Latchline Controls

Legacy program in. Verified documentation out.

Functional Design Specification & Migration Review

PROJECT	Rover Pallet Shuttle (MicroLogix 1400)
CONTROLLER / PROGRAM	CALD26
DATE	02 September 2026

SAMPLE DELIVERABLE

This sample was generated from a publicly available RSLogix 500 program (a MicroLogix 1400 pallet-shuttle controller published on GitHub). It contains no client data.

hello@latchlinecontrols.com

About this document

What this is

A complete engineering documentation package produced from a single export of a running legacy program (Studio 5000 .L5X, or RSLogix 500 .SLC for SLC-500 and MicroLogix). Deterministic analysis recovers the structure — every tag, every reference, every call, every piece of dead or disabled logic — and an AI documentation pass, verified by a controls engineer, recovers the intent. The result is what a migration, audit, or FAT actually needs: a functional specification, a tag dictionary, a full cross-reference, an engineering review findings register, and an executable test plan.

How to read it

Sections 1-3 describe what the program does and where every signal comes from. Section 4 lists every item that needs a human ruling before migration — disabled logic, uncalled routines, suspect tags. Section 5 is a factory acceptance test plan derived directly from the logic: every boolean output the program writes, with the conditions that drive it and drafted test steps. Statements the analysis could not prove from the export alone are marked VERIFY and collected for review — nothing is silently guessed.

The offer

Fixed price, fixed deadline, per program: legacy program in, verified documentation out. Conversion packages (PLC-5 / SLC-500 to ControlLogix with restructured code and FAT support) are quoted per system after a free structural analysis of your export.

Contents

1. Functional Design Specification · page 2
2. Tag Dictionary — in the full deliverable
3. Cross-Reference — in the full deliverable
4. Engineering Review Findings — in the full deliverable
5. FAT Test Plan — in the full deliverable

This is an excerpt. The full deliverable includes every section above, plus the complete rung listings appended to each routine.

Functional Design Specification

Scope: 1 program(s), 13 routine(s), 250 rung(s), 1783 instruction(s), 0 tag(s).

Program: Processor

Program overview — CALD26 (MicroLogix 1400, Bul.1766)

What the program controls

The program runs a battery-powered pallet-transport rover. The rover has two wheel drives (X and Y) selected by a motor-driven cam that changes wheel orientation, a three-position lift with analog position feedback, wheel encoders on high-speed counters, down- and up-looking photo eyes plus two Ethernet Keyence sensor units for floor-flag and pallet-presence detection, an Ethernet barcode reader for position ID, and a dock/charger. It receives motion commands from a System PLC ("TC") over a radio/Ethernet link and reports status back. A MobileView pendant provides manual control and can inject the same command frames the radio delivers. Tag descriptions are absent from the export; all names in this specification are inferred (see VERIFY below).

How the routines divide the work

- **LAD_2_MAIN** — loads constants, calls the twelve subroutines in a fixed non-sequential order (3, 4, 12, 5, 7, 8, 6, 9, 11, 10, 13, 14), skips LAD_5_SEQ_MOV in Mobile View mode (B3:10/2), and clears the math-overflow flag S:5/0 every scan.
- **LAD_3_POWER** — power-up detection, battery charging on the dock, low-battery alarm and 5-minute power shutdown, and the two safety-relay feed outputs driven by the network E-stop latch.
- **LAD_4_IO_MAP** — conditions discrete/analog/Keyence inputs into status bits (E-stop status B3:4/0, pallet-present, full-bin, battery %), drives indicator lights and buzzer.
- **LAD_5_SEQ_MOV** — decodes the accepted command, computes distance/speed/accel-decel profile, and sequences cam, lift and wheel requests to the three motor routines.
- **LAD_6_MOTOR_WHL / LAD_7_MOTOR_CAM / LAD_8_MOTOR_LIFT** — axis controllers: wheel velocity profile and analog speed outputs, cam window positioning with inching approach, lift positioning to three analog windows. Each also has a Mobile View manual path.
- **LAD_9_RADIO_IN / LAD_10_RADIO_OUT** — inbound command validation (address, transaction ID, data checks) and outbound 20-word status block plus network stop-command handling.
- **LAD_11_BARCODE** — four-message CIP handshake with the reader; produces position ID N11:39.
- **LAD_12_MBLVIEW** — pendant mode selection and HMI-button command injection into the radio buffer.
- **LAD_13_FAULTS** — thirteen timed fault detectors, single master fault bit and code, radio fault report, post-move encoder verification.
- **LAD_14_VELOCITY** — diagnostic velocity calculation and phase-duration FIFO logs; only F39:28 is consumed elsewhere.

Main operating sequence

1. A command arrives in N11:0–9 (radio, or pendant via LAD_12). LAD_9 validates it, checks rover ID (N10:2) and new transaction ID, and latches command data. Any accepted motion command clears the B30 fault file.
2. LAD_5 sets "sequence active" N7:10/1 (blocked while a fault B30:0/0 is set), computes the profile, and issues requests in command-specific order: cam index (N7:10/11) -> cam settle -> wheel move (N7:10/12, requires charger-clear permissive B9:0/2) -> lift (N7:10/13), or lift first for code 20; codes 5/8/22 are single-step variants.
3. LAD_7 positions the cam into the target window; LAD_6 drives the wheels through accel/cruise/decel/creep and stops on floor flags; LAD_8 moves the lift to the coded position. Completion bits (N7:11/1, N7:11/6, N7:10/15) return to LAD_5.

4. LAD_5 latches move-complete N7:12/0 and requests a radio report; LAD_10 sends the status block and retries every 5 s until acknowledged. LAD_5 then clears the N7:10–N7:13 words. LAD_6 zeroes the encoders and holds position while idle.
5. In parallel: LAD_11 reports new barcodes seen inside a position window, LAD_10 sends a periodic status every ~50 s while stationary, and LAD_3 charges when docked and idle.

