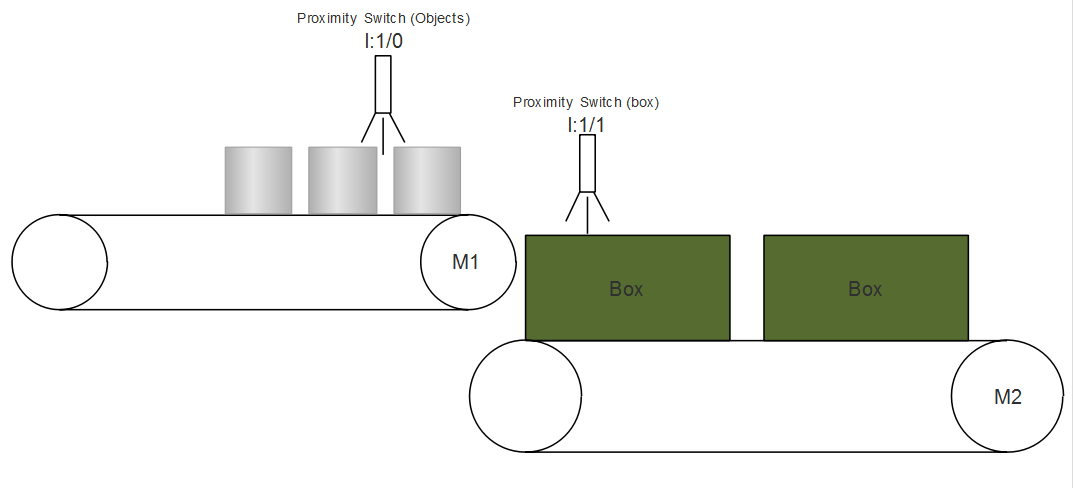
ECT286 Final Project Report

**Conveyor Belt Using PLC**

*Objects are moving on Conveyor Belt 1. When an empty box shows up, the conveyor starts, and five pieces get packed inside the box. Once it's full, the box gets carried to the storage area on Conveyor Belt 2. I set this up in LogixPro software using Ladder Logic programming.*



Jason Palazzo   
**Automation and Control 60681**

**Course Number: ECT2861**

Professor: **Eric Addeo**

DeVry University

College of Engineering and Information Sciences



2024

Executive Summary

This project delineates the automation process for controlling two conveyor belts utilizing a programmable logic controller (PLC). The primary objective is to efficiently manage the flow of objects through both conveyor systems, ensuring timely processing when specific conditions are fulfilled.

The project is organized around two principal routines: \*\*RUNG000\*\*, which governs the initiation and cessation of the entire system, and \*\*RUNG001\*\*, which manages the operation of Conveyor Belt 1's motor (M1). The operation is activated by a Box Detection proximity switch that engages the motor upon the detection of an empty box (I:1/1).

Upon commencement of the process, each identified object is recorded by an Object Detector (I:1/0), thereby incrementing Counter C5:0 until it reaches a preset value of 6, indicating that five objects have been detected. This triggers the Counter Done bit (C5:0/DN) and activates Timer T4:0 for a duration of two seconds, permitting the objects to descend into the box.

When C5:0/DN is activated, the Counter is reset immediately, generating a one-shot pulse that latches the Latching Motor 1 bit, ensuring consistent operation. Once the counter reaches its designated threshold, the motor for Conveyor Belt 2 (M2) is engaged, facilitating the movement of objects until an empty box is once again detected, at which point the conveyor ceases operation.

This cyclical process ensures efficient handling of objects, thereby maintaining a streamlined workflow. It is noteworthy that the counter's preset value of 6 has been strategically determined; upon reset, the accumulator is set to 1 due to the latching bit, ensuring continuity in operation.

The system relies on a variety of inputs and outputs, including start/stop inputs, latching bits for motor activation, and detection inputs, to provide precise control over conveyor operations. In summary, this automation project significantly enhances productivity and reduces the necessity for manual intervention in the object processing workflow.

