

## NGHIÊN CỨU VÀ ÁP DỤNG PHẦN MỀM OPENPLC EDITOR VÀO GIẢNG DẠY PLC CƠ BẢN

Hoàng Thân\*, Nguyễn Minh Hải, Nguyễn Thị Khánh Tình,  
Lê Huỳnh Lý, Tôn Nữ Thái Hiền

Khoa Điện – Điện tử, Trường Cao đẳng Công nghiệp Huế

\*Email: hthan@hueic.edu.vn

Ngày nhận bài: 17/10/2025; ngày hoàn thành phần biên: 12/11/2025; ngày duyệt đăng: 11/12/2025

### TÓM TẮT

PLC là thiết bị điều khiển cốt lõi trong công nghiệp và tự động hóa, do đó đào tạo lập trình PLC giữ vai trò quan trọng trong các cơ sở đào tạo kỹ thuật và đào tạo nghề. Tuy nhiên, chi phí đầu tư cho PLC, máy tính lập trình và phần mềm bản quyền còn cao, gây hạn chế trong công tác giảng dạy tại nhiều cơ sở đào tạo. OpenPLC Editor là phần mềm mã nguồn mở có khả năng thay thế các môi trường lập trình PLC truyền thống như TIA Portal hay GX Works, cho phép triển khai chương trình trên nhiều nền tảng phần cứng chi phí thấp như Arduino, Raspberry Pi và ESP32. Nhờ dung lượng nhỏ, tính linh hoạt cao và không yêu cầu cấu hình máy tính lớn, OpenPLC đặc biệt phù hợp cho mục đích đào tạo. Trong nghiên cứu này, nhóm tác giả đã xây dựng chương trình điều khiển bằng ngôn ngữ Ladder, mô phỏng trên Proteus 8.13 và thực nghiệm trên Arduino UNO R3 và ESP32 với bài toán điều khiển đèn giao thông. Kết quả cho thấy giải pháp đề xuất có tính khả thi và tiềm năng ứng dụng trong đào tạo và nghiên cứu PLC.

**Từ khóa:** OpenPLC Editor, PLC, Proteus 8.13, Vi điều khiển.

### 1. MỞ ĐẦU

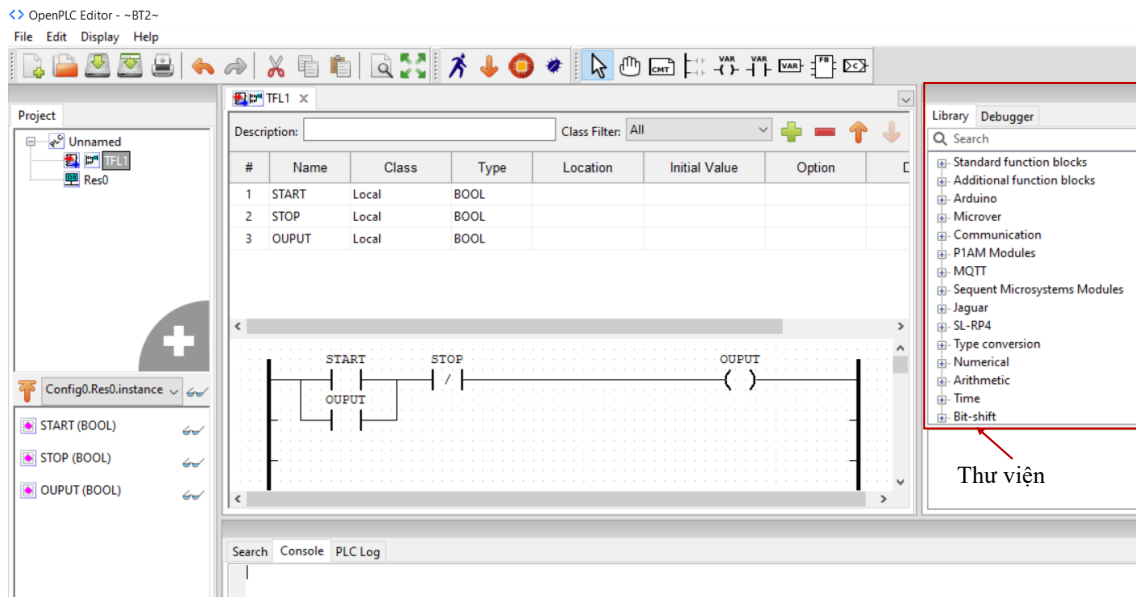
Lập trình điều khiển PLC (Programmable Logic Controller) là môn học điều khiển logic khả trình được thiết kế để giảng dạy trong tất cả các ngành “Khoa học kỹ thuật” từ chương trình đào tạo Đại học đến đào tạo nghề. Trong bối cảnh các trường đào tạo thiếu thiết bị dạy học cũng như phần mềm hỗ trợ vì giá thành và bản quyền thương mại lớn của các thương hiệu PLC như Siemens (Tia Portal), Mitsubishi (GX Developer, GX WORK 2,3)...vì vậy gây khó khăn cho các cơ sở đào tạo đại học nói chung và đào tạo nghề nói riêng trong việc đầu tư thiết bị cũng như học liệu để đào tạo. Đặc

