WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

IoT Enabled Programmable Logic Controller for Automation and Monitoring Design and Application to a Water Bottle Filling System

Mrs. Seema Wasnik[1],Vansh Nagare[2], Prerna Kumley[3], Swayam Wanjari[4],Sujal Dupare[5]

Department of Electronics and Communication Engineering
Priyadarshini College of Engineering Nagpur, Maharashtra, India

ABSTRACT-

Programmable logic controllers (PLCs) are widely used in automation to receive signals from input devices and control output devices according to programmed logic. This paper presents the design of an internet-of-things enabled programmable logic controller prototype based on an ESP32 development board and its role in a conveyor-based demonstration of a water bottle filling process. The design of the hardware prototype consists of an eight-channel digital input module, an eight-channel relay output module, 24 VDC power supply, buck converter, status LEDs, and an RS-485 module. Ladder Logic for the model is created in the OpenPLC environment; the details of the run time and target of the ESP32 have not yet been verified. A custom dashboard is created using React / Vite, Node.js, Express, and Socket.IO for communication between the software. The intended filling sequence detects a bottle, stops the conveyor, operates the filling mechanism, and resumes conveyance.

Index Terms- ESP32, programmable logic controller, OpenPLC, Ladder Logic, IoT, water bottle filling, conveyor automation, digital input and output.

I.INTRODUCTION

Automation systems are built using sensors, the control logic and the actuators, which are able to carry out repeatable operations. Programmable Logic Controller (PLC) monitors the input condition, and controls a piece of equipment based on a pre-defined sequence. A small controller can offer a practical platform for automation at small and educational scales.

The project shows an ESP32 PLC prototype based on the OpenPLC framework to create Ladder Logic programs. The final target device is capable of sensing digital input signals and controlling relay-based outputs. The communication from backend to the controller is still unconfirmed. Web-based dashboard for this project is being built using React with Vite, with Node.js and Express supporting, and Socket.IO enabling event-based messages between components.

Introduction The conveyor-based water bottle filling system has been chosen as the application for this demonstration. The process that is planned is to transfer the bottle to a filling position, sense the bottle, turn off the conveyor, turn on the filling process and then turn the conveyor back on after the filling has finished. Therefore the bottle filling process is the "application" that is to be demonstrated by the PLC controller.

II. PROPOSED SYSTEM

A. System Architecture

The proposed system consists of a digital input section, an ESP32 controller, a relay output section, a power supply section, and a software monitoring interface. The input module will be used to accept the field signals in compatibility. The response to the signals from the operator or sensors will be defined by the Ladder Logic program. Relay outputs will be used as switches on the conveyor drive and filling actuator.

Start-Stop/ Bottle Sensor Inputs
↓
8-Channel Input Module
↓
ESP32 + PLC Logic
↓
Relay / Driver Interface
↓
Conveyor Motor + Filling Actuator
Dashboard Backend Controller (communication path to be checked).

B. Filling Sequence for Water Bottles

The automatic cycle for the automation will initiate when the operator activates the system. The conveyor will then start and move bottles to the filling station. When the bottle-presence sensor senses a bottle is at a specified location the control logic will stop the conveyor, engage the filling actuator for a preset period or until the fill level is detected at the appropriate position, disengage the filling actuator and start the conveyor. The actual sequence will be determined by the current Ladder Logic, the sensors used and the filling method used.

C. Hardware Design

The proposed hardware comprises an ESP32 dev board, 8 channel digital input, 8 channel relay, 24V DC power supply, buck converter, RS-485 module, and input and output indicator LEDs. A BC547 transistor stage is being considered to resolve an input-interface polarity problem found in the development phase. Final circuit polarity and signal levels need to be checked on the hardware that is being built.

The RS-485 module is on the design plan, but a working communication link should only be claimed after testing. Conveyor motor, filling pump / valve must match output interface and power specifications.

D. Software Design

OpenPLC is utilized to develop the Ladder Logic, but the precise runtime and ESP32 target configuration are not yet finalized. The dashboard frontend is implemented in React and Vite. Node.js and Express are used to implement the backend application; the event communication on the software side utilizes Socket.IO. The physical connection to the controller and command protocol need to be validated before the dashboard is described as controlling hardware end to end.

III. METHODOLOGY

A. Hardware Assembly

Connect the following components as per the final layout; controller, input module, relay module, buck converter, sensor, conveyor drive and filling actuator. Prior to operation, check the supply voltage and converter output, input polarity, grounding arrangements and interface signal levels. Keep the tank and tubing clear of the exposed electrical connections.