Table of Contents

[Introduction 1](#_Toc13830381)

[Project Description 1](#_Toc13830382)

[Project Objectives 1](#_Toc13830383)

[List of Material 1](#_Toc13830384)

[Project Design 1](#_Toc13830385)

[Test Results 1](#_Toc13830386)

[Lessons Learned 2](#_Toc13830387)

[Author's Biography 2](#_Toc13830388)

[Works Cited 2](#_Toc13830389)

[Work Consulted 2](#_Toc13830390)

[Appendix A 2](#_Toc13830391)

[Appendix B 2](#_Toc13830392)

# Introduction

The project is all about making the movement of boxes easier with two conveyor belts in a processing facility. The idea is to speed up the workflow by using sensors that can tell when boxes are around and when they need to be moved.

When the system is turned on, it starts pushing boxes along the first conveyor belt. A counter keeps track of how many boxes pass through. Once it hits five, the system takes a short break to make sure those boxes can be processed properly. After that, the second conveyor belt kicks in to keep the boxes moving to the next stage.

This whole process keeps repeating, so boxes are handled smoothly without any manual work. This automation not only speeds things up but also keeps everything flowing nicely, which boosts overall productivity.

# Project Description

The project involves creating an automated system for managing the movement of boxes on two conveyor belts using a programmable logic controller (PLC). The system is designed to detect when boxes are present, count them, and control the operation of the conveyor belts accordingly.

**Project Overview:** The automation process begins with the activation of Conveyor Belt 1, which moves boxes along the belt. A sensor detects when an empty box is in place, triggering the motor to start. As boxes pass by, a counter increments for each detected object. Once the counter reaches five (indicating that five boxes have been detected), it activates a timer for two seconds, allowing time for the boxes to fall into a designated area. After this, Conveyor Belt 2 is engaged to move the boxes further along the process. The system continues to repeat this sequence, ensuring efficient handling of boxes without the need for manual operation.

**Inputs:**

1. **I:1/14** - Start button (Input)
2. **I:1/15** - Stop button (Input)
3. **I:1/0** - Object detection sensor (Input)
4. **I:1/1** - Box detection sensor (Input)

**Outputs:**

1. **O:2/0** - Conveyor Belt 1 motor (Output)
2. **O:2/1** - Conveyor Belt 2 motor (Output)

**Counters and Timers:**

1. **C5:0** - Object counter (Counter)
2. **T4:0** - Timer for conveyor operation (Timer)

**Hardware Requirements:**

* Programmable Logic Controller (PLC)
* Proximity sensors for object and box detection
* Motors for Conveyor Belts 1 and 2
* Power supply for sensors and motors
* Control panel with start and stop buttons
* Wiring and connectors for electrical connections

**Software Requirements:**

* PLC programming software to develop and upload the control logic (e.g., ladder logic)
* Simulation software (optional) for testing the control logic before deployment

This combination of inputs, outputs, hardware, and software enables the automated system to function effectively, enhancing productivity and efficiency in the handling of boxes.

# Project Objectives

# The objectives of the project are centered around enhancing the efficiency and reliability of box handling in a processing environment through automation. The specific objectives include:

# **Automation of Box Movement:** Implement a system that automates the movement of boxes along two conveyor belts, reducing reliance on manual handling and minimizing human error.

# **Efficient Detection and Counting:** Utilize proximity sensors to accurately detect the presence of boxes and count them, ensuring that processing occurs only when the correct number of boxes is present.

# **Improved Workflow:** Establish a seamless workflow where boxes are automatically moved to designated areas after processing, thereby enhancing operational efficiency and throughput.

# **Reduced Processing Time:** Optimize the time taken for boxes to transition between conveyor belts, ensuring a smooth operation that adheres to production schedules.

# **User-Friendly Control Interface:** Develop an intuitive control panel that allows operators to easily start and stop the system as needed, facilitating straightforward management of the automated process.

# The project will accomplish these objectives by integrating hardware and software components to create a cohesive automation system. The extent of achievement will be measured by improvements in processing speed, reduction in manual labor, and increased accuracy in box handling.

