# DESENVOLVIMENTO E ATUALIZAÇÃO DE MÓDULOS PARA EXECUÇÃO E/OU SIMULAÇÃO DE SISTEMAS BASEADOS BLOCOS DE FUNÇÕES ADERENTES À IEC61499 PARA VÁRIAS PLATAFORMAS

Camile Silva Neves, Yuri Kaszubowski Lopes, André Bittencourt Leal,
Herbert Albérico De Sá Leitão, Roberto Silvio Ubertino Rosso Junior

## INTRODUÇÃO

A norma IEC 61499 estabelece uma abordagem para o desenvolvimento de sistemas de automação baseados em blocos de função, sendo adequada a aplicações distribuídas. Entre suas principais características estão a modularidade, que permite organizar uma aplicação em blocos com funções específicas; a portabilidade, que possibilita sua utilização em diferentes plataformas; a distribuição, que permite executar partes da aplicação em diferentes dispositivos; e a interoperabilidade, relacionada à integração entre componentes e plataformas distintas [1].

Nesse contexto, o ICARU_FB é uma infraestrutura de software voltada à execução de aplicações aderentes à IEC 61499 em diferentes plataformas [1]. O projeto foi inicialmente desenvolvido pelo professor Leandro Israel Pinto e colaboradores, posteriormente passou por trabalhos de continuidade e atualização. Uma de suas partes principais é a ICARU_VM, uma máquina virtual desenvolvida em linguagem C e responsável pela execução das aplicações formadas por blocos de função.

O projeto utilizava originalmente módulos desenvolvidos em Python 2. Com a evolução dos sistemas computacionais, tornou-se necessária a atualização para Python 3, buscando manter a compatibilidade com computadores e sistemas operacionais atuais. Assim, este trabalho teve como objetivos atualizar os módulos para Python 3, configurar e validar a infraestrutura de execução da ICARU_VM em um Arduino Mega 2560 e iniciar a adaptação da camada responsável pela interação da máquina virtual com a plataforma ESP32.

## DESENVOLVIMENTO

Inicialmente, foram estudadas a norma IEC 61499, a infraestrutura ICARU_FB e as ferramentas Eclipse 4diac e Function Block Development Kit (FBDK), utilizadas para modelagem e execução de aplicações baseadas em blocos de função [1][2]. Em seguida, foram atualizados módulos e *scripts* originalmente desenvolvidos para Python 2, com atenção às diferenças no tratamento de *bytes*, textos e comunicação por *sockets* existentes entre as versões da linguagem.

Para validar o ambiente atualizado, foi configurada a comunicação entre o computador e um Arduino Mega 2560. Nesse processo, o DUT (*Device Under Test*) foi utilizado como componente responsável pela comunicação com o dispositivo em teste, enquanto o SERIALSERVER intermediou a comunicação pela porta serial. Esses componentes foram necessários para permitir a troca de informações entre o FBDK, os serviços de comunicação e a ICARU_VM.

Além da comunicação serial, foi utilizada a comunicação TCP (*Transmission Control Protocol*) entre os programas envolvidos na infraestrutura. O TCP é um protocolo de comunicação que estabelece uma conexão entre aplicações e permite a transmissão organizada e confiável de dados. No ambiente ICARU_FB, essa comunicação foi utilizada pelo DUT para trocar informações com os demais componentes do sistema, enquanto a comunicação serial foi utilizada na ligação com a placa. A separação entre esses meios de comunicação permitiu que os programas executados no computador se comunicassem com a infraestrutura e que esta, por sua vez, realizasse a comunicação com a ICARU_VM.

O arranjo experimental montado para a validação (Figura 1) permitiu comparar a resposta da infraestrutura sob as versões 2 e 3 do Python. Foram realizados testes de comunicação serial e TCP e verificada a execução da ICARU_VM no Arduino Mega 2560. Os blocos de função utilizados nesses testes já faziam parte do projeto ICARU_FB, sendo empregados para validar a infraestrutura existente. A estrutura da aplicação BasicIcaruIo e os blocos funcionais utilizados nessa etapa podem ser visualizados na Figura 2.

Paralelamente, foi iniciada a adaptação da camada de interação da ICARU_VM para o ESP32. Foram analisadas as diferenças entre a implementação utilizada no Arduino Mega 2560 e os recursos disponíveis no ESP32. A partir dessa análise, foram realizadas modificações preliminares nos módulos relacionados à memória, armazenamento e temporização, buscando manter a estrutura da máquina virtual e adequar os componentes dependentes da plataforma. O módulo de temporização, por exemplo, foi adaptado para utilizar o mecanismo de contagem de tempo disponibilizado pelo ESP32 [3].

## RESULTADOS

A atualização dos módulos para Python 3 possibilitou executar a infraestrutura em um ambiente computacional atual. Durante os testes, foram identificadas e tratadas diferenças relacionadas ao uso de *bytes*, textos e *sockets* entre as versões da linguagem, permitindo a continuidade da execução dos componentes.