biệt trong giai đoạn đẩy mạnh giáo dục nghề nghiệp từ năm 2025 của Bộ Giáo dục và Đào tạo [1]. Vì vậy, trường Cao Đẳng Công nghiệp Huế (Hueic) và Sở giáo dục và đào tạo Quảng Trị đã hợp tác về đào tạo trung cấp nghề Điện Công Nghiệp cho học sinh THPT tại một số trường THPT ở Quảng trị như trường THPT Nguyễn Hữu Thận, Trường THPT Darkrong... Khi triển khai đào tạo tại đây thì vấn đề về thiết bị đào tạo môn học lập trình PLC cơ bản gặp khó khăn do không thể chuyển thiết bị từ trường CĐCN Huế ra Quảng Trị được. Vì vậy nhóm nghiên cứu đã thực hiện nghiên cứu lập trình PLC trên một số phần mềm mô phỏng như PLC ladder simulator, một app dùng trên điện thoại thông minh và các phần mềm mô phỏng khác dành cho máy tính (PC) như TIA PORTAL, GX WORK, PLC Training... Tuy nhiên các phần mềm chỉ ở dạng mô phỏng trên máy không thu hút được việc học của sinh viên. Vì vậy chúng tôi sử dụng phần mềm OpenPLC Editor [2, 6-8] kết hợp với Arduino để sinh viên thực hành trực tiếp đồng thời mô phỏng trên phần mềm Proteus 8. Sự kết hợp này có các thuận lợi sau: thứ nhất các phần mềm có dung lượng nhỏ phù hợp với cấu hình máy tính ở các trường THPT, thứ 2 học sinh lập trình PLC như lập trình trên các phần mềm chuyên dụng khác, thứ 3 sử dụng phần cứng là Arduino nhỏ gọn, dễ di chuyển đến trường THPT và phù hợp với kinh phí đào tạo ở các trường phổ thông và cuối cùng là thu hút được sự thực hành của học sinh.

## **2. PHƯƠNG PHÁP NGHIÊN CỨU**

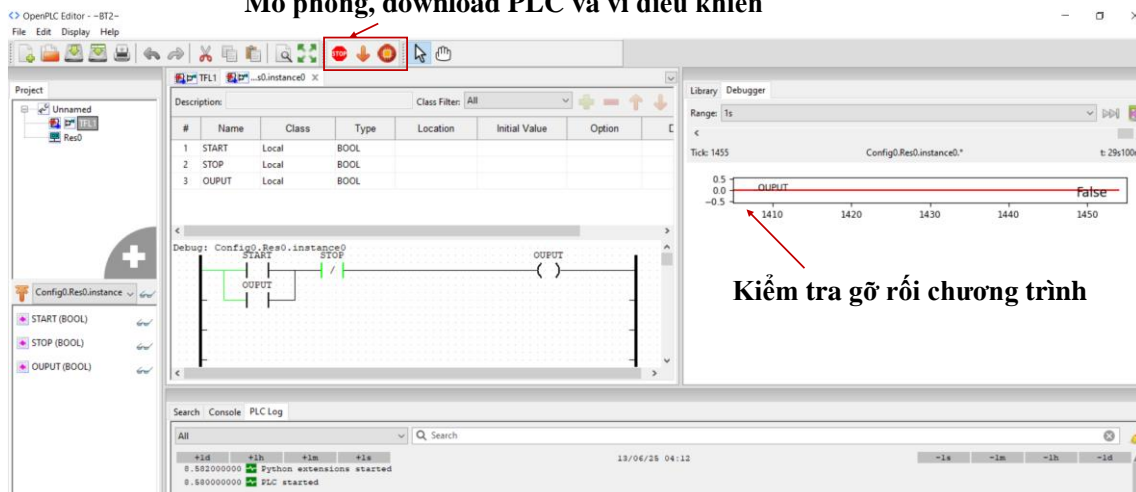
### **2.1. Phần mềm OpenPLC Editor**

Phần mềm OpenPLC Editor là một phần mềm mở có thể tải chương trình vào PLC cũng như các loại vi điều khiển như Arduino, ESP32, Raspberry Pi..., [3] giao diện OpenPLC Editor được mô tả như hình 1, cửa sổ soạn thảo như hình 2. Các dự án của OpenPLC dựa theo tiêu chuẩn IEC 61131-3 [4] nên có thể lập trình như PLC truyền thống.



