

Desenvolvimento de um Supervisório Educacional para Braço Robótico Robix com Arduino Mega e Python

André Victor Lucio¹, Namem Rachid Jaudy Neto¹, Ronan Marcelo Martins¹

Instituto Federal do Mato Grosso (IFMT) – Cuiabá, MT, Brasil

{a.lucio, namem.rachid}@estudante.ifmt.edu.br,
ronan.martins@ifmt.edu.br

Abstract. *Educational robotics demands open-source and low-cost implementation software tools for practical experiments. In this context, this work presents the development of an accessible educational supervisory system to control a "Robix" robotic arm, integrating a Python interface and an Arduino Mega controller. This is an applied research in which the Educational Industrial Robotics Supervisory (SuRI-Edu) was created, allowing real-time control via sliders and pose-based programming. Validation was conducted on a test bench through a pick-and-place experiment. Results demonstrated 100% success in trajectory execution with a communication latency of less than 50 ms, highlighting the solution's potential as an autonomous resource for the empirical validation of forward kinematics and trajectory planning in industrial robotics.*

Resumo. *A robótica educacional demanda ferramentas de software de código aberto (open-source) e de baixo custo de implementação para experimentações práticas. Nesse contexto, este trabalho apresenta o desenvolvimento de um supervisório educacional acessível para controlar um braço robótico "Robix", integrando uma interface em Python e um controlador Arduino Mega. Trata-se de uma pesquisa aplicada na qual foi criado o Supervisório de Robótica Industrial Educacional (SuRI-Edu), permitindo o controle em tempo real via sliders e programação por poses. A validação ocorreu em bancada por meio de um experimento de pick-and-place. Os resultados demonstraram 100% de sucesso na execução das trajetórias com latência de comunicação inferior a 50 ms, evidenciando o potencial da solução como um recurso autônomo para a validação empírica da cinemática direta e planejamento de trajetórias na robótica industrial.*

1 Introdução

No contexto da educação tecnológica, sistemas de controle de manipuladores robóticos envolvem conceitos de modelagem cinemática e integração *software/hardware* (CRAIG, 2005). Em laboratórios de graduação, abordagens baseadas em ensino por poses são adotadas em práticas introdutórias por favorecerem a compreensão empírica sobre limites articulares. Contudo, o uso de *softwares* proprietários esbarra em licenças restritas e problemas de manutenção. Essa barreira restringe a autonomia dos estudantes, impossibilitados de simular códigos fora do ambiente acadêmico.

Diante disso, a adoção dos princípios da cultura *maker* — centrada no “faça-você-mesmo” (*DIY*), na prototipagem rápida e no reaproveitamento de *hardwares* (*retrofit*) — surge como uma alternativa viável para modernizar equipamentos didáticos subutilizados.

Nesse cenário, este trabalho teve como objetivo desenvolver um protótipo de Supervisor de Robótica Industrial Educacional (SuRI-Edu) de código aberto (*open-source*) para o controle de um braço robótico do *kit* Robix, utilizando Python e Arduino Mega.

A proposta busca contornar as restrições de ferramentas comerciais de alto custo (como RoboDK ou MATLAB Robotics Toolbox), fornecendo uma interface gráfica baseada em controle visual via *sliders* e programação por poses, permitindo operação inicial do manipulador sem necessidade de programação textual.

Além disso, diferentemente de plataformas como RoboDK e MATLAB Robotics Toolbox, o SuRI-Edu pode ser executado localmente sem dependência de licenciamento proprietário, favorecendo sua adoção em laboratórios com restrição orçamentária. Este trabalho está estruturado em cinco seções. A Seção 2 apresenta a fundamentação teórica; a Seção 3 os materiais e métodos; a Seção 4 discute os resultados alcançados; por fim, a Seção 5 traz as considerações finais e trabalhos futuros.

2 Fundamentação Teórica

A robótica educacional materializa conceitos teóricos em aplicações práticas, exigindo *softwares* interativos para a experimentação empírica de trajetórias (NIKU, 2011). Sob a ótica da cultura *maker* (“aprender fazendo”), a integração de *hardwares open-source*, como o Arduino (SICILIANO; KHATIB, 2016), e linguagens de alto nível, como Python, contrapõe-se ao modelo de “caixa preta” das soluções comerciais. Essa sinergia tecnológica promove a investigação ativa, permitindo aos discentes manipular diretamente as lógicas de controle e os algoritmos de comunicação (SPONG *et al.*, 2006).

Ademais, enquanto grande parte da literatura *maker* foca na construção mecânica de novos robôs via impressão 3D, esta proposta destaca-se ao aplicar tais princípios no *retrofit* de equipamentos didáticos preexistentes. O desenvolvimento deste supervisor estende a vida útil do patrimônio institucional, unindo sustentabilidade tecnológica e redução de barreiras financeiras para a ampliação do acesso a práticas experimentais de robótica em ambientes educacionais com recursos limitados.

3 Materiais e Métodos

Esta pesquisa aplicada, desenvolvida nos laboratórios de robótica e no FabLab “IFMaker”, estruturou-se em três etapas: integração eletroeletrônica, arquitetura de *software/firmware* e ensaios operacionais na disciplina de Robótica Industrial.

Utilizou-se um braço “Robix” e um Arduino Mega 2560 cedidos pela instituição, cujo acoplamento elétrico foi aprimorado por uma Placa de Circuito Impresso (PCB) customizada. Atuando como shield de expansão acoplado ao microcontrolador, a PCB compactou o sistema e passou a gerenciar a potência e os sinais de controle dos servomotores. O investimento em componentes complementares totalizou R\$ 170,00, incluindo o valor de mercado do Arduino no orçamento para assegurar a replicabilidade da bancada em ambientes educacionais com recursos limitados.