B. Ladder Logic and Control Sequence

The control sequence to be implemented uses start and stop commands, the bottle-presence signal, the conveyor output and the filling output. When a bottle reaches the filling position the logic moves from the conveyor running state to the filling state. When this filling condition is complete the logic will disable filling and restore the conveyor to the running state. The precise timing and transitions must match the actual program and available sensors.

C. Input and Output Verification

Test the input module and BC547 interface for valid ON/OFF levels and correct logical polarity. Test each relay output independently before hooking the conveyor motor or filling actuator. The loads connected must be within the electrical ratings of the relay contacts, driver circuitry and power supply.

D. Dashboard Integration

Ensure that the frontend-to-backend and backend-to-controller connections are tested separately. Tests should establish whether operator commands reach the controller and whether changes in physical input/output states are reflected on the dashboard. Passing the user interface does not necessarily mean that hardware has been successfully integrated.

IV. HARDWARE AND SOFTWARE

Component / Tool	Intended role
ESP32 development board	Main controller platform
8-channel input module	Sensor and operator-input interfacing
8-channel relay module	Output switching interface
24 V DC supply	Power supply as specified by the circuit
Buck converter	DC voltage conversion
Bottle sensor	Bottle detection at filling position
Conveyor and motor	Bottle transport
Filling pump/valve and nozzle	Liquid dispensing
Input/output LEDs	Visual status indication
BC547 transistor stage	input polarity interface in progress
RS-485 module	Wired communication if implemented
OpenPLC / Ladder Logic	Control-program creation
React + Vite	Frontend of dashboard
Node.js + Express + Socket.IO	Back-end and software communication

V. IMPLEMENTATION STATUS

As of this writing, the hardware is in the process of being assembled. The entire physical input/output testing and end-to-end testing have not been proven. The input-interface polarity, OpenPLC runtime configuration, backend to controller communication, and the conveyor and filling actuator's applicability to output hardware is to be confirmed.

No response time, fill accuracy, throughput, or reliability results are claimed. After assembly, the input channels, relay outputs, bottle detection, filling sequence, and dashboard communication should be tested and the actual observations recorded.

VI. LIMITATIONS AND FUTURE WORKS

Complete prototype assembly and integrated testing. Input interface polarity and voltage will need to be validated and the specific run time and communication path will need to be confirmed. RS-485 must only be said to be working if a full link has been tested.

Future work to be done is to test all I/O channels, test the sequence of bottling from start to finish repeatedly to ensure the sequence works reliably, test the cycle time and accuracy of filling, observe the behavior when communication is lost. Safety-critical stopping should be controlled by independent hardware and not only dashboard software.

VII. CONCLUSION

This paper covers the design of an ESP32-based PLC prototype that can be used for automation and monitoring. Intended demonstration application: a conveyor-based water bottle filling line. Proposed setup: combines Ladder Logic development using OpenPLC, digital IO and relay output, power conversion, bottle detection, conveyor drive, filling station, and a custom dashboard design created in React & Vite with a Node.js/Express backend and Socket.IO.

Completion of hardware build and integration test is in progress and the entire control sequence and communication path has yet to be validated. It is as yet not complete to test all the complete control sequence and communication path. Draw performance conclusions only after actual observations are recorded after physical tests.

ACKNOWLEDGMENT

The authors are grateful to the project guide Mrs. Seema Wasnik and faculty of Department of Electronics & Communication Engineering, Priyadarshini College of Engineering, Nagpur for their guidance and academic support.

REFERENCES

[1] J. N. Kuriti, S. Yarragula, L. Kumar, T. Gunara, T. K. Kanala, and P. S. Karanam, "PLC Development with Arduino Uno," International Journal for Research in Applied Science & Engineering Technology (IJRASET), vol. 13, no. 3, Mar. 2025.

[2] L. I. Minchala et al., "An Approach to Industrial Automation Based on Low-Cost Embedded Platforms and Open Software," Applied Sciences, vol. 10, no. 4696, Jul. 2020, doi: 10.3390/app10144696.

[3] S. M. Rasal and V. D. Chaudhari, "Temperature Sensor Interfacing with Embedded PLC," International Journal of Application or Innovation in Engineering & Management (IJAIEM), vol. 3, no. 2, Feb. 2014.