

PROVA DE CONCEITO DE UM SISTEMA HÍBRIDO PARA ATUALIZAÇÃO REMOTA DA CONFIGURAÇÃO (BITSTREAM) EM PLATAFORMAS EMBARCADAS BASEADAS EM FPGA

Pré-Projeto apresentado ao Programa EmbarcaTech, do Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE), como requisito parcial para a avaliação do Projeto Final da Residência Tecnológica em Sistemas Embarcados.

Autor: Marcio Barbosa | 06/11/2025

Orientador: Prof. Dr. David Ciarlini C. Freitas | **Mentor(a):** Luana Moura

1. RESUMO

Sistemas embarcados evoluíram para arquiteturas híbridas, combinando microcontroladores e FPGAs para alcançar desempenho e paralelismo. Essa combinação permite não apenas o processamento em tempo real, mas também a atualização remota de firmware, em ambientes de operação contínua.

O objetivo é desenvolver uma Prova de Conceito (PoC) de um sistema híbrido composto por um microcontrolador RP2040 executando *FreeRTOS* e uma FPGA *Lattice ECP5*, capaz de realizar atualizações OTA (*Over-the-Air*) do *bitstream* da FPGA, além de interagir com sensores de telemetria em tempo real. O projeto propõe processar, comunicar e reconfigurar a MCU e FPGA, estabelecendo aprendizado para futuras aplicações.

2. PROBLEMATIZAÇÃO E JUSTIFICATIVA

O crescente número de dispositivos embarcados e de Internet das Coisas (IoT) implantados em campo impõe desafios logísticos e de segurança. Segundo El Jaouhari e Bouvet (2022), até 2025 o total de dispositivos conectados deve superar o dobro da população mundial. Métodos convencionais de atualização exigem acesso físico, elevando custos operacionais e dificultando o gerenciamento em larga escala (Brown, 2018).

A técnica Over-the-Air (OTA), amplamente utilizada em microcontroladores, consiste na transmissão remota de **firmware**¹ por tecnologias como Wi-Fi, GSM ou BLE. No entanto, sua aplicação em FPGAs ainda é limitada, envolve a substituição do *bitstream*², configuração lógica que define o **hardware reconfigurável**³ e não apenas o software do microcontrolador.

Segundo Dubey (2009), as FPGAs destacam-se por permitir reconfiguração dinâmica e execução determinística em paralelo, sendo ideais para aplicações que exigem atualização rápida e resposta em tempo real. O

¹ **Firmware**, El Jaouhari e Bouvet (2022) definem o termo OTA como a transmissão de um software, neste caso firmware. Dubey (2009) separa claramente o fluxo de design de hardware (Hardware/*Bitstream*) do fluxo de design de software (Firmware), código-fonte de software compilado para o processador embarcado.

² **Bitstream**, configuração lógica que define o hardware. El Jaouhari e Bouvet (2022) descrevem o processo de boot onde o *bitstream* é copiado da memória flash externa para a FPGA. Dubey (2009), arquivo de configuração que é carregado em FPGAs baseadas em SRAM a cada inicialização.

³ **Hardware reconfigurável**, refere-se aos chips físicos (RP2040, ECP5) ou, no contexto da FPGA, à lógica configurada (os circuitos) criada com seu código Verilog. Dubey (2009) define o fluxo de design de hardware (HDL) como sendo adequado para aplicações rápidas em tempo real.

autor também ressalta que o particionamento entre hardware e software deve respeitar restrições temporais: **tarefas de alta frequência, como controle SPI, devem residir em hardware, enquanto o gerenciamento e a supervisão são mantidos em software.**

A proposta deste projeto, que utiliza o microcontrolador RP2040 como servidor de controle (camada de software) e a FPGA Lattice ECP5 como cliente reconfigurável (camada de hardware), figura 1. Esta proposta adota o microcontrolador RP2040, executando o FreeRTOS, como servidor OTA e a FPGA Lattice ECP5 como cliente reconfigurável, responsável pela aplicação do novo bitstream. A comunicação entre ambos ocorrerá via *Serial Peripheral Interface (SPI)*, cujo controle, conforme Dubey (2009), pode ser implementado em uma máquina de estados finitos (FSM) dentro da FPGA, garantindo sincronismo e integridade temporal dos dados. Sensores para telemetria serão adicionados na RP2040 e FPGA para apresentar troca de dados e comunicação entre as soluções.

