PROJECT ABSTRACT

Automatic Car Parking Lift Demo Model

A bench-scale automated parking lift: sensor-verified platform, motorized lead-screw/rack lift, limit-switch safety and an Arduino park/retrieve state machine. Delivered built-to-order with model, firmware, drawings, report, PPT and viva kit.

01. Title

Automatic Car Parking Lift Demo Model

02. Introduction / Background

Urban parking is a geometry problem: land is scarce and cars are not getting smaller, so cities increasingly stack vehicles with automated parking lifts. The engineering behind those systems — a lift mechanism that carries a tonne safely, sensors that know a car is correctly positioned, and a controller that sequences the cycle without human error — is excellent final-year material, but students rarely build it because the real machines are industrial-scale. The gap is a bench-scale model demonstrating the complete loop honestly: presence sensing, lift actuation, end-of-travel safety and automatic sequencing, all visible.

03. Problem Statement

Students study mechanisms and embedded control separately but rarely integrate them into a sequenced machine: how presence sensing gates motion, how limit switches bound travel, how a state machine runs a park/retrieve cycle, and how an E-stop fits in. The project addresses this with a two-level lift model where the Arduino runs the full cycle and every safety interlock is demonstrable.

04. Objectives

- Build a two-level frame with a motorized platform lift (lead-screw / rack-and-pinion).
- Verify car presence and position with IR sensors before any travel is allowed.
- Bound travel with top/bottom limit switches plus a software travel timeout.
- Implement the automatic park/retrieve state machine with LCD status.
- Provide a latching emergency stop with manual-reset recovery.
- Deliver a load-check and cycle-timing test procedure within the design envelope.
- Deliver the complete student kit: model, firmware, drawings, report, PPT and viva Q&A.

05. Proposed Methodology

The frame is fabricated in steel/wood with an adjustable upper-deck height. A DC gear motor drives the lead-screw/rack lift; IR sensors confirm the model car is fully on the platform. The Arduino state machine (IDLE → DETECTED → LIFTING → PARKED → RETRIEVING) gates every transition on sensor and limit-switch inputs, with a software timeout as a second layer and a latching E-stop cutting motor drive. The LCD shows the live state. The student runs the cycle-timing and load-check procedure and records results.

06. System Architecture / Working

The car is placed on the platform; IR sensors confirm full positioning. On PARK, the Arduino verifies the sensors and drives the motor up until the top limit switch trips, latching PARKED. The lower bay is freed for a second car. On RETRIEVE the controller re-verifies the bay, lowers to the bottom switch and releases. The E-stop halts the motor instantly at any point; recovery needs a manual reset and a fresh state check.

07. Hardware / Software Requirements

- Arduino-class microcontroller
- DC gear motor with lead-screw / rack-and-pinion lift
- IR proximity sensors (presence + position)
- Limit switches (travel ends)
- Latching emergency-stop button
- 16x2 LCD status display
- Motor driver module with current-limited supply
- Steel/wood two-level frame, model cars
- Arduino IDE (C/C++ firmware)

08. Expected Outcomes

- A working lift model running the automatic park/retrieve cycle.
- Demonstrated safety interlocks: presence gating, limit switches, E-stop.
- Student-measured cycle times and load-check results within the design envelope.
- A complete report, PPT and viva Q&A on mechanisms, state machines and machine safety.

09. Applications

- Mechatronics teaching laboratories and mechanism practicals.
- Demonstrating sequenced automation with safety interlocks.
- Student studies of lift mechanisms and sensor gating.

10. Future Scope

- Multi-bay management with bay-allocation logic.

- RFID-based user access and billing simulation.
- Rotary-table car transfer as an added axis.
- Mobile app for remote park/retrieve requests.