Hình 1. Giao diện OpenPLC Editor

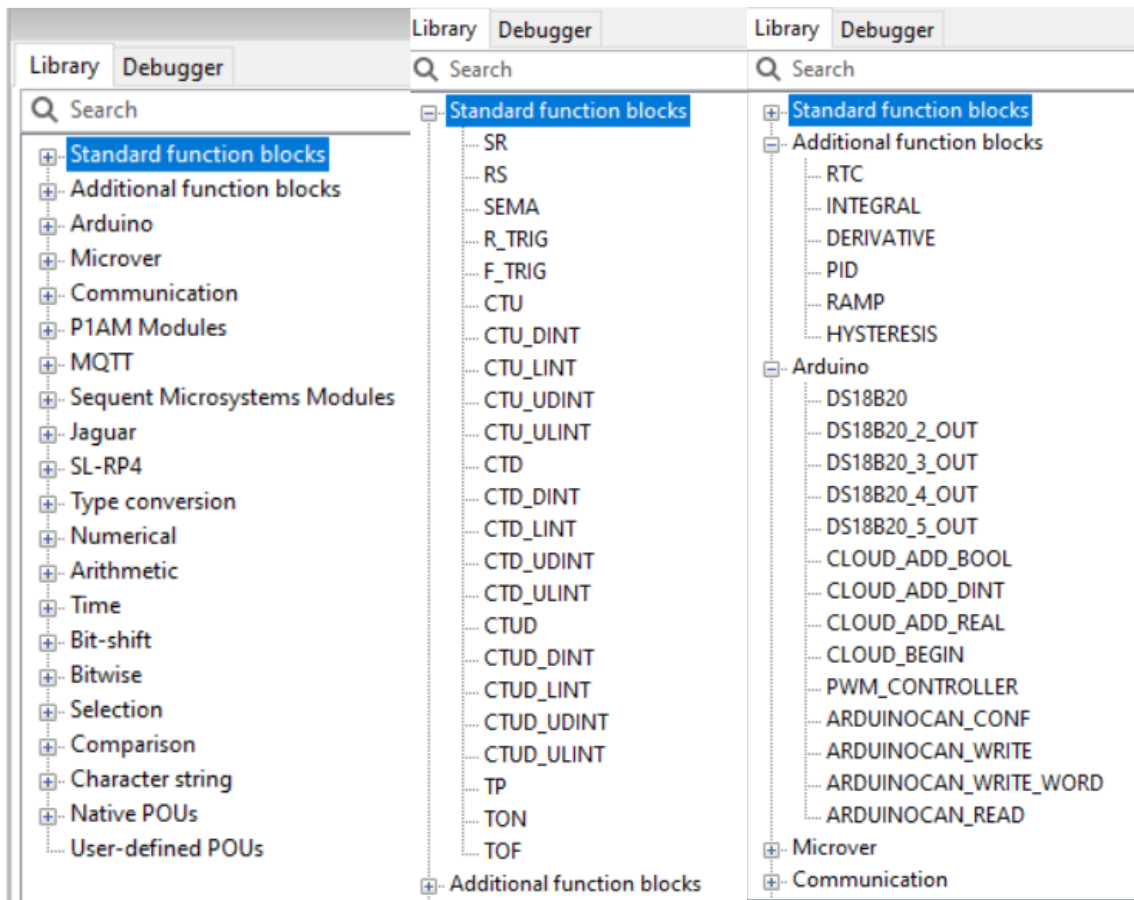
### Mô phỏng, download PLC và vi điều khiển



Hình 2. Cửa sổ soạn thảo của OpenPLC Editor

## 2.2. Thư viện OpenPLC Editor

Phần mềm OpenPLC Editor có đầy đủ thư viện như các phần mềm PLC truyền thống khác được mô tả như hình 3.



Hình 3. Các thư viện chức năng trong OpenPLC

### 2.3. Giao tiếp OpenPLC với các loại vi điều khiển

OpenPLC có thể giao tiếp với vi điều khiển như Arduino, ESP32, Raspberry Pi... dựa trên các địa chỉ vật lý được biểu diễn như Hình 4 [5].