# **Project Deliverables:**

# **Automated Control System:** A fully functional PLC-based control system that manages the operation of both conveyor belts, including programming logic to handle detection, counting, and motor control.

# **Documentation:** Comprehensive documentation detailing the system design, installation procedures, operational guidelines, and troubleshooting instructions to support future maintenance and use.

# **Testing and Validation Report:** A report documenting the testing process, results, and validation of the system to ensure it meets operational requirements and specifications.

# **User Training:** Training sessions for operators and maintenance personnel to ensure they are familiar with the system's operation and can effectively manage it.

# **Maintenance Plan:** A structured maintenance plan outlining routine checks, potential issues, and recommended procedures to ensure the longevity and reliability of the system.

# By delivering these objectives and outputs, the project aims to provide a robust solution that enhances productivity, accuracy, and overall operational efficiency in the handling of boxes within the facility.

# List of Material

If the project is physically built, the following list of hardware and software will be required to ensure successful implementation of the automated box handling system:

**Hardware Requirements:**

1. **Programmable Logic Controller (PLC):**
   * A suitable PLC with enough input and output channels to handle the sensors and motors.
2. **Proximity Sensors:**
   * Object detection sensor (e.g., I:1/0).
   * Box detection sensor (e.g., I:1/1).
3. **Conveyor Belt Motors:**
   * Motor for Conveyor Belt 1 (M1).
   * Motor for Conveyor Belt 2 (M2).
4. **Control Panel:**
   * Start and stop buttons (e.g., I:1/14 and I:1/15).
   * Indicator lights to show system status (optional).
5. **Power Supply:**
   * Adequate power supply units for PLC, sensors, and motors.
6. **Wiring and Connectors:**
   * Electrical wiring for connections between the PLC, sensors, and motors.
   * Connectors for secure and reliable connections.
7. **Mounting Hardware:**
   * Brackets, supports, and other mounting components to secure the conveyor belts and sensors.
8. **Conveyor Belts:**
   * Two conveyor belts of appropriate size and specifications for the application.

**Software Requirements:**

1. **PLC Programming Software:**
   * Software to develop and upload control logic (e.g., RSLogix, TIA Portal, or similar).
2. **Simulation Software (optional):**
   * Software for simulating the PLC program before deployment to ensure logic accuracy and operational efficiency.
3. **Documentation Tools:**
   * Software for creating operation manuals and maintenance documentation (e.g., Microsoft Word, Google Docs).
4. **Data Logging Software (optional):**
   * Software for monitoring and logging system performance data for analysis and optimization.

By incorporating these hardware and software components, the automated box handling system will function effectively, enhancing operational efficiency and reliability in the processing environment.

# Project Design

A computer screen shot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A computer screen shot of a computer

Description automatically generated

A computer screen shot of a computer

Description automatically generated

**Process Explanation**

1. Initialization: The system starts up and waits for the start button to be pressed.

2. Box Detection: The object detection sensor checks for the presence of boxes.

3. Counting: Each detected box increments the counter by one.

4. Timer Activation: After counting five boxes, a timer activates for two seconds.

5. Movement to Conveyor Belt 2: Once the timer expires, the system moves to Conveyor Belt 2.

# Test Results

A screenshot of a computer

Description automatically generated

A computer screen shot of a computer

Description automatically generated

# Lessons Learned

When looking over the project, there are a few things that could be done better or approached differently:

1. More Sensors: Adding extra sensors could make the system more reliable. For example, putting in more box detection sensors would help catch any boxes that get missed, especially if they're not lined up right or stacked.

2. Better Control Logic: Instead of just counting boxes, a smarter control system could be used. This could involve prioritizing different box types and sizes or even using machine learning for better maintenance predictions and smoother operations.

3. Simpler User Interface: It would be great to have a more user-friendly interface on the control panel, maybe even with a touchscreen. This kind of interface could show real-time feedback, alerts, and diagnostics, making it easier for operators to keep everything running.