Interlock and alarm philosophy

- **Master permissive B3:4/0** (E-stop status from LAD_4) gates every automatic motion path, the sequencer, the barcode cycle and the analog speed outputs. It is standard PLC logic; I:0.0/8 alone satisfies it. VERIFY: existence and coverage of a hardwired safety circuit independent of the PLC, and the polarity/meaning of I:0.0/8–11.
- **Network stops** (99 E-stop, 152 stop, 252 kill) are software commands. Code 99 drops the safety-relay feed outputs, which LAD_3 re-arms automatically after 3 s. VERIFY (safety): acceptability of auto re-arm and of communications-mediated stops that wait for any in-flight message.
- **Gaps to review:** wheel drive enables (O:0.0/2, /3) are never unlatched on fault or E-stop in the PLC; lift jog and cam jog bypass B3:4/0 and have only analog/software travel limits; in Mobile View, analog outputs and enables hold last values when a button is released. VERIFY each with the plant safety assessment.
- **Fault handling** is centralised in LAD_13: the first fault to trip latches B30:0/0, writes one code to N11:41, sets a code bit in B30:30–37, latches drive inhibit N7:11/10, and reports by radio up to three times. A fault blocks new sequences (LAD_5) and zeroes encoders (LAD_6) but does not directly stop cam or lift motors. Reset is remote only — any accepted motion command or code 152 clears the whole B30 file (LAD_9). No communication loss to the Keyence units, barcode reader or System PLC is detected as a fault.
- **Overflow masking:** LAD_2 clears S:5/0 every scan, so divide-by-zero and overflow conditions program-wide are silent.

VERIFY: Many gating and configuration bits (B3:8/0–4, B3:9/2, B3:10/5–15, B3:11/x, B3:13/0, B3:16/x, N7:3, all setpoints) have no PLC writer and are assumed HMI- or data-table-sourced; their values and sources must be captured before migration.

Routine: LAD_2_MAIN

Purpose

LAD_2_MAIN is the main routine of the Processor program. It loads three constants each scan, calls the twelve functional subroutines in a fixed order, and clears the math-overflow status bit and one program bit at the end of every scan.

Functional description

Constant initialisation (rung 0) Every scan the routine writes three fixed values:

- N10:1 = 1
- N10:2 = 1026
- N10:3 = 9999

Purposes are inferred from usage, since no descriptions are present in the export:

- N10:2 (1026) is read only by the LAD_9_RADIO_IN and LAD_10_RADIO_OUT routines. It is inferred to be a radio message identifier, node, or address constant used by those routines.
- N10:3 (9999) is read by the LAD_3_POWER and LAD_4_IO_MAP routines. It is inferred to be a sentinel / out-of-range comparison value (see those routines for how it is used).
- N10:1 (1) is not read by any routine in the program digest. It may be read externally (HMI, radio, or message data table) — see Review items.

Because these are written unconditionally every scan, any attempt to change them from an HMI or via a message write is overridden on the next scan.

Subroutine scheduling (rungs 1–12) The subroutines are called once per scan in this order. Note the order does not match the file numbers:

Scan order	Routine	Condition
1	LAD_3_POWER	Unconditional
2	LAD_4_IO_MAP	Unconditional
3	LAD_12_MBLVIEW	Unconditional
4	LAD_5_SEQ_MOV	Only while B3:10/2 is OFF
5	LAD_7_MOTOR_CAM	Unconditional
6	LAD_8_MOTOR_LIFT	Unconditional
7	LAD_6_MOTOR_WHL	Unconditional
8	LAD_9_RADIO_IN	Unconditional
9	LAD_11_BARCODE	Unconditional
10	LAD_10_RADIO_OUT	Unconditional
11	LAD_13_FAULTS	Unconditional
12	LAD_14_VELOCITY	Unconditional

Scan-order consequences worth preserving in a migration:

- LAD_12_MBLVIEW runs before LAD_5_SEQ_MOV, so the B3:10/2 gate (written in LAD_12_MBLVIEW) is evaluated with the current-scan value.
- LAD_5_SEQ_MOV runs before the three motor routines, so sequence commands written in LAD_5 act on the motor routines in the same scan.
- LAD_9_RADIO_IN runs after the motor routines, so radio inputs decoded there are not acted on by the motor logic until the following scan.
- LAD_13_FAULTS runs after all motion and communications routines, so it evaluates the state produced in the same scan.

End-of-scan housekeeping (rung 13)

- S:5/0 (processor minor-error bit: math overflow / division-by-zero trap) is unlatched every scan. On this controller family, if S:5/0 is still set at the end of the scan the processor declares a major fault and stops. Clearing it here prevents that fault and allows the program to keep running after an arithmetic overflow.
- B3:12/15 is unlatched every scan. It is read by the LAD_13_FAULTS and LAD_14_VELOCITY routines. No routine in the program writes it (the only write in the digest is this unlatch), so it can only be set by an external source such as an HMI or a message write, and it is then held ON for at most part of one scan. It behaves as a one-shot acknowledgement or trigger bit (inferred).

Interlocks and permissives

- **LAD_5_SEQ_MOV is skipped while B3:10/2 is ON.** B3:10/2 is produced by the LAD_12_MBLVIEW routine and is also read directly by the LAD_6_MOTOR_WHL, LAD_7_MOTOR_CAM, LAD_8_MOTOR_LIFT and LAD_13_FAULTS routines. Its purpose is inferred to be a manual / HMI-control mode selector that suspends the automatic movement sequence. This is safety-relevant: when a subroutine is not scanned on this platform, every output, bit and timer written only inside it retains its last value and timers stop accumulating. Any sequence command or state bit that LAD_5_SEQ_MOV last left ON stays ON while B3:10/2 is ON, unless another routine clears it. See the LAD_5_SEQ_MOV and LAD_12_MBLVIEW sections for the affected tags.
- **S:5/0 is cleared every scan.** This is not an interlock but is safety-relevant: it suppresses the processor's math-overflow major fault. Overflowed arithmetic results elsewhere in the program (for example in the LAD_14_VELOCITY float calculations) are therefore used silently rather than stopping the processor.

No other permissives exist in this routine. All other subroutine calls are unconditional.

Alarms and faults

None generated in this routine. Fault handling is in the LAD_13_FAULTS routine. This routine does suppress one processor fault (S:5/0 math overflow) as described above.

Review items

- VERIFY: Meaning of B3:10/2. It is written by the LAD_12_MBLVIEW routine and gates the automatic sequence. Confirm with the plant that skipping LAD_5_SEQ_MOV (rather than forcing its outputs to a safe state) while this bit is ON is intended behaviour, and confirm what state the LAD_5_SEQ_MOV outputs are expected to hold during that time.
- VERIFY: Clearing S:5/0 every scan masks arithmetic overflow errors program-wide. Confirm whether this is intended, and whether the migrated platform should replicate it or should instead detect and alarm overflows.
- VERIFY: N10:1 (=1) is written here but not read by any routine in the program. Confirm whether it is read externally (HMI, radio link, barcode reader, or message table). If not, it is dead data.
- VERIFY: N10:2 (=1026) and N10:3 (=9999) are inferred to be a radio/message constant and a sentinel value respectively. Confirm their meaning and whether they should become configurable parameters in the migrated system rather than hard-coded constants.
- VERIFY: B3:12/15 is unlatched here and read by the LAD_13_FAULTS and LAD_14_VELOCITY routines, but no routine sets it. Confirm the external source (HMI or message) that sets this bit, or whether the consuming logic is effectively dead.
- Rung 0 rewrites N10:1–N10:3 every scan, so any HMI or message write to these words is overwritten within one scan. Flagging in case an operator-adjustable value was intended.
- The subroutine call order (3, 4, 12, 5, 7, 8, 6, 9, 11, 10, 13, 14) is not sequential by file number. The migration must preserve this order, in particular MBLVIEW before SEQ_MOV, SEQ_MOV before the motor routines, and RADIO_IN after the motor routines.
- No tag descriptions were present in the export for any tag used in this routine. All tag purposes above are inferred.

