT do kernel Linux, que é o padrão para sistemas embarcados com requisitos de soft real-time. Com Linux embarcado, o PREEMPT_RT é uma opção de compilação direta do kernel.
- **Gerenciamento de energia não controlável:** o subsistema Doze Mode e App Standby do Android pode suspender processos da aplicação em momentos imprevisíveis — comportamento perigoso em sistemas de monitoramento contínuo.

4.4 Segurança e Superfície de Ataque

O Android carrega consigo um ecossistema projetado para conectividade com serviços de nuvem e lojas de aplicativos — componentes que representam superfície de ataque desnecessária em sistemas industriais isolados:

- **Google Play Services:** mesmo em AOSP sem GMS, partes do framework do Android mantêm dependências de rede que precisam ser explicitamente removidas — processo complexo e sujeito a regressões.
- **CVEs acumulados:** o Android acumula centenas de CVEs ativos em versões sem suporte. O NVD (National Vulnerability Database) registra mais de 2.800 CVEs atribuídos ao Android desde 2014, com atualizações mensais contínuas.
- **Modelo de permissões inadequado para OT:** o modelo de sandbox do Android foi projetado para isolar aplicativos de usuário entre si — não para isolar um sistema embarcado industrial de redes externas (modelo Purdue, IEC 62443).
- **ADB exposto:** displays Android frequentemente deixam o Android Debug Bridge (ADB) ativo por padrão em interfaces USB OTG — vetor de ataque trivial em ambientes industriais.

Nota de segurança: A IEC 62443, norma de referência para segurança em automação industrial (OT), exige que os componentes tenham superfície de ataque mínima, atualizações de segurança garantidas durante o ciclo de vida, e capacidade de auditoria de acesso. Um display Android sem suporte ativo de patches não atende a nenhum desses critérios de forma garantida.

4.5 Custo Total de Propriedade (TCO)

A análise de custo de displays Android raramente considera o TCO completo. O custo de hardware aparentemente baixo esconde custos operacionais e de suporte significativos:

Item de Custo	Display Android	Linux Embarcado
Custo de hardware inicial	Médio–Baixo	Médio
Suporte de SO por 10 anos	NÃO garantido	SIM (LTS kernels, Yocto LTS)
Custo de retrabalho por obsolescência de API	Alto (ciclos de 2–3 anos)	Baixo (ABI estável)
Customização de hardware específico	Complexo / limitado	Total (device tree, drivers)
Certificações (CE, IEC 62443, FDA)	Difícil — SO não auditável	Viável — SO auditável e minimal
Suporte técnico pós-venda	Dependente do fabricante do módulo	Independente de fornecedor

5. Por Que Linux Embarcado é o Padrão Industrial

Linux embarcado — construído com ferramentas como Yocto Project, Buildroot ou Debian mínimo — é o sistema operacional dominante em automação industrial, equipamentos médicos, controle de processos e agtech de alto desempenho por razões técnicas e operacionais bem fundamentadas.

5.1 Controle Completo da Pilha de Software

Com Linux embarcado, o engenheiro de sistema controla cada componente da pilha de software, do bootloader ao driver de display:

- **Bootloader customizável (U-Boot):** suporte a boot seguro (Secure Boot), verificação de assinatura de firmware, recuperação por rede (TFTP) ou cartão SD.
- **Kernel configurável:** apenas os drivers necessários são compilados. Um sistema embarcado típico usa 15 a 30% dos módulos de um kernel genérico — reduzindo superfície de ataque e tamanho de imagem.
- **Rootfs mínimo:** com Yocto ou Buildroot, é possível criar imagens com menos de 50 MB que incluem somente o necessário para a aplicação. Sem serviços desnecessários, sem gerenciador de pacotes em produção.
- **Suporte a PREEMPT_RT:** o patch de kernel PREEMPT_RT transforma o Linux em um RTOS de soft real-time, com latências de interrupção na faixa de microssegundos — adequado para controle de processo e sincronização com hardware de campo.

