PROJECT ABSTRACT

Automatic Sequential Starter Panel for Two Three-Phase Motors

An automatic sequential starter panel that starts two 3-phase motors in enforced order with inter-start delay, interlocking and overload protection. Delivered built-to-order with wired panel, demo motors, test procedure, report, PPT and viva kit.

01. Title

Automatic Sequential Starter Panel for Two Three-Phase Motors

02. Introduction / Background

Direct-on-line starting draws 5–7 times a motor's rated current; starting two large motors together can trip the incomer, sag plant voltage and stress the supply transformer. The industrial answer is sequential starting: motors start one after another with a defined delay, and process logic often demands an order — the downstream machine must run before the upstream one feeds it. Students learn contactor control in theory but rarely wire a real interlocked sequence. This project builds a demonstration panel for two three-phase motors with an Arduino-based sequence controller (or timer-relay logic), auxiliary-contact interlocking, adjustable inter-start delay and per-motor thermal overload protection.

03. Problem Statement

Motor-control teaching usually stops at a single DOL starter diagram, so students never see how plants enforce starting order, handle a failed first motor, or protect the sequence against faults. The project addresses this with a real wired panel where the student exercises the normal sequence, injects faults, and observes the interlocked protection behavior — the standard industrial practice, made demonstrable.

04. Objectives

- Wire a two-bay starter panel: contactor + thermal overload relay per motor.
- Enforce starting order with auxiliary-contact feedback and an adjustable inter-start delay.
- Back up the controller with hardwired auxiliary-contact interlocking.
- Annunciate run/fault/sequence-blocked states on indicator lamps.
- Provide latching emergency stop with deliberate reset.
- Deliver the complete student kit: panel, demo motors, test procedure, report, PPT and viva Q&A.;

05. Proposed Methodology

The panel is built in three stages. (1) Power wiring: incomer, two contactor+OLR bays and two fractional-HP demo motors wired per the supplied schematic on DIN rail. (2) Control: the Arduino sequence controller (or timer relays) implements start order, inter-start delay,

running confirmation via auxiliary contacts, and fault blocking; hardwired interlocks duplicate the order enforcement. (3) Testing: the student runs the normal sequence, measures the inter-start delay, injects a simulated overload to show motor 2 being blocked, and exercises the e-stop — all logged for the report.

06. System Architecture / Working

Start energizes contactor 1 if healthy; motor 1 starts DOL and its auxiliary contact confirms running. After the adjustable delay, and only with running confirmed, contactor 2 energizes and motor 2 starts. An overload trip on motor 1 blocks motor 2 and lights the fault lamp. Stop shuts down in reverse order (motor 2 first). The latching e-stop drops both contactors and requires deliberate reset before any restart.

07. Hardware / Software Requirements

- 2 x 3-phase contactors with auxiliary contacts
- 2 x thermal overload relays
- Arduino-based sequence controller (or timer-relay logic)
- Control transformer / SMPS
- Indicator lamps, start/stop/e-stop push-buttons
- 2 x fractional-HP 3-phase demo motors
- DIN-rail metal enclosure
- Control schematic and test procedure

08. Expected Outcomes

- A working panel demonstrating enforced sequential starting of two motors.
- Measured inter-start delay and documented fault-injection responses.
- A verified interlock that cannot start the motors out of order.
- A complete report, PPT and viva Q&A; on motor starting and sequence control.

09. Applications

- Teaching motor control circuits and interlocking in electrical labs.
- Demonstrating protection coordination in starter panels.
- A base for three-motor or PLC-based sequence extensions.

10. Future Scope

- Three-motor sequence extension.
- PLC-based sequence controller variant.
- Current-monitoring with inrush capture.
- SCADA-style status logging of sequence events.