Routine: LAD_3_POWER

Purpose

LAD_3_POWER manages the rover's electrical power state: it detects power-relay start-up, controls battery charging when the rover is docked, shuts down power and raises alarms on low battery, and drives the PLC outputs that feed the safety power relay in response to E-stop commands. It is called unconditionally from LAD_2_MAIN rung 1 every scan.

Functional description

The tag dictionary in this export is empty. Every tag meaning below is inferred from the tag name, its usage in this routine, the rung comments, and the cross-routine digest, and should be treated as inferred until confirmed.

Power-relay start-up detection (rungs 0–1) A one-shot (storage B9:0/4, output B9:0/5) fires on the first program scan (S:1/15), on the rising edge of B3:4/0 (inferred: safety power relay status, mapped from field input in LAD_4_IO_MAP), or on the rising edge of N7:10/1 (inferred: a sequence state bit produced by LAD_5_SEQ_MOV; from its use in rungs 3 and 5 it appears to indicate the rover is moving/active).

Rung 1 builds a sealed "power active" flag B3:2/6. It is set by the start-up one-shot B9:0/5 or by B3:3/1 (inferred: stop-state/radio flag produced by LAD_9_RADIO_IN — the rung comment calls it "Stop State"), and seals through itself. It holds only while B3:4/0 is true and while EITHER N7:11/10 is off (inferred: a stop/fault request produced by LAD_10_RADIO_OUT and LAD_13_FAULTS) OR all of the following are off: N7:12/8 (written by LAD_6_MOTOR_WHL), N7:10/12 (written by LAD_5_SEQ_MOV / LAD_6_MOTOR_WHL), O:0.0/10 (lift motor output, LAD_8_MOTOR_LIFT) and O:0.0/8 (camera motor output, LAD_7_MOTOR_CAM). In effect: if N7:11/10 is active and any motor is running, the power-active flag drops. B3:2/6 is consumed only by rung 2 of this routine.

Yellow indicator flash on power-up (rung 2) While B3:2/6 is true, T4:25 (1.7 s) runs once. Only while T4:25 is timing, T4:0 (0.33 s) and T4:49 (0.33 s) alternate to produce B3:6/2, the yellow-light flash bit, which LAD_4_IO_MAP maps to a physical output. B3:6/2 is on while T4:49 is timing, giving roughly 0.33 s on / 0.33 s off. Because T4:25 is a non-retentive TON that stays done once it expires, the flash is a burst of about 1.7 s (two to three flashes) after B3:2/6 rises, not a continuous flash.

Battery charging on the charger (rungs 3–4) Charging is permitted when N7:10/1 is off (rover not moving), B3:4/0 is true (power relay active), field input I:0.0/7 is true (inferred: docked-on-charger sensor), and the current position ID N11:39 (inferred: barcode/position ID produced by LAD_11_BARCODE) lies between N7:70 and N7:71 (inferred: low/high charger-position ID setpoints). When all are true, T4:1 (15 s) starts. After 15 s the charging relay O:0.0/7 energizes. B9:0/0 (inferred: charge indicator, mapped to an output in LAD_4_IO_MAP) is on only while the rung is true and T4:1.ACC < 5, i.e. for the first 5 s of the dwell, then goes off. Loss of any condition drops the charging relay immediately.

Rung 4: whenever the charging relay O:0.0/7 is off, T4:31 (3 s) runs; when it times out B9:0/2 (inferred: "OK to move" permissive, consumed by LAD_5_SEQ_MOV) is set. The comment states this waits for the charger to finish moving down. B9:0/2 is off while charging and for 3 s after charging stops.

Low-battery power shutdown (rung 5) When N7:10/1 is off (not moving), the battery level N7:0 (inferred: battery percentage, produced in LAD_4_IO_MAP) is at or below cutoff setpoint N7:18, and the rover is NOT on a charger (I:0.0/7 off, or N11:39 outside N7:70..N7:71), T4:2 (300 s = 5 min) runs. When it times out, O:0.0/6 energizes. Per the rung comment this shuts off the power relay to prevent full discharge. O:0.0/6 is written nowhere else in the program.

Low-battery alarm and charger request (rungs 6–7) When N7:0 ≤ N7:5 (inferred: minimum operating percentage) continuously for 3 s (T4:34), the low-battery alarm B3:7/15 is latched and, once per event (ONS N7:20/7), B9:2/3 is latched. B9:2/3 is the "send radio message" trigger consumed by LAD_5_SEQ_MOV and LAD_10_RADIO_OUT; the comment states the message requests that the rover be sent to a charger. B3:7/15 is also read by LAD_10_RADIO_OUT for message content.

While B3:7/15 is latched and N7:0 > N7:53 (inferred: full-charge clear threshold) for 30 s (T4:17), B3:7/15 is unlatched and B9:2/3 is latched once more (ONS N7:20/8) to report the cleared state.

E-stop and safety relay outputs (rungs 8–9) B3:8/15 (inferred: E-stop command latch, set in LAD_10_RADIO_OUT) controls two PLC outputs, O:0.0/0 and O:0.0/1, which the comment describes as the safety contacts feeding the Safety Power Relay. Both outputs are ON while B3:8/15 is OFF and drop to OFF when B3:8/15 is set. Rung 9: while both outputs are off, T4:37 (3 s) runs; when it times out, B3:8/15 is unlatched by this routine, and on the next scan rung 8 re-energizes both outputs. The E-stop command therefore produces a 3 s de-energization of the safety outputs followed by automatic re-arm by the PLC, unless LAD_10_RADIO_OUT keeps re-latching B3:8/15.

HMI (Mobile View) E-stop status (rungs 10–11) When B3:3/1 is true and the received radio word N11:20 (LAD_9_RADIO_IN) equals N10:3 (a value written in LAD_2_MAIN, inferred to be the E-stop command code), B3:4/9 (HMI E-stop status) is driven by OTE. Rung 11 starts T4:18 (0.5 s) while B3:4/9 is on and unlatches it when done. Because B3:4/9 is an OTE output of rung 10, the OTU in rung 11 is overwritten every scan while rung 10 is true and is redundant when rung 10 is false; rung 11 has no effective behavior. No sibling routine reads B3:4/9 (see Review items).