5.2 Ciclo de Vida Longo e Suporte Garantido

- Kernels LTS (Long-Term Support) do Linux recebem patches de segurança e manutenção por 6 anos (ex: kernel 5.15 LTS: 2021–2027; kernel 6.6 LTS: 2023–2029)
- O Yocto Project oferece suporte de 2 anos por release (Kirkstone LTS: 2022–2026; Scarthgap LTS: 2024–2028)
- A ausência de dependência de fornecedor garante que o produto pode receber atualizações independentemente da existência comercial do fabricante do módulo
- Migrações de hardware são facilitadas pela portabilidade do Device Tree — a mudança de SoC exige adaptação do DTS, não reescrita da aplicação

5.3 Determinismo, Performance e Recursos de Sistema

Linux embarcado oferece acesso direto e determinístico a todos os recursos de hardware:

- **GPIO, SPI, I²C, CAN, UART:** acessíveis via sysfs, libgpiod, ou drivers de kernel diretos. Latências previsíveis com agendamento de prioridade SCHED_FIFO ou SCHED_RR.
- **DRM/KMS para display:** controle total do pipeline gráfico — resolução, frequência de atualização, HDR, rotação — sem intermediários. Frameworks como Qt Embedded, GTK, LVGL ou aplicações Wayland/Weston operam diretamente sobre DRM.
- **Conectividade industrial nativa:** protocolos como OPC-UA (open62541), MQTT (Mosquitto), Modbus TCP (libmodbus), CANopen (SocketCAN) são bibliotecas C/C++ de código aberto que se integram diretamente ao Linux sem camadas de adaptação.
- **Containers e virtualização leve:** Docker e Podman executam sobre Linux embarcado, permitindo deploy de aplicações em containers para isolamento e atualização independente dos componentes.

5.4 Segurança Auditável e Certificável

Para produtos que precisam de certificações de segurança (IEC 62443, CE Machinery Directive, FDA 21 CFR Part 11), Linux embarcado é a única opção viável:

- **Código-fonte 100% auditável:** cada componente da pilha é verificável. Auditorias de segurança (SAST, DAST, penetration testing) são realizáveis sobre o sistema completo.
- **SELinux / AppArmor:** políticas de controle de acesso mandatório permitem confinamento rigoroso de processos — padrão em sistemas que precisam demonstrar conformidade com modelos Purdue (PERA).

- **dm-verity e IMA:** verificação de integridade de rootfs em runtime, impedindo execução de código não autorizado.
- **OTA seguro:** frameworks como Mender ou SWUpdate permitem atualizações over-the-air com rollback automático, verificação de assinatura e janelas de manutenção programadas.

6. Comparativo Técnico Consolidado

Critério	Display Smart	Display Android	Linux Embarcado
Ciclo de vida garantido	2–5 anos (típico)	3–5 anos (OS support)	✓ 10–15 anos (LTS kernel)
Boot time até UI operacional	5–15 s (firmware)	18–35 s	✓ 2–8 s (otimizável <2 s)
Acesso a GPIO / I ² C / SPI	Via host MCU apenas	NDK / HAL (complexo)	✓ Nativo, direto, determinístico
Suporte PREEMPT_RT (soft RT)	Não	Não	✓ Sim (patch padrão)
Auditabilidade de segurança	Impossível (binário fechado)	Parcial (AOSP parcialmente open)	✓ Total (100% open source)
Patches de segurança 10 anos	Não	Não garantido	✓ Sim (LTS kernels)
Portabilidade de aplicação	Zero (plataforma fechada)	Limitada (versões Android)	✓ Alta (POSIX, C/C++, Qt)
Conformidade IEC 62443	Não certificável	Difícil	✓ Viável (política SELinux)
Consumo de RAM em idle	< 5 MB (MCU)	350–900 MB	✓ 30–200 MB (configurável)
OTA com rollback garantido	Não	Limitado	✓ Sim (Mender / SWUpdate)
Integração OPC-UA / MQTT nativo	Via host MCU	Via APK adicional	✓ Bibliotecas C nativas
Temperatura de operação típica	0–50°C	0–60°C	✓ –40–85°C (componentes ind.)
Suporte de logs e diagnóstico remoto	Nenhum	Limitado (logcat)	✓ journald, syslog, ELK, Loki