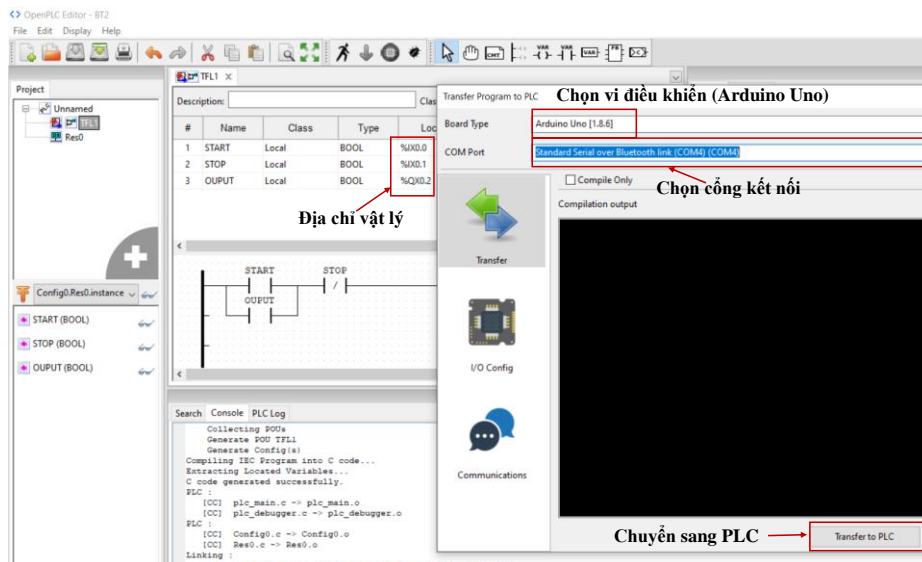
<b>Uno, Leonardo, Nano, Micro, Zero</b>		
Digital In	2, 3, 4, 5, 6	%IX0.0 - %IX0.4
Digital Out	7, 8, 12, 13	%QX0.0 - %QX0.3
Analog In	A0, A1, A2, A3, A4, A5	%IW0 - %IW5
Analog Out	9, 10, 11	%QW0 - %QW2
<b>Esp 32</b>		
Digital In	17, 18, 19, 21, 22, 23, 27, 32, 33	%IX0.0 - %IX0.7, %IX1.0
Digital Out	01, 02, 03, 04, 05, 12, 13, 14, 15, 16	%QX0.0 - %QX0.3
Analog In	34, 35, 36, 39	%IW0 - %IW5
Analog Out	25, 26	%QW0 - %QW2
<b>Raspberry Pi</b>		
Digital In	03, 05, 07, 11, 13, 15, 19, 21 23, 29, 31, 33, 35, 37	%IX0.0 - %IX0.7 %IX1.0 - %IX1.5
Digital Out	08, 10, 16, 18, 22, 24, 26, 32, 36, 38, 40	%QX0.0 - %QX0.7 %QX1.0 - %QX1.2
Analog In	-	-
Analog Out	12	%QW0 (PWM)

Hình 4. Địa chỉ vật lý của các loại vi điều khiển

### 3. KẾT QUẢ VÀ THẢO LUẬN

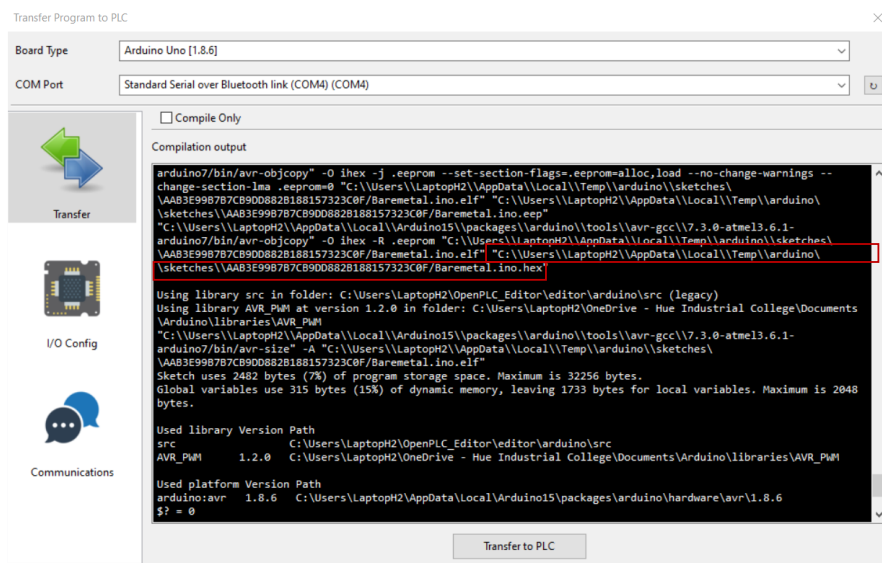
#### 3.1 Mô phỏng với phần mềm Proteus 8

Chương trình PLC Ladder được soạn thảo trên phần mềm OpenPLC được chạy thử trên phần mềm Proteus 8.13. Đầu tiên khai báo địa chỉ vật lý chạy dịch chương trình như hình 5