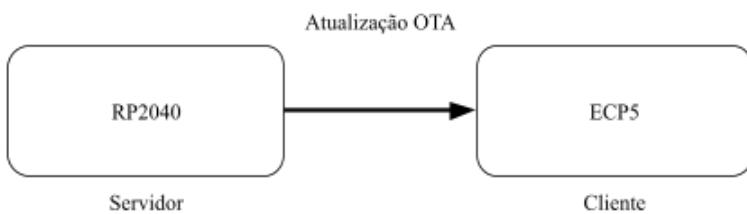


Figura 1. Exemplo de interação servidor e cliente.

O projeto propõe validar dois cenários:

- **Cenário 1 – OTA Persistente:** o RP2040 grava o bitstream na memória Flash externa, e a ECP5 recarrega a configuração no boot;
- **Cenário 2 – OTA Direta:** o RP2040 envia o bitstream diretamente à SRAM interna da ECP5, para reconfiguração rápida.

Dessa forma, o projeto busca adaptar princípios consolidados de OTA em MCUs para o contexto de FPGAs, ampliando a compreensão sobre métodos de **atualização remota de hardware reconfigurável** e fornecendo subsídios técnicos para aplicações futuras em sistemas embarcados híbridos.

3. SOLUÇÃO PROPOSTA

3.1. Objetivo Geral: Desenvolver uma Prova de Conceito (PoC) de um sistema embarcado híbrido composto por um microcontrolador RP2040 (executando *FreeRTOS*) e uma ECP5. O projeto validará a atualização remota do hardware da FPGA (o envio do *bitstream*), um processo que será gerenciado pelo firmware do RP2040 via protocolo SPI. Em seguida, será validada a troca de dados em tempo real entre ambos os dispositivos.

3.2. Arquitetura da Solução e Abordagem: A solução proposta consiste em uma arquitetura híbrida (Figura 2), onde o RP2040 atua como servidor de controle e a ECP5 como cliente reconfigurável. O firmware do RP2040, gerenciado pelo FreeRTOS, coordenará o processo OTA e a troca de dados, validando os dois cenários de atualização.

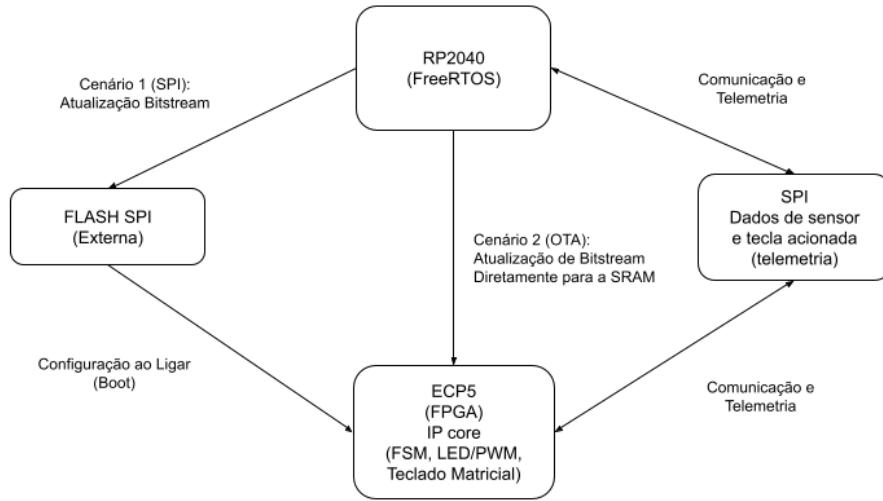


Figura 2 - Arquitetura do sistema MCU - FPGA.