Rung 12 is the routine return.

Interlocks and permissives

- **Safety relay outputs O:0.0/0 and O:0.0/1** are held ON only while E-stop latch B3:8/15 is clear. Setting B3:8/15 (LAD_10_RADIO_OUT) drops both outputs, de-energizing the external Safety Power Relay. The PLC auto-clears the latch 3 s later (rung 9) and re-energizes the outputs. Safety-relevant — see Review items.
- **Low-battery power shutdown O:0.0/6** energizes after 5 continuous minutes of N7:0 ≤ N7:18 while not moving and not on a charger. This removes power to the rover. It does not act while N7:10/1 is true.
- **Movement permissive B9:0/2** (consumed by LAD_5_SEQ_MOV) is withheld while the charging relay O:0.0/7 is energized and for 3 s after it de-energizes.

- **Charging relay O:0.0/7** requires: rover not moving (N7:10/1 off), power relay active (B3:4/0), docked input I:0.0/7, position ID within N7:70..N7:71, and a 15 s dwell. Any condition dropping de-energizes the relay immediately.
- **Power-active flag B3:2/6** requires B3:4/0 and drops if N7:11/10 is active while any motor output/state (N7:12/8, N7:10/12, O:0.0/10, O:0.0/8) is on. Its only consumer is the indicator flash.

Alarms and faults

- **Low battery alarm B3:7/15** — latched when N7:0 ≤ N7:5 for 3 s; unlatched when N7:0 > N7:53 for 30 s. Both transitions trigger a radio message via B9:2/3 (see LAD_10_RADIO_OUT).
- **Low-battery shutdown O:0.0/6** — see interlocks; a protective action rather than an annunciated alarm.
- **HMI E-stop status B3:4/9** — driven when the received radio word matches N10:3.

Review items

- VERIFY: The tag dictionary in this export is empty. All tag meanings above (B3:4/0 power relay status, N7:10/1 moving/active, B3:3/1 stop state, N7:11/10 stop/fault request, I:0.0/7 docked sensor, N7:0 battery level, N7:5/N7:18/N7:53 battery setpoints, N7:70/N7:71 charger position range, B3:8/15 E-stop latch, N10:3 E-stop command code) are inferred and must be confirmed against the online tag descriptions or electrical drawings.
- VERIFY (safety): Rungs 8–9 auto-clear the E-stop latch B3:8/15 and re-energize safety outputs O:0.0/0 and O:0.0/1 three seconds after they drop, with no operator reset in the PLC. Confirm the external Safety Power Relay circuit has its own latching/manual-reset and that this behavior is acceptable to the safety assessment. Migration must not silently change this timing or add/remove a reset.
- VERIFY (safety): Confirm the physical function of O:0.0/6 (rung 5). The comment says it shuts off the power relay; the wiring and its interaction with the safety relay are not in the export.
- VERIFY: Setpoint values N7:5, N7:18, N7:53, N7:70, N7:71 and the source/scaling of N7:0 (produced in LAD_4_IO_MAP) are not in this export.
- VERIFY: Rung 2 flashes the yellow light for only ~1.7 s after B3:2/6 rises (T4:25 is a one-shot TON). The comment implies a continuous flash while the relay is energized. Confirm whether the burst behavior is intended.
- VERIFY: In rung 3 the charge indicator B9:0/0 is on for only the first 5 s of the 15 s dwell and off thereafter, including while the charging relay is energized. Confirm this is the intended indication.
- VERIFY: B3:4/9 (HMI E-stop status) is written here but read by no other routine in the program. It may be read directly by the Mobile View HMI (LAD_12_MBLVIEW does not reference it). Confirm whether it is used; if not, rungs 10–11 are dead code.
- Suspicious: Rung 11 (OTU B3:4/9 after 0.5 s) has no effect, because B3:4/9 is an OTE output of rung 10 and is rewritten every scan. Likely dead logic.
- Suspicious: Rung 1's hold condition permits the power-active flag to remain set while motors run as long as N7:11/10 is off; the intent of the N7:11/10 branch should be confirmed with the plant engineer.
- Note: N7:20/7 and N7:20/8 are ONS storage bits and must be preserved as one-shot storage in the migrated platform; N7:20 is also used as storage by other routines.
- Note: Timer ACC values in the export (T4:25 = 170, T4:31 = 3) are retained data from the saved program state, not configuration.

Routine: LAD_4_IO_MAP

Purpose

LAD_4_IO_MAP conditions the rover's physical and networked inputs into internal status bits and words, and drives the rover's indicator lights and buzzer. It is the boundary layer between raw I/O (discrete inputs, analog inputs, two Keyence Ethernet sensor units) and the sequence, motor, fault and radio routines that consume the mapped values.

Functional description

Keyence sensor polling (rungs 0–2). A self-resetting timer (T4:13, 1 ms base, preset 30) produces a one-scan pulse roughly every 30 ms. Each pulse triggers two MSG instructions over the Ethernet port (SLC:1 MultiHop): MG20:5 exchanges data with the device at 192.168.100.4 using N14:0 as the local buffer, and MG20:6 exchanges data with 192.168.100.5 using N15:0 as the local buffer. Because this routine reads N14:15–N14:18 and N15:15–N15:20 as sensor values, the messages are almost certainly reads from the Keyence units into N14 and N15. No routine reads the MG20:5 or MG20:6 status bits (/DN, /ER, /EN), so a lost or failed message is not detected anywhere in the program; the N14/N15 sensor words simply stop updating and the derived presence bits freeze at their last values.

E-stop status (rung 3). B3:4/0 is set when either I:0.0/8 is on, OR all three of I:0.0/9, I:0.0/10 and I:0.0/11 are on. The rung comment states these are the physical E-stops on the rover and the E-stop/deadman from the MobileView pendant. B3:4/0 is consumed as a condition by nearly every other routine (POWER, SEQ_MOV, MOTOR_WHL, MOTOR_CAM, MOTOR_LIFT, RADIO_OUT, BARCODE, FAULTS). I:0.0/9 is also read directly by the MBLVIEW routine, which supports the comment that at least that input is MobileView-related. The polarity and meaning of the individual inputs are not defined in the export; see Interlocks and Review items.