A integração entre FBDK, DUT, SERIALSERVER e ICARU_VM foi validada no Arduino Mega 2560. Os testes demonstraram o funcionamento da comunicação serial entre o computador e a placa, bem como o estabelecimento da comunicação TCP utilizada pelo DUT.

Durante o carregamento de aplicações pelo FBDK, foi identificado o estado NOT_READY, impedindo a conclusão da execução da aplicação. Esse comportamento foi observado tanto no ambiente atualizado para Python 3 quanto no ambiente original com Python 2. Dessa forma, os testes realizados não relacionam o problema especificamente à migração entre as versões da linguagem, sendo necessária uma investigação posterior para identificar sua causa.

Na plataforma ESP32, foram identificadas diferenças em relação à implementação utilizada no Arduino Mega 2560 e realizadas adaptações preliminares nos módulos dependentes da plataforma, principalmente nos componentes relacionados à memória, armazenamento e temporização. Essas modificações constituem a base para a continuidade da implementação e dos testes no ESP32.

## CONSIDERAÇÕES FINAIS

O trabalho contribuiu para a atualização da infraestrutura ICARU_FB e para a compreensão de sua organização e mecanismos de comunicação. A migração para Python 3 tornou os módulos compatíveis com ambientes computacionais mais atuais, enquanto os testes no Arduino Mega 2560 permitiram verificar o funcionamento da comunicação necessária à execução da ICARU_VM.

A adaptação para o ESP32 também permitiu identificar diferenças entre as plataformas e realizar modificações preliminares nos módulos dependentes de hardware, principalmente nos componentes relacionados à memória, armazenamento e temporização.

Como continuidade, pretende-se finalizar a integração dos módulos, realizar a compilação completa da máquina virtual no ESP32 e validar sua execução física utilizando os blocos de função já existentes no projeto. Também deverá ser investigada a causa do estado NOT_READY identificado durante os testes com o FBDK.

**Palavras-chave:** IEC 61499; sistemas embarcados; indústria 4.0; ICARU_FB; Arduino Mega 2560; comunicação serial.

**ANEXOS**

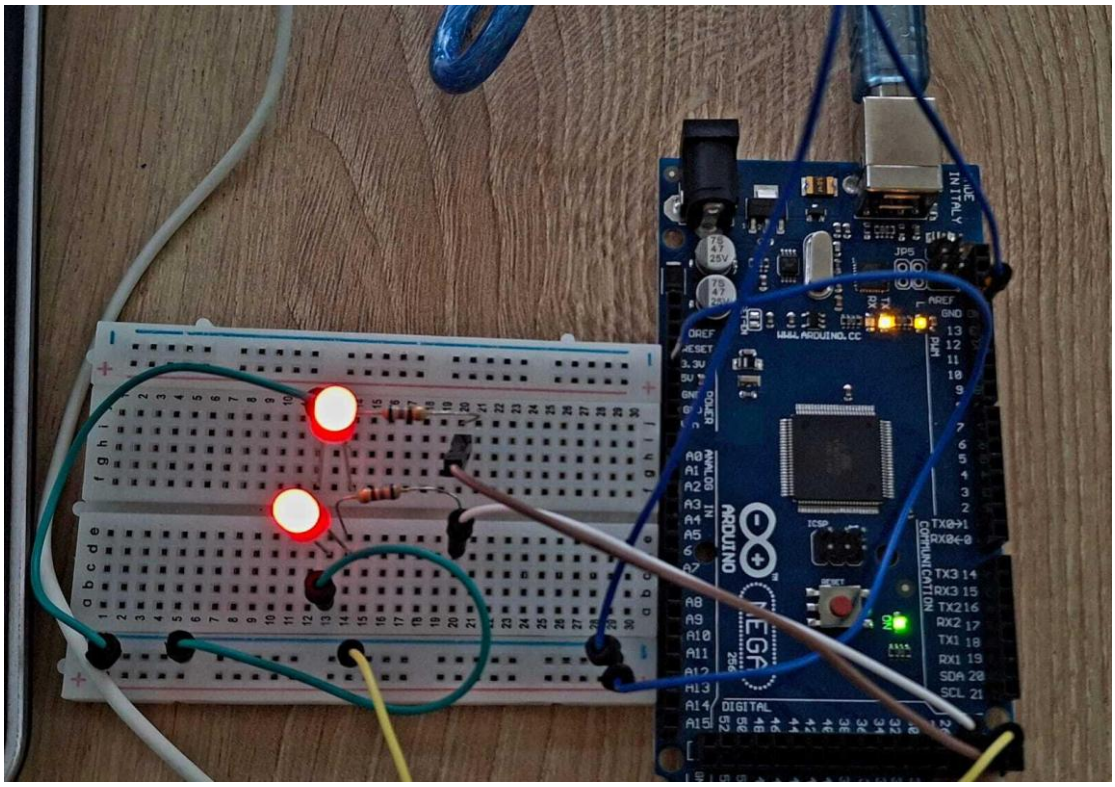


**Figura 1.** *Arranjo experimental utilizado na comparação do comportamento da infraestrutura ICARU_FB executada em ambientes baseados em Python 2 e Python 3.*