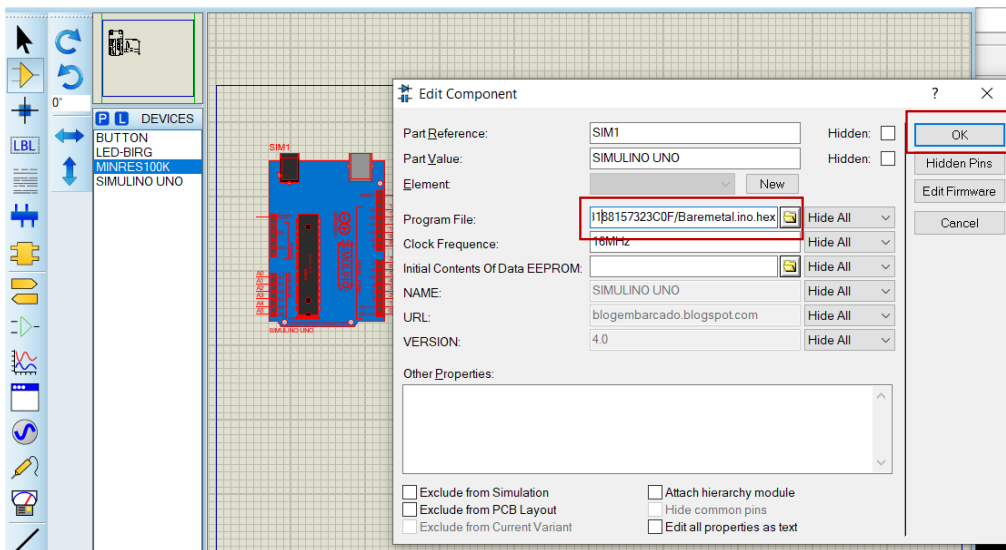
Hình 5. Đăng ký địa chỉ vật lý và dịch chương trình