Down-looking photo eyes (rungs 4–7). Four discrete inputs are copied directly into bits of the N14:70 status word: I:0.0/13 -> N14:70/0, I:0.0/15 -> N14:70/6, I:0.1/1 -> N14:70/7, I:0.1/3 -> N14:70/13. Per the comment these detect floor flags for fine positioning at intersections. N14:70 is consumed bit-wise by MOTOR_WHL, MOTOR_CAM and FAULTS and as a whole word by RADIO_OUT (status reported to the host).

Pallet-present sensing (rungs 8–13, 18). Six Keyence sensor values are each compared against a single threshold word, N14:72. A value greater than or equal to N14:72 sets the corresponding "present" bit:

- N15:15 -> N14:70/2
- N15:20 -> N14:70/9
- N14:15 -> N14:70/3
- N15:16 -> N14:70/4
- N14:16 -> N14:71/1
- N14:17 -> N14:71/0

Rung 18 combines two of these: B3:3/3 (inferred: "pallet on rover") is set when N14:70/2 OR N14:70/9 is true. B3:3/3 is consumed by MOTOR_WHL and RADIO_OUT. N14:72 is not written by any routine, so it is a fixed data-table constant.

Up-looking photo eyes (rungs 14–17). Four more Keyence values are compared against the same N14:72 threshold. Per the comment these confirm the pallet is centred and square on the rover:

- N15:17 -> N14:70/10
- N14:18 -> N14:70/11
- N15:18 -> N14:71/3
- N15:19 -> N14:71/2

Full-bin sensors (rungs 19–22). Four discrete inputs are copied to B3:12 bits: I:0.0/12 -> B3:12/0, I:0.0/14 -> B3:12/2, I:0.1/0 -> B3:12/3, I:0.1/2 -> B3:12/6. The export comment "Full Bin Sensors — detects if a pallet is located in the storage bay in the target move location" is attached to rung 22 only. Rungs 19–21 are uncommented, but they have the same structure and their outputs are consumed by the same routines (SEQ_MOV and FAULTS), so I infer that all four bits are full-bin sensors. This is an inference.

Rover indicator lights (rungs 23–24). Two base-unit outputs drive indicator lights:

- O:0.0/5 is on when (B3:6/5 OR B3:6/7) AND NOT B3:6/2, OR B3:6/2, OR B9:0/0.
- O:0.0/4 is on when (B3:6/4 OR B3:6/6) AND NOT B3:6/2, OR B3:6/2, OR B9:0/0.

B3:6/4–7 are written by MOTOR_WHL and are inferred to be wheel-motion/direction indications. B3:6/2 and B9:0/0 are written by POWER; both force the lights on regardless of motion. As written, the XIO B3:6/2 in the first branch has no effect because B3:6/2 alone is a parallel path — each output reduces to a plain OR of its motion bits, B3:6/2 and B9:0/0. See Review items.

Buzzer (rung 25). O:0.4/13 (expansion slot 4) is on when N7:10/12 is set AND N10:7 equals 3 AND B3:6/2 is off. N7:10 is the sequence status word written by SEQ_MOV (and one bit by MOTOR_WHL); N10:7 is written by SEQ_MOV and is inferred to be a move-type or destination-type code, with 3 corresponding to a storage-bay entry. The rung comment (shifted, see Review items) describes the buzzer as warning operators that rover motion can create a pinch-point hazard.

Battery charge percentage (rungs 26–27). The analog input word I:0.4 is divided by the constant F8:14 to give F8:19 (inferred: battery voltage as a fraction of full charge). F8:19 is multiplied by 100.0 into N7:0 and N7:0 is clamped to a maximum of 100. There is no lower clamp. N7:0 is consumed by POWER, RADIO_OUT and FAULTS. F8:14 is not written by any routine, so it is a fixed data-table constant.

Lift analog pass-through (rung 28). The analog input word I:0.7 is copied unconditionally to N7:2. I:0.7 is also read directly by MOTOR_LIFT, and N7:2 is consumed by MOTOR_LIFT and RADIO_OUT; I infer this is the lift position feedback.

Radio-addressed command bit (rung 29). When N11:20 (written by RADIO_IN; inferred: rover address in the received message) equals N10:3 (written by MAIN; inferred: this rover's address), bit 5 of N11:17 is copied into B3:1 through a 0020h mask. Only B3:1/5 is affected. No routine reads any bit of B3:1, and no routine writes N11:17 directly (the MBLVIEW routine performs a file-level write to #N11:0 that may cover it). See Review items.

Exit (rung 30). Unconditional RET.

Interlocks and permissives

- **E-stop status B3:4/0 (rung 3)** — safety-relevant. B3:4/0 = I:0.0/8 OR (I:0.0/9 AND I:0.0/10 AND I:0.0/11). This single bit gates motion and sequencing in POWER, SEQ_MOV, MOTOR_WHL, MOTOR_CAM, MOTOR_LIFT, RADIO_OUT, BARCODE and FAULTS. Note that I:0.0/8 alone satisfies the rung and therefore bypasses the other three inputs. The E-stop function is implemented in standard PLC logic here; whether a hardwired safety circuit also removes motor power cannot be determined from the logic.
- **Indicator light outputs O:0.0/4 and O:0.0/5** — forced on by B3:6/2 or B9:0/0 (both from POWER); otherwise follow wheel-motion bits B3:6/4–7 from MOTOR_WHL.
- **Buzzer O:0.4/13** — requires N7:10/12 set, N10:7 = 3, and B3:6/2 clear.
- **Pallet-present / up-looking bits** — are permissives consumed downstream (MOTOR_WHL, SEQ_MOV, FAULTS), and all depend on the Keyence messages continuing to succeed. There is no communication-health interlock on them.
- **Full-bin bits B3:12/0, /2, /3, /6** — permissives consumed by SEQ_MOV and FAULTS.

Alarms and faults

None generated in this routine. This routine produces the raw status bits (B3:4/0, B3:12/x, N14:70/x, N14:71/x, N7:0) that the FAULTS routine evaluates; see the LAD_13_FAULTS section. No fault is raised for a failed or stale Keyence message (MG20:5 / MG20:6 error and done bits are unused program-wide).

Review items

- VERIFY: Polarity and wiring of E-stop inputs I:0.0/8, I:0.0/9, I:0.0/10, I:0.0/11. Confirm whether B3:4/0 = 1 means "E-stop circuit healthy" or "E-stop active", and confirm which physical device each input represents. The consuming routines treat B3:4/0 as a single condition; migration must preserve its exact polarity.
- VERIFY: Intent of the I:0.0/8 parallel branch in rung 3. As written, I:0.0/8 alone asserts B3:4/0 and bypasses I:0.0/9–11 entirely. If I:0.0/8 is a "MobileView not connected" or bypass input, confirm that this behavior is acceptable to the plant's safety assessment. Confirm whether a hardwired E-stop safety relay exists independent of this logic.
- VERIFY: MSG configuration of MG20:5 and MG20:6 — direction (read vs. write), element count, remote data address, and timeout. The export shows the raw parameter list but the meaning of each field is not decodable with confidence. The local buffers are N14:0 and N15:0.

