



Descaracterização de TV Box TX9 para Armbian Linux: para uso como plataforma de computação embarcada educacional

Romulo de Aguiar Beninca | romulo.beninca@ifsc.edu.br¹
Leonardo Ronald Perin Rauta | leonardo.rauta@ifsc.edu.br²
Ademir Goulart | ademir.goulart@ifsc.edu.br³
Erik Adelar Klein | erikadelarklein@gmail.com⁴
Vinícius Soares dos Santos | vinicius.ss2008@aluno.ifsc.edu.br⁵

Edital de Ensino nº 28/2024
Trabalho de Pesquisa e Extensão
DT 4 - Inovação Tecnológica, Processos Produtivos e Futuro das Tecnologias

Resumo:

Este trabalho apresenta a descaracterização técnica de dispositivos TV Box TX9 Pro, originalmente destinados ao consumo multimídia, com o objetivo de convertê-los em plataformas seguras de computação embarcada de baixo-custo para ensino, pesquisa aplicada e prototipagem. A motivação decorre da presença de firmwares Android desatualizados, inconsistentes e potencialmente inseguros, incluindo processos com privilégios elevados, serviços desconhecidos e versões adulteradas do sistema operacional. A metodologia envolveu análise do firmware via ADB (Android Debug Bridge), extração e engenharia reversa do *Device Tree Blob* (DTB), identificação de componentes de hardware e desenvolvimento de drivers e serviços para periféricos específicos, como o módulo WiFi SSV6051, o receptor infravermelho e o display frontal FD6551. O sistema original foi substituído por Armbian Linux, com kernel atualizado, suporte a Docker e integração dos principais dispositivos. Como resultado, obteve-se um minicomputador funcional, com interfaces de rede, controle remoto e display operacional, apto para aplicações em sistemas embarcados, redes, segurança e Internet das Coisas. Conclui-se que a descaracterização amplia a vida útil dos equipamentos, reduz riscos de segurança e viabiliza seu uso como recurso didático em ambientes educacionais.

Palavras-chave: linux embarcado; engenharia reversa; tv box; segurança computacional; computação educacional

Introdução:

Dispositivos TV Box de baixo custo, baseados em arquiteturas ARM (*Advanced RISC Machine*) como o SoC (*System-on-Chip*) Amlogic S905W, são amplamente

¹ Servidor da área de Informática do IFSC – Câmpus Gaspar.

² Servidor da área de Informática do IFSC – Câmpus Gaspar.

³ Servidor da área de Informática do IFSC – Câmpus Gaspar.

⁴ Estudante do curso Técnico integrado em Informática do IFSC – Câmpus Gaspar.

⁵ Estudante do curso Técnico integrado em Informática do IFSC – Câmpus Gaspar.

distribuídos no Brasil, sendo frequentemente utilizados para consumo multimídia. Parte desses equipamentos, oriundos de apreensões realizadas pela Receita Federal.

Entretanto, a utilização direta desses dispositivos apresenta limitações relevantes, especialmente no que diz respeito à segurança e confiabilidade do software embarcado. Análises do firmware Android original evidenciam a presença de processos com privilégios elevados sem documentação clara, sistemas desatualizados, serviços de origem desconhecida e inconsistências entre a versão declarada e a versão real do sistema operacional. Essas características tornam o ambiente inadequado para uso institucional.

Além disso, a simples instalação de aplicações sobre o sistema original não elimina tais vulnerabilidades, uma vez que as camadas fundamentais permanecem inalteradas. Torna-se, portanto, necessário um processo de descaracterização técnica, no qual o dispositivo deixa de exercer sua função original e passa a operar como uma plataforma genérica de computação embarcada, baseada em software livre e auditável.

Nesse contexto, este trabalho tem como objetivo realizar a descaracterização de dispositivos TV Box TX9, por meio da substituição do sistema Android por uma distribuição Linux embarcada, associada à recuperação funcional de seus principais componentes de hardware. Para isso, são empregadas técnicas de análise de firmware, engenharia reversa do Device Tree e desenvolvimento de drivers para integração de periféricos. Como resultado, busca-se transformar esses dispositivos em plataformas reutilizáveis para ensino, pesquisa aplicada e experimentação em áreas como sistemas embarcados, redes e segurança da informação.