7. Critérios Práticos de Seleção de Plataforma

A tabela a seguir consolida os critérios de decisão para a seleção de arquitetura de display em função das características do projeto:

Característica do Projeto	Smart/Proprietário	Android	Linux Embarcado
Protótipo / PoC rápido (< 3 meses)	✓ Adequado	✓ Adequado	Investimento maior
Produção industrial (>500 unidades)	Risco alto	Risco médio	✓ RECOMENDADO
Ciclo de vida do produto > 7 anos	NÃO recomendado	NÃO recomendado	✓ RECOMENDADO
Exportação para UE (CE, CRA 2024)	Não certificável	Não garantido	✓ Viável
Integração com sensores industriais (CAN, Modbus)	Via MCU host apenas	Complexo	✓ Nativo
Temperatura ambiente > 60°C	NÃO adequado	NÃO adequado	✓ Adequado
Requisito de boot < 5 segundos	Depende	NÃO	✓ SIM (otimizável)
Aplicação com soft real-time (< 1 ms jitter)	NÃO	NÃO	✓ SIM (PREEMPT_RT)
Budget de hardware extremamente limitado	✓ Ponto forte	✓ Adequado	Custo superior
Equipe com experiência Android/Java	Não se aplica	✓ Vantagem	Curva de aprendizado
Produto com atualização OTA garantida	NÃO	Parcial	✓ SIM (Mender/SWUpdate)
Segurança cibernética auditável	NÃO	Parcial	✓ SIM

Nota para engenheiros de produto: A escolha por Android frequentemente resulta de pressão comercial por redução de custo inicial de hardware e disponibilidade de desenvolvedores. O custo real aparece no prazo de 3 a 5 anos, quando o suporte de SO encerra, as APIs mudam, e a certificação regulatória se torna obstáculo intransponível. Linux embarcado exige maior investimento inicial em engenharia de BSP, mas amortiza esse custo ao longo de um ciclo de vida 2 a 3 vezes maior.

8. Aplicações Industriais e Agtech: Análise Específica

Os displays industriais TFT fornecidos pela HTMG são projetados para operar em ambientes onde as limitações descritas neste artigo têm consequências diretas. As principais aplicações do portfólio HTMG incluem:

8.1 Automação Agrícola e Maquinário de Precisão

Máquinas agrícolas operam em condições que eliminam imediatamente displays Android e Smart como opção séria:

- Temperatura de operação: -10°C (madrugadas frias no Sul) a $+70^{\circ}\text{C}$ (cabines de máquinas no cerrado sem ar condicionado)
- Vibração e shock mecânico: ISO 16750-3 (vibração de veículo off-road) — telas com fixação inadequada e conectores de borda não sobrevivem a uma safra
- Ambiente eletromagnético severo: alternadores, motores de partida e inversores geram transientes que requerem proteção EMI (conforme CISPR 25 / ISO 11452) — não disponível em módulos Android genéricos
- Tempo de boot: um operador que aciona o implemento espera a tela estar operacional em menos de 3 segundos — padrão impossível para Android
- Ciclo de vida: tratores e pulverizadores têm vida útil de 12 a 20 anos. A tela de IHM precisa de peças disponíveis e suporte de software por igual período

8.2 IoT Industrial e Painéis de SCADA

Painéis de supervisão industrial requerem:

- **Conectividade OPC-UA e MQTT nativa:** protocolo padrão de Industry 4.0. Linux embarcado suporta o stack open62541 (OPC-UA) diretamente em C, sem overhead de JVM ou framework Android.
- **Integração com PLCs via Modbus TCP / EtherNet/IP:** bibliotecas disponíveis para Linux (libmodbus, EIP Stack Group) com latências determinísticas.
- **Logging e historian local:** SQLite, InfluxDB (versão embarcada) ou TimescaleDB sobre PostgreSQL — opções impossíveis em displays Android com 512 MB de RAM.
- **VPN e segmentação de rede:** WireGuard, OpenVPN, VLAN tagging — controle total sobre a pilha de rede em Linux, impossível em Android sem root.