- VERIFY: Element count of the MG20:5 message. The threshold constant N14:72 and the status words N14:70/N14:71 live in the same data file as the N14:0 receive buffer. If the message length is 71 words or more, the incoming data will overwrite the status words and the threshold.
- VERIFY: Value and units of threshold N14:72, and confirm that a single threshold is intended for all ten Keyence sensor comparisons (six pallet-present, four up-looking).
- VERIFY: Value of F8:14 (battery scaling divisor). If F8:14 is 0.0 the DIV in rung 26 sets the math overflow flag (S:5/0, which MAIN writes) and the result is indeterminate. Also confirm the analog input scaling for I:0.4 and whether a lower clamp on N7:0 (negative percentage) is required.
- VERIFY: Analog input I:0.7 assignment (inferred: lift position). Both this routine (to N7:2) and MOTOR_LIFT use it.
- VERIFY: Assignment of I:0.0/12, I:0.0/14, I:0.1/0 (rungs 19–21). Inferred as full-bin sensors from structure and consumers, but only rung 22 carries the "Full Bin Sensors" comment.
- VERIFY: Meaning of N10:7 = 3 (buzzer condition, rung 25) and N7:10/12. Both come from SEQ_MOV; confirm 3 is the storage-bay-entry move code.
- VERIFY: Physical I/O for O:0.0/4, O:0.0/5 (indicator lights) and O:0.4/13 (buzzer, expansion slot 4). Confirm the expansion module in slot 4 exists in the hardware configuration.
- Suspicious — redundant logic in rungs 23 and 24: the XIO B3:6/2 in the motion branch is defeated by the parallel XIC B3:6/2, so each light is simply ON whenever any of its motion bits, B3:6/2 or B9:0/0 is true. If the intent was a flashing light (e.g., alternate motion/flasher), the logic does not do that. VERIFY: intended light behavior with the plant engineer.
- Suspicious — likely dead output, rung 29: B3:1/5 is written but no routine in the program reads any bit of B3:1. Also, N11:17 has no direct writer; the only candidate is the file-level write to #N11:0 in MBLVIEW. VERIFY: whether rung 29 is obsolete or B3:1/5 is consumed by an HMI/host outside the PLC.
- Suspicious — no error handling on the Keyence messages. MG20:5/ER, MG20:6/ER, /DN and /EN are not read anywhere. A cabling or sensor-unit failure leaves the pallet-present and up-looking bits frozen at their last state with no fault. Recommend adding communication watchdog logic during migration.
- Observation — rung comments in the tail of the routine are offset from the rungs they describe: "Rover Indicator Lights" is attached to rung 27 (battery math) but describes rungs 23–24; "Buzzer" is attached to rung 29 but describes rung 25; "Battery Charge Level Conversion" is attached to the RET (rung 30) but describes rungs 26–27. Comments in rungs 0–18 align correctly. This documentation follows the logic, not the comment positions.
- Observation — the routine writes N14:70 bits 0, 2, 3, 4, 6, 7, 9, 10, 11, 13 and MOTOR_CAM writes bit 14. Bits 1, 5, 8 and 12 of N14:70 are read by MOTOR_WHL and FAULTS but are not written by any routine; see those sections. VERIFY: whether these are spare, obsolete, or expected to be set from outside the program.
- Observation — the MSG triggers re-arm every ~30 ms with no check that the previous message completed. On a MicroLogix 1400 the MSG ignores a new trigger while enabled, so this is functionally tolerated but is not good practice; consider triggering on /DN or /ER during migration.

Routine: LAD_5_SEQ_MOV

Purpose

LAD_5_SEQ_MOV decodes a validated motion command received over radio (LAD_9_RADIO_IN), computes the distance, speed and acceleration/deceleration profile for that move, and sequences the cam, lift and wheel motion requests that the motor routines (LAD_6/7/8) act on. It also issues the move-complete message and resets the motion latches when the move is done, aborted, or the E-stop is reset.

Functional description

All tag names below are inferred from usage and rung comments; the export contains no tag descriptions.

Sequence enable (rungs 0–1)

- `N7:10/1` is the "motion sequence in process" latch. It is set when LAD_9_RADIO_IN raises `B9:0/7` (new valid motion command) and no fault is active (`B30:0/0` off, produced by LAD_13_FAULTS). On the same rising edge a radio transmit request `B9:2/3` is set (consumed by LAD_10_RADIO_OUT, inferred as "send message").

- `N7:10/1` is unlatched whenever `B3:4/0` (E-stop OK, mapped in LAD_4_IO_MAP) is off. Every parameter and sequencing rung below is gated by `B3:4/0` AND `N7:10/1`.
- `N7:10/1` is consumed by nearly every other routine (POWER, WHL, CAM, LIFT, RADIO_IN/OUT, BARCODE, MBLVIEW, FAULTS) as the global "a move is active" status.

Command decoding (rungs 2, 3, 5) The command image comes from LAD_9_RADIO_IN: `N11:23` = command code, `N11:24`/`N11:25`/`N11:27`/`N11:28` = parameters. Codes observed in this routine:

N11:23	Sequence
5	Index cam only
8	Move lift only
19	Wheel move, then lift
20	Lift, then wheel move
22	Wheel move with no cam positioning step
21	Excluded from stop-type and direction latching (no sequence in this routine)

- Stop type `N10:7` <- `N11:25`. Unless code 21: 1 -> `N7:10/8` (move to intersection), 2 -> `N7:10/10` (move into storage location), 3 -> `N7:10/9` (move commanded distance with buzzer and lights). These bits are consumed by LAD_6_MOTOR_WHL.
- Lift target `N10:9` <- `N11:24` for code 8, or <- `N11:27` for codes 19–20. Consumed by LAD_8_MOTOR_LIFT.
- Direction word `N7:7` <- `N11:28`, except for code 5 where `N7:7` <- `N11:24`. Unless code 21, `N7:7` latches: 1 -> `N7:10/2`, 2 -> `N7:10/3`, 3 -> `N7:10/4`, 4 -> `N7:10/5`, 5 -> `N7:10/14`. Directions 1 and 3 also set axis group `N7:10/7`; directions 2 and 4 set axis group `N7:10/6`. The two groups map to the two cam positions `B3:5/9` / `B3:5/10` from LAD_7_MOTOR_CAM, which indicates a four-way vehicle where the cam selects the active wheel set.