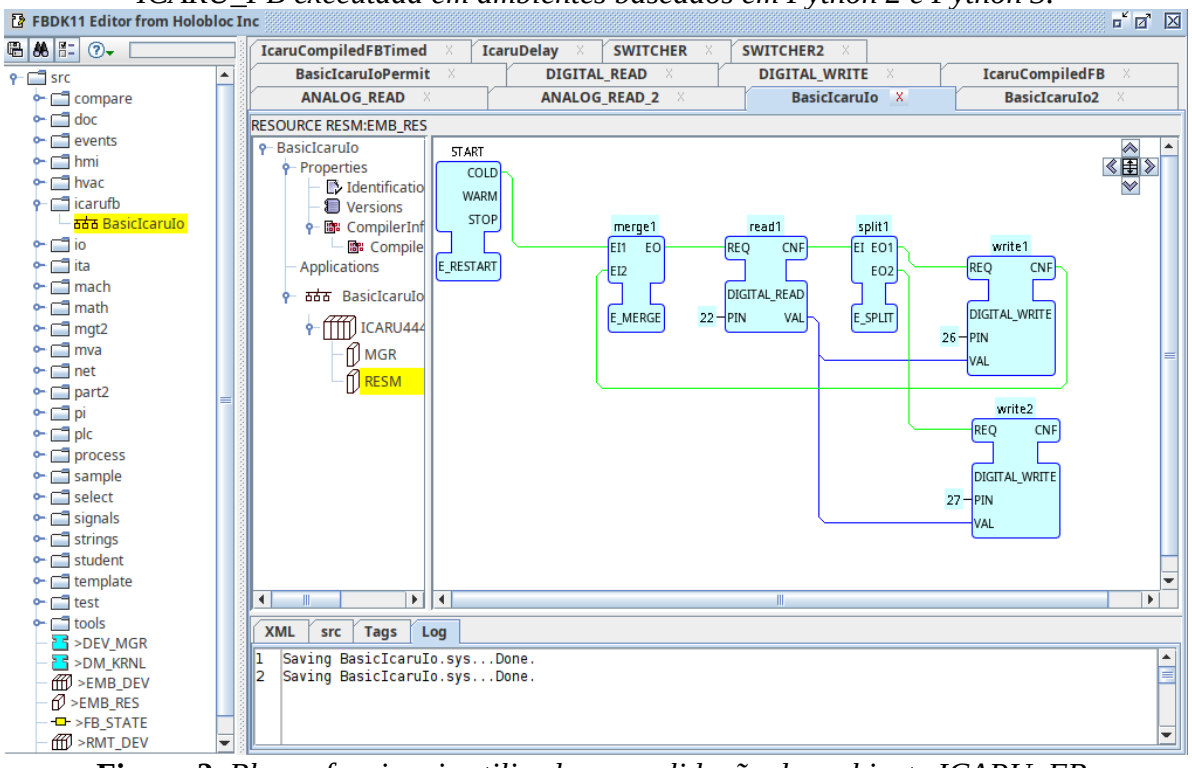



**Figura 2.** *Blocos funcionais utilizados na validação do ambiente ICARU_FB.*

**REFERÊNCIAS BIBLIOGRÁFICAS**

[1] PINTO, Leandro Israel; VASCONCELLOS, Cristiano D.; ROSSO, Roberto Silvio Ubertino; NEGRI, Gabriel Hermann. ICARU-FB: an IEC 61499 compliant multiplatform software infrastructure. IEEE Transactions on Industrial Informatics, v. 12, n. 3, 2016. Disponível em: https://ieeexplore.ieee.org/document/7445830. Acesso em: 20 jul. 2026.

[2] PINTO, Leandro I.; LEAL, André B.; ROSSO, Roberto S. U. Safe dynamic reconfiguration through supervisory control in IEC 61499 compliant systems. In: 2017 IEEE 15th International Conference on Industrial Informatics (INDIN). [S. l.]: IEEE, 2017. Disponível em: https://ieeexplore.ieee.org/document/8104866. Acesso em: 20 jul. 2026.

[3]ESPRESSIF SYSTEMS. ESP32 series: datasheet: version 5.3. [S. l.]: Espressif Systems, 2024. Disponível em: https://documentation.espressif.com/esp32_datasheet_en.pdf. Acesso em: 20 jul. 2026.