Tabela 1: Componentes e custos do sistema de controle.

Descrição do item	Quant.	Preço uni. (R\$)	Preço total (R\$)
Arduino Mega 2560 R3 + Cabo USB	1	85,00	85,00
Shield de expansão para Servomotores	1	25,00	25,00
Fonte de Alimentação 5V 5A	1	45,00	45,00
Cabos <i>Jumpers</i> (M/F e M/M)	1	15,00	15,00
TOTAL	-	-	170,00

A arquitetura computacional opera em duas camadas. No cliente, a interface gráfica (GUI) foi desenvolvida em Python (VAN ROSSUM; PYTHON SOFTWARE FOUNDATION, 2026), empregando a biblioteca pySerial (LIECHTI, 2026) para viabilizar uma comunicação assíncrona leve e de baixa latência. No nível embarcado, o *firmware* estruturado em C++ na Arduino IDE atua recebendo as *strings* de comando formatadas e acionando as juntas do robô via PCB.

4 Resultados e Discussão

Os ensaios em bancada mostraram comunicação serial estável entre a GUI do SuRI-Edu (Python) e o Arduino a 115200 bps, com latência inferior a 50 ms, suficiente para controle contínuo em tempo real. No modo manual, o problema principal foi o envio excessivo de comandos pelos *sliders* (*serial flooding*), mitigado por um filtro temporal que limita a transmissão a um comando a cada 50 ms; no modo automático, o sistema estima o tempo de curso pelo maior deslocamento angular e só envia o próximo comando após a conclusão do movimento, enquanto o *firmware* usa uma máquina de estados baseada em `millis()` e comandos delimitados para manter a execução previsível e suave. A Figura 1 ilustra a interface, enquanto a Figura 2 exibe o sistema integrado na bancada.

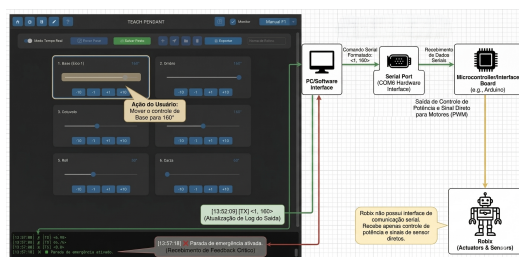


Figura 1: Interface SuRI-Edu e diagrama do sistema. Fonte: Autores (2026).

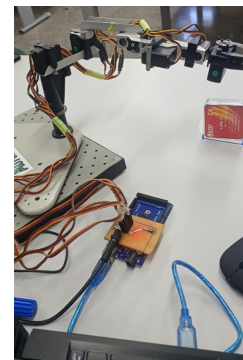


Figura 2: Protótipo em operação na rotina *pick-and-place*. Fonte: Autores (2026).

A validação funcional foi realizada por uma rotina de *pick-and-place* com 19 poses gravadas em arquivo, executada com 100% de sucesso e sem perda de sincronismo entre juntas. Durante os ensaios operacionais, a rotina foi executada repetidamente sem ocorrência de travamentos de comunicação *serial* ou dessincronização entre juntas, evidenciando estabilidade operacional da arquitetura supervisória proposta.

Para manter a operação segura, a aplicação permaneceu responsiva durante a execução e permitiu parada de emergência via teclado/GUI, além de bloquear comandos fora da faixa válida (motor e angle) para reduzir o risco de danos, aspecto relevante por se tratar de um manipulador em malha aberta. Por fim, o SuRI-Edu se apresenta como alternativa *open-source* de baixo custo ($\approx$ R\$ 170,00) e com potencial de replicação, uma vez que requer apenas seis saídas PWM, permitindo inclusive a substituição do Arduino Mega por placas mais acessíveis.

5 Considerações Finais

Este trabalho desenvolveu o supervisor *open-source* SuRI-Edu, viabilizando o *retrofit* de um braço Robix via Python e Arduino. A arquitetura assíncrona mitigou gargalos de latência, garantindo controle fluido em tempo real e êxito nas rotinas de *pick-and-place*. A ausência de prevenção automática de colisões pautou a necessidade futura de implementar limites virtuais e cinemática inversa. Por fim, o projeto demonstra o potencial da cultura *maker* no *retrofit* de laboratórios didáticos, utilizando ferramentas de baixo custo voltadas à experimentação prática em robótica industrial.

Declaração de Uso de IA Generativa

A ferramenta Gemini (Google) foi utilizada como auxílio metodológico na condensação estrutural, revisão ortográfica e tradução do *Abstract*, conforme diretrizes da SBC. Conteúdo validado e de inteira responsabilidade técnica dos autores.

Referências

- ARDUINO (2026). Arduino Mega 2560 Rev3 – Documentation. Disponível em: <https://docs.arduino.cc/hardware/mega-2560>. Acesso em: 29 mar. 2026.
- CRAIG, J. J. (2005). *Introduction to Robotics: Mechanics and Control*. 3. ed. Upper Saddle River: Pearson.
- LIECHTI, C. (2026). *pySerial Documentation 3.5 2026*. Disponível em: <https://pyserial.readthedocs.io/>.
- NIKU, S. B. (2011). *Introduction to Robotics: Analysis, Control, Applications*. 2. ed. Hoboken: John Wiley & Sons.
- SICILIANO, B.; KHATIB, O. (2016). *Springer Handbook of Robotics*. 2. ed. Cham: Springer.
- SPONG, M. W. et al. (2006). *Robot Modeling and Control*. Hoboken: John Wiley.
- VAN ROSSUM, G.; PYTHON SOFTWARE FOUNDATION (2026). *Python Language Reference 3.13.9*. Disponível em: <https://docs.python.org/3/>.