Motion parameter calculation (rungs 4, 6–10, 15) — recomputed every scan while in process.

- Total move distance in encoder counts: `L12:21 = (F8:22 + F8:23/1000) × F8:0`, copied to `L12:18`. `F8:22`/`F8:23` are the commanded distance (whole and thousandth units) from LAD_9_RADIO_IN; `F8:0` is the count-per-unit scale for the selected axis (`F8:45` for group `N7:10/7`, `F8:46` for group `N7:10/6`). `L12:9 = L12:18 + N7:60` is an over-travel limit consumed by LAD_13_FAULTS.
- Speed: `F8:7 = N11:24 / F8:8`, capped at 9.0 (rung 4; see Review items — the author recommends removing this). Speed setpoint `F8:2 = F8:7 ×` one of four multipliers selected by axis group and by lift status `B3:5/13` (`F8:16`/`F8:12` for group 7, `F8:15`/`F8:11` for group 6). `F8:2` is floored at `F8:1`. Consumed by LAD_6_MOTOR_WHL.
- Accel/decel distances: `N7:29 <- N7:50`; `N7:28 <- N7:51` if `N11:24 < 1000`, else `N7:52`; `N7:27 = N7:29 + N7:28`. If the move (`N11:24`) exceeds `N7:27`, accel counts `L12:19 = N7:29 × F8:0` and decel counts `L12:11 = N7:28 × F8:0`. For short moves, accel is 40 % of `L12:10` and decel is the remainder. `L12:12 = L12:19 + L12:11`.
- Final-approach: `N7:8 = F8:5 × F8:0`. If `L12:18 ≥ N7:8`: `L12:10 = L12:18 − N7:8`, `L12:29 = L12:18 − 500`; otherwise `L12:10 = 0`.
- Profile breakpoints (rung 10): `L12:13 = L12:19/3`, `L12:14 = L12:19`, `L12:15 = L12:18 − L12:11` (decel start), `L12:17 = 2/3 × L12:11`, `L12:16 = L12:15 + L12:17`. Consumed by LAD_6_MOTOR_WHL, LAD_9_RADIO_IN, LAD_11_BARCODE.
- Move-complete window (rung 15, unconditional): `L12:28 = (L12:18 − F8:31 × F8:0) / F8:0`, i.e. commanded distance minus `F8:31` units. Move complete for codes 19/20 is only accepted when travelled distance `F8:17` (from LAD_10_RADIO_OUT) exceeds `L12:28`.

Cam settle (rungs 11–12) When LAD_7_MOTOR_CAM reports the cam positioned (`N7:11/1`) and the position feedback matches the selected axis group, `T4:28` times 0.5 s. On timeout, `B3:7/1` is unlatched and `T4:11` is reset one-shot (both `B3:7/1` and `T4:11` are cam-motion status/timer items owned by LAD_7_MOTOR_CAM and LAD_13_FAULTS).

Command sequences (rungs 13, 14, 16–18) Each sequence rung seals itself with a scan-coil (`B3:5/0`, `B3:5/1`, `N7:12/3`, `N7:12/4`, `B3:3/2`) and runs until motion complete `N7:12/0` is latched.

- Code 5: request cam move `N7:10/11`; complete when cam positioned and settle timer done.
- Code 8: request lift move `N7:10/13`; complete when LAD_8_MOTOR_LIFT reports `N7:10/15`.
- Code 19: request cam `N7:10/11` (subject to direction sensor check); after cam settle and wheel-drive ready `B9:0/2` (from LAD_3_POWER), request wheel move `N7:10/12`; after wheel move complete `N7:11/6` (from LAD_6_MOTOR_WHL), request lift; complete when lift done and `F8:17 > L12:28`.
- Code 20: request lift first; after `N7:10/15`, request cam (with sensor check), then wheels after settle and `B9:0/2`; complete when wheels done and `F8:17 > L12:28`.
- Code 22: request wheel move `N7:10/12` directly (sensor check and `B9:0/2`), with no cam positioning or settle; complete on `N7:11/6`. No `L12:28` window is applied.

Move-complete messaging (rungs 19–22) On `N7:12/0`, `B3:4/2` seals in and a transmit request `B9:2/3` is issued. `T4:15` (5 s) re-triggers the request every 5 s until LAD_9_RADIO_IN raises `B3:3/1` (inferred acknowledgement), which drops `B3:4/2`. `N7:11/12` records that the message was issued. Rung 20 resets counter `C5:5` on acknowledgement or on its own DN bit.

Reset (rungs 21, 23) `B3:3/5` is a one-scan pulse on the rising edge of E-stop OK. Reset occurs when (a) the complete message was issued and the seal `B3:4/2` has dropped (acknowledged), (b) `B3:3/5` fires, or (c) `N7:11/10` (external abort/reset, written by LAD_3_POWER, LAD_10_RADIO_OUT, LAD_13_FAULTS) is on. Reset clears `T4:11` and the whole words `N7:10`, `N7:11`, `N7:12`, `N7:13`, which removes every motion request, status and complete bit — including bits written by the motor routines.

Interlocks and permissives

- **E-stop OK** `B3:4/0` (from LAD_4_IO_MAP): required for all command decoding, parameter calculation and sequencing. Loss unlatches the in-process bit; the rising edge on reset clears all motion words. This is PLC-level logic only; it does not replace the hardwired E-stop circuit.
- **No active fault** `B30:0/0` (from LAD_13_FAULTS): required to start a new sequence. A fault does not itself abort a running sequence in this routine; abort comes via `N7:11/10` or `B3:4/0`.
- **Wheel drive ready** `B9:0/2` (from LAD_3_POWER): required to issue the wheel move request `N7:10/12` in all three wheel sequences.
- **Cam positioned and settled**: for codes 19 and 20, the wheel request waits for `N7:11/1` from LAD_7_MOTOR_CAM plus the 0.5 s settle timer `T4:28`. Code 22 bypasses this.
- **Direction sensor check**: when `B3:5/13` (lift status from LAD_8_MOTOR_LIFT) is off, the sensor bit for the selected direction must be clear — `B3:12/0` (dir 1), `B3:12/2` (dir 2), `B3:12/3` (dir 3), `B3:12/6` (dir 4), all from LAD_4_IO_MAP — before the cam request (codes 19/20) or wheel request (code 22) is latched. When `B3:5/13` is on the check is bypassed. Once latched, the request is not withdrawn if a sensor trips.
- **Sequence ordering**: lift request in code 19 waits on wheel complete `N7:11/6`; wheel request in code 20 waits on lift complete `N7:10/15`.
- **External abort** `N7:11/10`: clears all motion requests immediately.

