PROJECT ABSTRACT

Parcel Sorting Conveyor using Barcode Scanner

A model-scale parcel sorting conveyor: a motorized belt moves parcels under a barcode scanner, and servo-driven diverter gates route each parcel to its lane from a destination table, with a throughput dashboard, delivered as a built-to-order project with fabrication drawings, Arduino firmware, report, PPT and viva kit.

01. Title

Parcel Sorting Conveyor using Barcode Scanner

02. Introduction / Background

Distribution hubs sort thousands of parcels a day by destination, and the core of that automation is a loop a student can build on a bench: move, identify, divert. Manual sorting does not scale, fixed diverters cannot change destinations, and a sorter that cannot read a label is just a conveyor. This project builds the working loop at model scale: a belt conveyor driven by a DC gear motor with speed control, a serial barcode scanner module mounted over the belt, a through-beam photoelectric sensor that triggers a scan window as each parcel arrives, and two servo-driven diverter gates routing parcels to three lanes (two sort lanes plus an error lane). The Arduino matches each decoded barcode against a destination table and fires the right gate at the right time. A dashboard streams throughput, per-lane counts, the scan log and unread-barcode events.

03. Problem Statement

Sorting automation is usually taught as a diagram — conveyor, scanner, diverter — without the timing and sensing problems that make or break the real thing: when to trigger the scan, how to align the gate firing with the parcel position, and what to do with labels that cannot be read. The problem this project addresses is a working bench-scale sorter that confronts all three: a photoelectric arrival trigger opens the scan window, the destination table maps codes to lanes, unread or unknown codes go to a dedicated error lane, and a documented minimum-gap procedure defines the reliable operating envelope. The result is a sorter students can measure, not just admire.

04. Objectives

- Build a speed-controlled belt conveyor with a barcode scanner arch.

- Trigger scans with a photoelectric arrival sensor aligned to parcel position.
- Route parcels to three lanes with two servo-driven diverter gates.
- Implement a configurable destination table with error-lane handling for unread codes.
- Provide a dashboard with throughput, per-lane counts and scan log.

05. Proposed Methodology

Parcels are placed on the belt driven by a DC gear motor with PWM speed control. The through-beam photoelectric sensor detects each arriving parcel and triggers the serial barcode scanner's read window. The decoded code is sent to the Arduino, which looks it up in the destination table: lane A, lane B, or the error lane for unknown or unread codes. At the gate position the Arduino fires the matching servo diverter, pushing the parcel into its lane; the gate then returns to neutral. Every scan, sort decision and error streams to the dashboard. The minimum-gap test procedure measures the smallest parcel spacing the rig sorts reliably.

06. System Architecture / Working

The mechanical tier is the belt conveyor with motor, scanner arch, arrival sensor, diverter gates and three lanes. The control tier is the Arduino: scan triggering, serial decoding, destination lookup, gate timing and dashboard streaming. The dashboard tier shows throughput, per-lane counts, scan log, error events and the configuration view (destination table, belt speed). Data flows from parcel arrival to scan trigger to decode to lane decision to gate actuation to dashboard logging — a closed identify-and-divert loop.

07. Hardware / Software Requirements

Hardware: Arduino (Uno/Mega), DC gear motor with motor driver, belt conveyor frame, serial barcode scanner module, through-beam photoelectric sensor, 2x servo motors with diverter gates, 12 V supply for the motor, 5 V for logic/servos/scanner. Software: Arduino firmware, web dashboard (HTML/CSS/JS), destination-table configuration guide, minimum-gap test procedure.

08. Expected Outcomes

- A working model sorter that scans and diverts parcels to the correct lanes.
- A destination table editable without reflashing the Arduino.
- A live dashboard with throughput, lane counts and scan log.
- Error-lane behavior demonstrated with unread labels.
- A measured minimum parcel gap defining the reliable operating envelope.

09. Applications

- Teaching industrial automation: sensing, timing, actuation loops.
- Logistics and supply-chain coursework demonstrations.
- Exhibition projects showing a complete identify-and-divert machine.
- Foundation for vision-based sorting or multi-lane extensions.

10. Future Scope

- Camera-based label reading for damaged or rotated barcodes.
- Additional diverter gates and lanes for more destinations.
- Weighing integration for dimension/weight-based sort rules.
- A fleet view for multiple conveyor lines.