8.3 Equipamentos com Conformidade Regulatória

Para equipamentos destinados à exportação (UE, EUA, Japão) ou setores regulados (agroquímico, farmacêutico, médico):

- CE Marking (Diretiva de Máquinas 2006/42/CE): exige análise de riscos de software — impossível com firmware binário fechado
- FDA 21 CFR Part 11 (sistemas de registro eletrônico): exige trilha de auditoria imutável, controle de versão de software, validação IQ/OQ/PQ — viável apenas com SO auditável
- CRA (Cyber Resilience Act, UE 2024): exige política de divulgação de vulnerabilidades e atualizações durante o ciclo de vida — Android sem suporte ativo não atende
- LGPD / GDPR para dados de processo: controle de acesso a dados de telemetria exige SO com modelo de permissões auditável

9. Conclusão

A escolha da plataforma de display para um produto industrial é uma decisão arquitetural de alto impacto que raramente pode ser revertida após o início da produção. Este artigo demonstrou que displays Smart com firmware proprietário e displays Android, apesar de suas vantagens reais na fase de prototipagem e em aplicações de ciclo curto, apresentam limitações estruturais que os inviabilizam para produtos industriais sérios.

As limitações fundamentais são de natureza arquitetural — não contornáveis com otimizações de software ou customizações do fornecedor. Ciclo de vida do SO incompatível com ciclo de vida industrial, ausência de determinismo de I/O, impossibilidade de auditoria de segurança e incompatibilidade crescente com normas regulatórias europeias e americanas são problemas que emergem sistematicamente em campo.

Linux embarcado — com toda a curva de aprendizado e investimento inicial que representa — é o único caminho sustentável para produtos que precisam operar por 10 a 15 anos, passar por certificações regulatórias, e oferecer suporte de campo previsível.

A HTMG fornece displays TFT industriais selecionados para operar com Linux embarcado, com suporte de BSP (Board Support Package), documentação de device tree e engenharia de aplicação para integração em plataformas i.MX, Rockchip industrial e AM64x. Contato técnico: displays@htmg.com.br | +55 (19) 3869-7017

Referências Técnicas

- IEC 62443-4-2: Security for Industrial Automation and Control Systems — Component Security Requirements
- Android Enterprise Recommended Requirements (Google, 2023)
- Yocto Project Long-Term Support Releases (Kirkstone, Scarthgap) — yoctoproject.org
- Linux Kernel LTS Support Timeline — kernel.org/category/releases.html
- Mender.io: Over-the-Air Software Update Framework for Embedded Linux
- SWUpdate: Software Update for Embedded Systems (sbabic.github.io/swupdate)
- EU Cyber Resilience Act (CRA) — Regulation (EU) 2024/2847
- MISRA C:2023 — Guidelines for the Use of the C Language in Critical Systems
- NVD — National Vulnerability Database: Android CVE History (nvd.nist.gov)
- ISO 16750-3: Road Vehicles — Environmental Conditions — Mechanical Loads
- CISPR 25: Radio Disturbance Characteristics for Protection of Receivers Used on Board Vehicles
- [open62541](https://github.com/open62541/open62541): Open Source OPC UA Stack (github.com/open62541/open62541)
- Linux Foundation — Embedded Linux Landscape Report 2024
- Buildroot: A Simple, Efficient and Easy-to-Use Embedded Linux System (buildroot.org)
- IPC-7711/7721: Rework, Modification and Repair of Electronic Assemblies