Thứ hai copy đoạn địa chỉ file HEX như hình 6



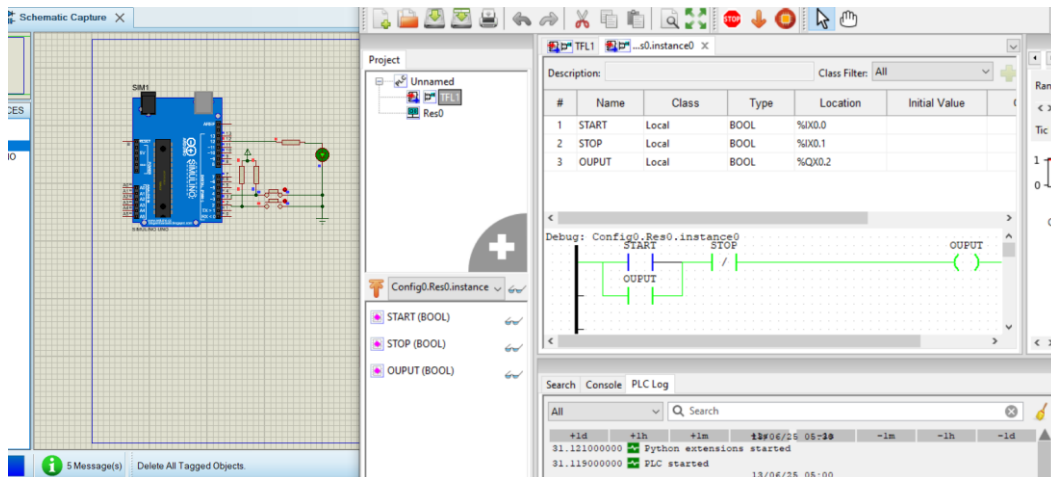
Hình 6. Chương trình dịch đã hoàn thành

Tiếp theo dán địa chỉ vào Arduino trên Proteus 8 như Hình 7

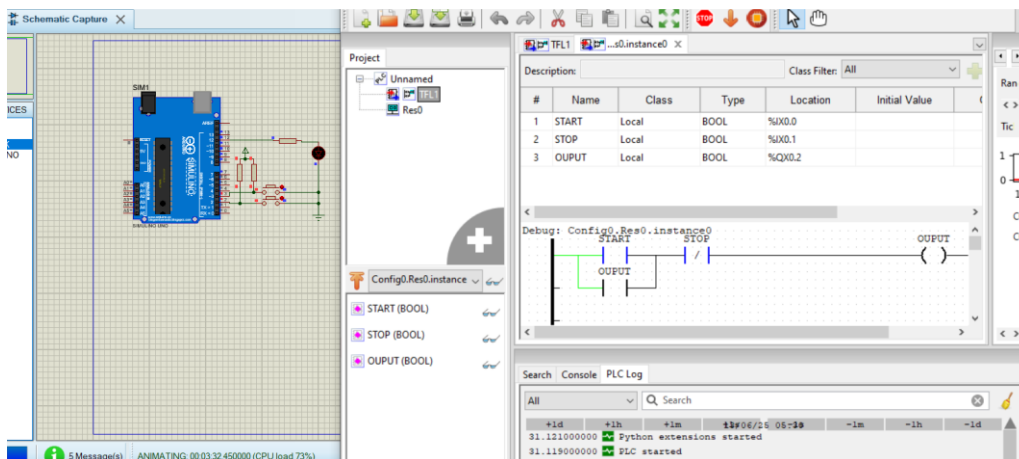


Hình 7. Copy file Hex vào Arduino

Cuối cùng kiểm tra kết quả như hình 8 và hình 9

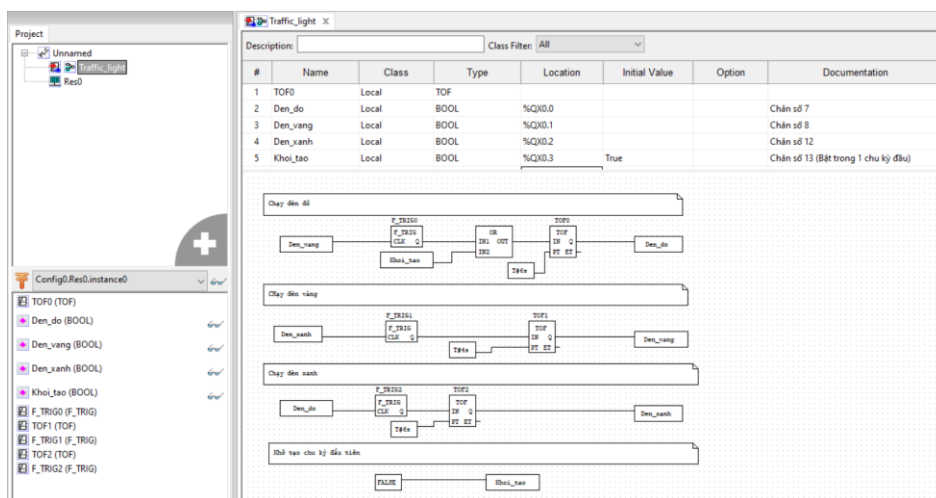


Hình 8. Khởi động chương trình



Hình 9. Dừng chương trình

### 3.2 Kết nối với Arduino Uno

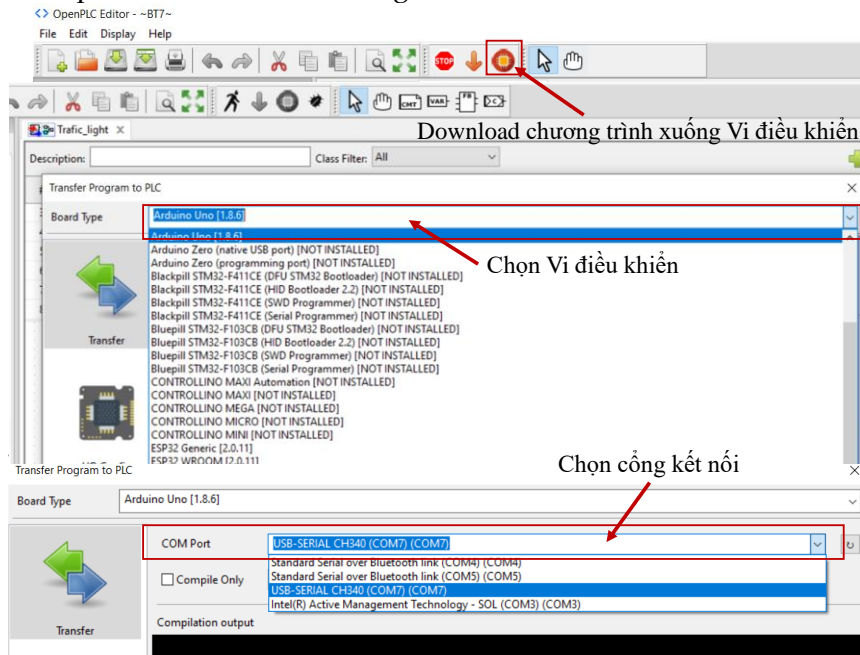


Hình 10. Chương trình đèn giao thông với FBD

Để thực hiện kết nối với Arduino nhóm tác giả thực hiện thêm một bài tập đó là lập trình đèn giao thông đơn giản như hình 10 với ngôn ngữ FBD (Function Block Diagram)