Durante os testes, o **RP2040** coletará dados do sensor **AHT10** e enviará à **ECP5**, que os processará e acionará LEDs conforme a variação de temperatura. A **ECP5** executará a varredura de um teclado matricial, implementado por uma máquina de estados finitos (FSM), retornando o valor da tecla ao **RP2040** para exibição no terminal serial.

3.3. Objetivos Específicos

1. Implementar o ambiente de controle no RP2040 sob *FreeRTOS*, estruturando tarefas para:
 - a. Leitura do sensor de temperatura (AHT10), via I2C;
 - b. Envio dos dados de telemetria (temperatura) à ECP5 via SPI;
 - c. Recebimento de dados (tecla pressionada) da ECP5 via SPI e exibição no visor;
 - d. Execução de tarefas de atualização OTA (upload de *bitstream*, verificação, reconfiguração).
2. Implementar o ambiente de controle no ECP5, desenvolver IP *Cores* modulares (em SystemVerilog):
 - a. Receber dados de telemetria (temperatura) do RP2040 (via SPI Slave);
 - b. Acionar periféricos internos (LEDs) com base nos dados recebidos;
 - c. Implementar um IP Core (FSM) para a varredura do Teclado Matricial;
 - d. Enviar o dado (tecla pressionada) para o RP2040 via SPI Slave;
 - e. Exportar registradores de versão e estado para o RP2040.
3. Implementar e validar o processo de atualização OTA da ECP5 utilizando os dois cenários Persistente (Flash) e Direto (SRAM).
4. Validar os testes da PoC em bancada, confirmando a funcionalidade do projeto:
 - a. Tempo de atualização OTA e resposta de reconfiguração da ECP5;
 - b. Comunicação SPI contínua pós-update;
 - c. Consistência dos dados do sensor e leitura de teclas do teclado.

4. REQUISITOS DE SISTEMA

4.1. Requisitos do Sistema

Hardware	Software
MCU: Raspberry Pi RP2040 (BitDogLab)	FreeRTOS para gerenciamento de tarefas no RP2040
FPGA: Lattice ECP5 (placa de desenvolvimento)	Drivers I2C e SPI implementados em C
Sensor: AHT10 e teclado matricial	Ferramentas de simulação: Icarus Verilog, GTKWave
Interfaces: <ul style="list-style-type: none">• SPI para comunicação MCU–FPGA;• I2C para sensores;• UART/USB para depuração;	IP Cores em SystemVerilog para recepção SPI, FSM de controle e periféricos
Alimentação: 5 V / 3.3 V	Ferramentas de síntese: Lattice Diamond, e soluções para VScode
Memória Flash SPI externa para armazenamento de bitstream	

Tabela 1 - Requisitos técnicos.

5. METODOLOGIA E CRONOGRAMA DE EXECUÇÃO

O desenvolvimento será dividido nas três fases macro definidas pelo Programa:

Etapa	Entregável	Data	Detalhes
1	Pré-Projeto	06/11/25	Revisão Bibliográfica e Definição do Projeto
2	Relatório de Evidências	21/12/25	Evolução dos Cenários
3	Relatório Final e Vídeo	08/02/26	Projeto Final

Tabela 2 - Cronograma de Execução do Projeto.

6. REFERÊNCIAS

BROWN, Benjamin Bucklin. Over-the-Air (OTA) *Updates in Embedded Microcontroller Applications: Design Trade-Offs and Lessons Learned. Analog Dialogue*, v. 52, n. 11, p. 1-7, nov. 2018.

EL JAOUHARI, Saad; BOUVET, Eric. *Toward a generic and secure bootloader for IoT device firmware OTA update*. In: 2022 INTERNATIONAL CONFERENCE ON INFORMATION NETWORKING (ICOIN), 36., 2022, Jeju. Anais [...]. [S.l.]: IEEE, 2022. p. 90-95. DOI: 10.1109/ICOIN53446.2022.9687242.

DUBEY, Rahul. *Introduction to Embedded System Design Using Field Programmable Gate Arrays*. London: Springer-Verlag, 2009. ISBN 978-1-84882-015-9.