Fundamentação teórica

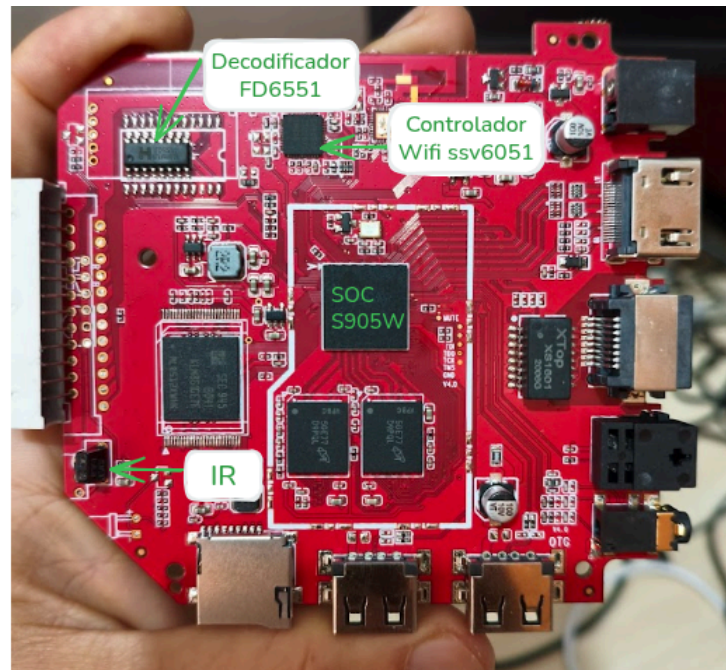
Sistemas embarcados baseados em ARM, como TV Boxes, integram CPU, memória e periféricos em um único SoC (System-on-Chip). Diferente de computadores com BIOS/UEFI, essas plataformas dependem do Device Tree para informar ao kernel Linux os componentes presentes na placa, como interfaces de rede, portas GPIO (General Purpose Input/Output), receptores infravermelho e controladores de display (Kernel Documentation, 2023).

Nesse contexto, distribuições Linux embarcadas, como o Armbian, permitem substituir firmwares Android desatualizados ou pouco transparentes por ambientes mais controláveis, auditáveis e atualizáveis. Associada à engenharia reversa, essa abordagem possibilita reaproveitar dispositivos originalmente destinados ao consumo multimídia como plataformas educacionais para ensino, pesquisa aplicada e experimentação em sistemas embarcados, redes e segurança. (Armbian, 2024).

Procedimentos metodológicos

O trabalho foi conduzido em quatro etapas: análise do hardware e do firmware original, instalação do sistema operacional, recuperação dos periféricos e validação funcional. Inicialmente, realizou-se a inspeção visual da placa e a análise do Android embarcado via ADB. A inspeção, Figura 1, permitiu identificar o SoC Amlogic S905W, o controlador de display FD6551, o módulo Wi-Fi SSV6051 e o receptor infravermelho. A análise do software revelou inconsistências relevantes, como divergência entre a versão Android informada e a versão real do sistema, uso de kernel antigo e presença de componentes pouco documentados.

Figura 1 — Placa do TX9 Pro: SoC Amlogic S905W, controlador FD6551



Fonte: Autores (2026)

Em seguida, o *Device Tree Blob* foi extraído e descompilado para mapear os principais periféricos e orientar a substituição do firmware original. Com base nessas informações, foi instalada a distribuição Armbian, inicialmente em cartão SD e, após validação, na memória interna do dispositivo (Beninca, 2026).

Primeiramente, a integração do módulo Wi-Fi SSV6051 exigiu a adaptação de um driver externo aberto, originalmente voltado a versões anteriores do kernel Linux, para compatibilizá-lo com o kernel moderno utilizado no Armbian. Foram ajustados os cabeçalhos e a identificação de versão do kernel, compilado o módulo `ssv6051.ko`, instalado o módulo no diretório correspondente e copiados os arquivos auxiliares de firmware necessários para a inicialização correta da interface sem fio (Beninca, 2026). Em segundo lugar, a implementação do controle do *display* FD6551 foi realizada por meio de uma técnica de *bit-bang* via portas GPIO, o que demandou a escrita de um código customizado para respeitar a temporização (*timing*) e a sequência específica de inicialização exigidas pelo controlador. Por fim, a ativação do receptor infravermelho (IR) foi feita via mapeamento direto no *Device Tree* e no *kernel*, permitindo a posterior captura e tradução dos códigos emitidos pelo controle remoto original.

Por fim, foram realizados testes funcionais e de desempenho para verificar a inicialização do sistema, o funcionamento das interfaces de rede, do display frontal e do receptor infravermelho. Também foram avaliadas a estabilidade da conexão Wi-Fi e a viabilidade do uso do dispositivo como plataforma Linux embarcada para aplicações educacionais.

Resultados e Discussões

4.1 Display Frontal e Receptor IR

Após a implementação dos serviços, o display frontal passa a exibir um contador crescente a partir do primeiro ciclo de clock após power-on (boot service), transitando para relógio HH:MM com indicadores de LAN/Wi-Fi após a estabilização da rede. As Figuras 2 exibe o dispositivo durante os testes de integração entre receptor IR (Infravermelho) e display: ao pressionar botões no controle original, o display exibe a confirmação da ação.