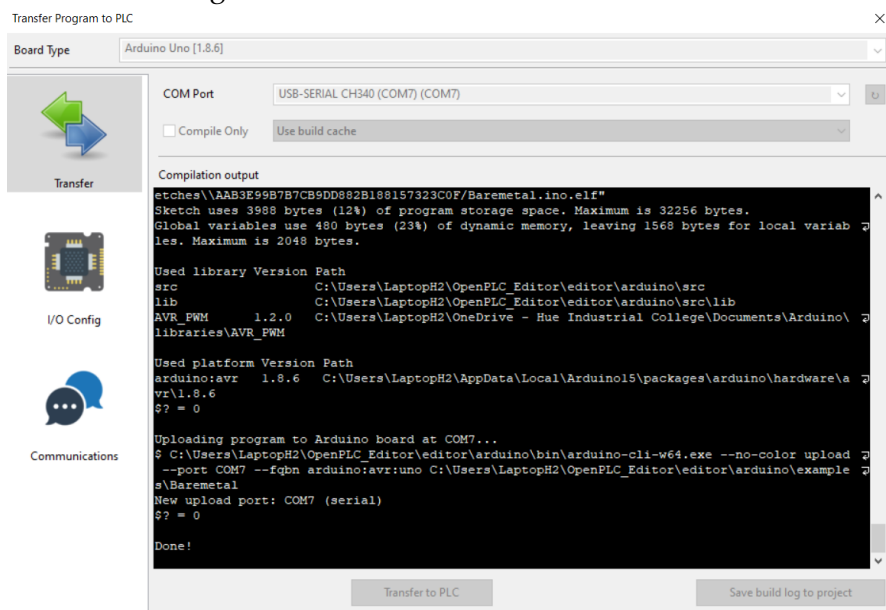
Sau khi kiểm tra độ chính xác của chương trình bằng cách chạy mô phỏng và kiểm tra hoạt động của chương trình.

Bước tiếp theo Download chương trình vào Arduino như hình 11



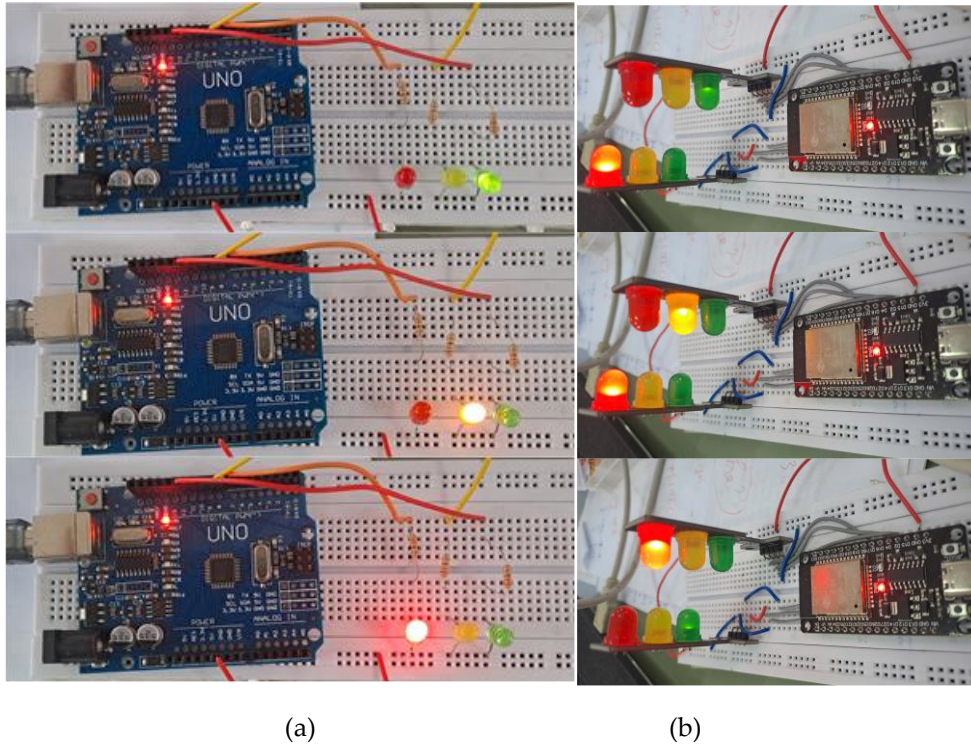
Hình 11. Download chương trình xuống Arduino