4. Data Logging: Bringing in data logging from the start could provide useful insights into how the system is performing and help spot any slowdowns or issues over time.

5. Safety Features: Adding more safety measures, like emergency stop buttons and guards around the conveyor belts, would be a smart move to keep operators and equipment safe.

6. Maintenance Alerts: Setting up a system that monitors conveyor belt usage and notifies operators when maintenance is needed could help keep everything running smoothly and extend the equipment's life.

7. Testing and Simulation: Doing more testing and simulations before rolling everything out could help catch any potential issues with the control logic or overall design, reducing the chance of problems later on.

8. Planning for Growth: Designing the system to be scalable means it can be upgraded more easily down the line, which is great for handling increased production demands without needing a complete overhaul.

Making these changes could lead to a more solid, efficient, and user-friendly automated box movement system, which would ultimately boost productivity and safety.

# Author's Biography



With a passion for engineering, problem-solving, and innovative solutions, I bring a diverse skill set and professional background to tackle complex engineering challenges. As a proactive and detail-oriented team player with excellent communication, I am driven to create impactful engineering solutions. Explore my portfolio to discover the depth of my capabilities.

# .Works Cited

**Title:** Programmable Logic Controllers: Principles and Applications  
**Author:** John W. Webb, Ronald A. Reis  
**Publisher:** Prentice Hall  
**Year Published:** 2004  
**Pages Reviewed:** 150-200

[2] **Title:** Industrial Automation: Hands On  
**Author:** Frank Lamb  
**Publisher:** McGraw-Hill Education  
**Year Published:** 2013  
**Pages Reviewed:** 75-120

[3] **Title:** Understanding PLCs and Automation Systems  
**URL:** <https://www.automation.com/automation-news/article/understanding-plcs-and-automation-systems>

# 

# Work Consulted

* Alan Brown, Industry Expert, for best practices in conveyor system automation.
* Emily White, Technical Trainer, for user interface and operator training recommendations.

# Appendix A

A computer screen shot of a program

Description automatically generated

Appendix B

[Julia Valentine](https://devryu.instructure.com/courses/115456/users/228810)

Dec 12 1:06pm| Last reply Dec 15 9:27pm

Manage Discussion by Julia Valentine

PLCs use different types of programming languages, each with its own strengths depending on the application. Ladder Logic is popular because it looks like electrical relay circuits, making it easy to use and troubleshoot in industrial settings. Function Block Diagram (FBD) is great for process control since it uses a graphical interface to handle things like PID loops and complex systems. Structured Text (ST) is more like traditional programming languages, so it works well for advanced calculations and data-heavy tasks. Sequential Function Chart (SFC) is perfect for processes that need to happen in specific steps, like batch processing. Instruction List (IL) is a low-level language that’s less common now but can still be helpful in simpler systems. The best language really depends on what you’re working on—Ladder Logic for standard industrial control, FBD for processing tasks, and ST if you need more flexibility.

[**Jason Palazzo**](https://devryu.instructure.com/courses/115456/users/245108)

Dec 15 8:54pm

Manage Discussion by Jason Palazzo

Reply from Jason Palazzo

Hello Julia,

You've given a solid overview of the different programming languages used in PLCs and where they're typically applied. The way you explained Ladder Logic as a user-friendly choice because it looks like electrical relay circuits is super helpful—it really shows why it’s a go-to in industrial settings. Your point about Function Block Diagram (FBD) being great for process control is right on; its visual layout makes handling complex systems easier for operators. Also, mentioning Structured Text (ST) as similar to traditional programming languages really highlights its value for advanced calculations, which is essential in data-heavy scenarios. Your discussion of Sequential Function Chart (SFC) being suited for stepwise processes adds depth, especially for those working with batch processing. And your note on Instruction List (IL) being less common but still useful in simpler systems is a good reminder that even the lesser-known languages have their use. Your writing clearly showcases the strengths of each language, making it easy to see how the right choice should fit the specific tasks at hand. Great job!