Figuras 2 — Testes de integração: controle remoto original tvbox operando com display.



Fonte: Autores (2026)

4.2 Desempenho de Rede

Foram realizados testes de transmissão de dados utilizando a ferramenta `iperf3`, com duração de 30 segundos, sucessivas vezes.

Tabela 2 – Resultados de throughput de rede após descaracterização.

Interface	Direção	retransmissões	Throughput médio	Throughput máx.
Ethernet	Upload	740	91,4 Mbps	~115 Mbps
Ethernet	Download	61	94,1 Mbps	~95 Mbps
Wi-Fi SSV6051	Upload	386	6,9 Mbps	~9,4 Mbps
Wi-Fi SSV6051	Download	13	7,5 Mbps	~13,6 Mbps

Fonte: Autores (2026)

A interface Ethernet operou próxima ao limite teórico de 100 Mbps (*Fast Ethernet*), com 61 retransmissões em download, demonstrando que o driver interno do S905W funciona corretamente no kernel mainline. O Wi-Fi SSV6051, embora funcional, apresentou throughput médio de ~7 Mbps e alta variabilidade, com quedas momentâneas para 0 bps. Em 48 horas de operação contínua, a interface wlan0 permaneceu ativa sem travamentos, confirmando a eficácia da correção de power management aplicada. Para aplicações que exijam largura de banda, recomenda-se a interface Ethernet.

Considerações finais:

Os resultados demonstram que dispositivos TV Box de baixo custo podem ser descaracterizados e convertidos em plataformas funcionais de computação embarcada, desde que se realize uma análise técnica detalhada do hardware e do firmware original. A substituição do ambiente Android por uma distribuição Linux embarcada possibilitou mais controle, transparência e segurança, eliminando dependências de software proprietário e reduzindo riscos de componentes desconhecidos ou desatualizados.

A recuperação funcional dos principais periféricos — incluindo interface de rede, receptor infravermelho e display frontal — evidenciou que, embora esses dispositivos apresentem limitações e ausência de documentação oficial, é possível restabelecer sua operação por meio de técnicas de engenharia reversa e desenvolvimento de drivers. Destaca-se que o processo não se limita à instalação de um novo sistema operacional, exigindo intervenções em nível de kernel, firmware e descrição de hardware.



Uma vez descaracterizados e com o Armbian Linux funcional, estes minicomputadores adquirem uma vasta gama de aplicações práticas. Eles podem ser empregados na infraestrutura de laboratórios como servidores DNS/DHCP locais, plataformas para execução de containers Docker em aulas de redes, concentradores para sensores de Internet das Coisas (IoT) na automação de ambientes, ou até mesmo atuarem como instâncias de controle em projetos de robótica e segurança da informação.

Do ponto de vista educacional e institucional, a proposta mostrou-se relevante ao viabilizar o reaproveitamento de equipamentos que, de outra forma, seriam descartados ou subutilizados. A transformação desses dispositivos em minicomputadores permite sua aplicação em atividades práticas envolvendo sistemas operacionais, redes, segurança da informação e Internet das Coisas, aproximando estudantes de problemas reais de engenharia.

Por fim, conclui-se que a descaracterização de TV Boxes representa uma alternativa viável para ampliação do ciclo de vida de dispositivos eletrônicos, contribuindo tanto para a redução do lixo eletrônico quanto para a criação de ambientes didáticos mais acessíveis e experimentais. Como trabalhos futuros, sugere-se a ampliação do suporte a diferentes modelos de hardware, a integração com plataformas de automação e o desenvolvimento de materiais didáticos estruturados para uso em sala de aula.

Referência ao fomento recebido

Agradecemos ao Instituto Federal de Santa Catarina – Câmpus Gaspar, pelo apoio por meio do Edital de Ensino nº 28/2024, fundamental para a viabilização deste projeto.

Referências

Arbiam. Armbian: optimized Linux for ARM boards. Disponível em:

<https://www.armbian.com/>. Acesso em: 21 abr. 2026.

Docker. Install Docker Engine Ubuntu. Disponível em:

<https://docs.docker.com/engine/install/ubuntu/>. Acesso em: 20 abr. 2026.



Linux Kernel. Linux and the Devicetree. Disponível em:

<https://docs.kernel.org/devicetree/usage-model.html>. Acesso em: 21 abr. 2026.

BENINCA, Romulo de Aguiar. Relatório técnico: transformação e descaracterização de TV Box TX9 para Armbian Linux. GitHub, 2026. Disponível em:

https://github.com/rbeninca/tvbox/blob/main/RELATORIO_TECNICO.MD. Acesso em: 2 jun. 2026.