Download chương trình hoàn tất như hình 12



Hình 12. Download chương trình xuống Arduino hoàn tất

Kết quả được biểu diễn như hình 13



Hình 13. Chương trình chạy thực tế trên (a) Arduino (b) ESP 32

#### 4. KẾT LUẬN

Nghiên cứu đã mang đến một giải pháp hữu hiệu để đào tạo lập trình PLC đối với những cơ sở đào tạo chưa đủ điều kiện để trang bị các loại PLC truyền thống với giá thành PLC cao, phần mềm có dung lượng lớn nên đòi hỏi hệ thống máy tính có cấu hình cao. Vì vậy giải pháp sử dụng phần mềm mở OpenPLC Editor kết nối với vi điều khiển như Arduino, ESP32, Raspberry PI... đem lại những ưu điểm cũng như các nhược điểm sau: thứ nhất phần mềm mở nên không đòi hỏi bản quyền thương mại, có dung lượng nhỏ nên đòi hỏi máy tính không yêu cầu cấu hình cao; thứ 2 phần mềm này phù hợp với đào tạo PLC cơ bản cho hệ trung học nghề (hệ 9+) và các ngành đào tạo không chuyên; thứ 3 phần mềm có thể được viết với ngôn ngữ Ladder, Block Function... trên các loại vi điều khiển; thứ 4 phần mềm cũng có thể lập trình điều khiển và giám sát SCADA và có đầy đủ thư viện như PLC truyền thống. Tuy nhiên với mức độ phổ biến hiện nay của phần mềm OpenPLC Editor chỉ mới dừng lại ở nghiên cứu và đào tạo PLC cơ bản mà chưa thể ứng rộng rãi trong công nghiệp. Nghiên cứu và áp dụng phần mềm OpenPLC Editor với các vi điều khiển và PLC mitsubishi FX3U để điều khiển động cơ, kết nối cảm biến đồng thời kết hợp với các phần mềm thực tế ảo khác như Factory IO sẽ được nhóm tác giả nghiên cứu thực hiện.

## **TÀI LIỆU THAM KHẢO**

- [1]. Quyết định số 596/QĐ-BGDĐT về “Quy định chức năng, nhiệm vụ, quyền hạn và cơ cấu tổ chức của Cục Giáo dục nghề nghiệp và Giáo dục thường xuyên” ngày 03/03/2025.
- [2]. OpenPLC.(2023). Openplc overview, <https://autonomylogic.com/docs/openplc-overview/>
- [3]. Dr. Don Wilcher (2023), “Embedded electronics is a subfield of electronics that can unite the power of programming with the power of electronics”, Embedded Controls Development with OpenPLC, DesignNews, CEC.
- [4]. "IEC 61131-3:2013". International Electrotechnical Commission (IEC). Retrieved 11 September 2021.
- [5]. OpenPLC.(2023). Openplc overview, <https://autonomylogic.com/docs/2-4-physical-addressing/>
- [6]. E. Vishnu Easwaran, Rajesh Kushalkar, Nivedita Tigadi, Kannan M. Moudgalya, Akshav Chipkar, Alois Zoitl (2018), “Programmable Logic Controller: Open Source Hardware and Software for Massive Training”, IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society, ISBN:978-1-5090-6685-8
- [7]. Erik Kučera, Oto Haffner, Peter Drahoš and Ján Cigánek, “Educational Case Studies for Pilot Engineer 4.0 Programme: Monitoring and Control of Discrete-Event Systems Using OPC UaandCloudApplications”, Appl. Sci. 2022, 12, 8802. <https://doi.org/10.3390/app12178802>.
- [8]. Dr. Hugh Jack (2023), “Teaching Industrial Control with Open-Source Software”, 2023 Annual Conference and Exposition, June 25-28, 2023

## A STUDY ON THE USE OF OPENPLC EDITOR SOFTWARE IN BASIC PLC INSTRUCTION

Hoang Than\*, Nguyen Minh Hai, Nguyen Thi Khanh Tinh,  
Le Huynh Ly, Ton Nu Thai Hien

Hue Industrial College

\*Email: hthan@hueic.edu.vn

### ABSTRACT

Programmable Logic Controllers (PLCs) are core control devices in industrial and automation systems; therefore, PLC programming education plays a crucial role in technical and vocational training institutions. However, the high investment costs associated with PLC hardware, programming computers, and licensed software pose significant challenges to teaching activities at many educational institutions. OpenPLC Editor, an open-source software platform, provides an effective alternative to traditional PLC programming environments such as TIA Portal and GX Works, enabling program deployment on various low-cost hardware platforms including Arduino, Raspberry Pi, and ESP32. Owing to its small footprint, high flexibility, and low hardware requirements, OpenPLC is particularly suitable for educational purposes. In this study, the authors developed a control program using the Ladder language, performed simulations in Proteus 8.13, and conducted experimental implementations on Arduino UNO R3 and ESP32 for a traffic light control application. The results demonstrate that the proposed solution is feasible and shows strong potential for application in PLC education and research.

**Keywords:** Microcontroller, OpenPLC Editor, PLC, Proteus 8.13.