Alarms and faults

None generated in this routine. The routine supplies `L12:9` (distance + `N7:60`) and the request/complete bits `N7:10/x`, `N7:12/x` that LAD_13_FAULTS uses for over-travel and timeout supervision.

Review items

- Client comment on rung 4 (quoted): "Set maximum speed according to distance moved … This has generally decreased pallets per hour and is best removed." The logic is still active and caps `F8:7` at 9.0. Plant engineer to decide whether the speed-vs-distance limit is carried into the migrated program.
- VERIFY: Command code assignments (5, 8, 19, 20, 21, 22) and parameter meanings (`N11:24`, `N11:25`, `N11:27`, `N11:28`) against the traffic-controller radio protocol. Code 21 is referenced only as an exclusion.

- VERIFY: Meaning of `B3:5/13` (lift status). It both selects a speed multiplier and bypasses the direction sensor check. Confirm the bypass is intended and safe.
- VERIFY: Meaning of direction sensor bits `B3:12/0`, `/2`, `/3`, `/6` and confirm each is mapped to the correct travel direction 1–4.
- VERIFY: `N10:9` is loaded as a whole word (target lift height) but LAD_8_MOTOR_LIFT reads only bits `N10:9/0` and `N10:9/1`. Confirm the height value is a small position code, not an engineering value.
- VERIFY: `N7:8 = F8:5 × F8:0` stores a float product into a 16-bit integer. Confirm the product cannot exceed 32767 with the configured `F8:0` and `F8:5` values.
- VERIFY: Rung 15 divides by `F8:0` unconditionally every scan. If `F8:0` is 0 (e.g. before the first move selects `F8:45`/`F8:46`), a math error results. Confirm initial values of `F8:0`, `F8:45`, `F8:46`.
- VERIFY: Setpoint values of `F8:1`, `F8:5`, `F8:8`, `F8:11`, `F8:12`, `F8:15`, `F8:16`, `F8:31`, `F8:45`, `F8:46`, `N7:50`, `N7:51`, `N7:52`, `N7:60` — none are in the export.
- Rung 8 short-move branch uses `L12:10`, which is computed in rung 9 (later in the same scan). On the first scan of a move the accel/decel values use the previous move's `L12:10`; they self-correct on the second scan. Confirm no downstream routine latches on the first-scan values.
- `L12:29 = L12:18 − 500` is only updated when `L12:18 ≥ N7:8`; otherwise it retains the previous move's value. Consumed by LAD_6_MOTOR_WHL.
- Rung 20 resets `C5:5`, but no routine in the program increments it (digest shows the only write is this RES). Likely dead code.
- Code 22 issues the wheel move without the cam settle step and without the `L12:28` completion window. Confirm this is intended (e.g. a cam-already-positioned move).
- Rung 23 clears the full words `N7:10`, `N7:11`, `N7:12`, `N7:13`, including status bits owned by LAD_6/7/8 (`N7:11/0–/9`, `N7:12/8`, `/9`, `N7:13/0`, `/1`). Migration should preserve this word-level clear or explicitly replicate each bit.
- If a command (`B9:0/7`) is held while the E-stop is active, rung 1 re-latches `N7:10/1` immediately after rung 0 clears it; the latch is then cleared only by the E-stop-reset pulse `B3:3/5`. Behavior is consistent but worth confirming with the plant engineer.
- VERIFY: Hardwired E-stop circuit and its relationship to `I:0.0/7` / `B3:4/0` — outside this export.

Routine: LAD_6_MOTOR_WHL

Purpose

LAD_6_MOTOR_WHL controls the rover's X and Y wheel drives. It resets the wheel encoders, computes the wheel speed command (N7:4) along a distance-based profile, handles the stop/creep/centering phases at the end of a move, holds position when idle, manages the drive enable outputs, writes the analog speed references (O:1.0–O:1.3), and flashes the direction lights.

Functional description

Tag naming note. The export contains no tag descriptions. All signal meanings below are inferred from rung comments, the routine name, and cross-routine usage in the digests. Key inferences:

- B3:4/0 — system run permissive / "OK" bit (written in LAD_4_IO_MAP, read by nearly every routine).
- B30:0/0 — fault summary (written in LAD_13_FAULTS).
- N7:10/1 — move sequence active; N7:10/12 — wheel move in progress (both set in LAD_5_SEQ_MOV; N7:10/12 is cleared here at move complete).
- N7:10/2–/5 — four commanded travel directions; N7:10/6, /7 — axis select; N7:10/8, /9, /10 — stop-type selectors.
- B3:5/9, B3:5/10 — cam-in-position for the two wheel orientations (written in LAD_7_MOTOR_CAM). N7:1 — cam position (written in LAD_7_MOTOR_CAM). N7:12/9 — cam in motion (LAD_7_MOTOR_CAM).

- N14:70/x — mapped track/position flag sensors (written in LAD_4_IO_MAP). N14:70/0, /7 belong to one axis; N14:70/6, /13 to the other; N14:70/2, /9 are pallet-edge sensors; N14:70/12, /8, /5, /1 are per-direction stop sensors.
- L12:0 — distance travelled this move (absolute encoder counts). L12:10–L12:18, L12:29 — profile breakpoints (written in LAD_5_SEQ_MOV). L12:2 — drift from the zero reference while idle.
- N7:4 — wheel speed command (integer, written to analog outputs). F8:1 / F8:2 — low / high speed setpoints; F8:0 — counts per inch scaling.
- B3:10/2 — Mobile View (HMI manual) mode; B3:10/5–/8 — manual direction buttons (B3:10/2 written in LAD_12_MBLVIEW; B3:10/5–/8 are not written by any routine).
- N11:23 — radio command code (LAD_9_RADIO_IN / LAD_10_RADIO_OUT); 152 is a stop command per rung comment. N11:39 — barcode-derived value (LAD_11_BARCODE).
- F39:28 — measured velocity (LAD_14_VELOCITY).

1. Encoder reset (rungs 0–2). Two one-shot reset requests are generated: B3:9/0 on the rising edge of (B9:0/5 from LAD_3_POWER, or N7:12/0 from LAD_5_SEQ_MOV) while B3:4/2 is off; B3:9/1 on the rising edge of "move sequence not active" (N7:10/1 off) while B3:4/2 is off. Whenever the system is not OK (B3:4/0 off), a reset request is present, or the fault summary B30:0/0 is on, the routine zeroes both wheel encoder accumulators (HSC:0.ACC, HSC:3.ACC), the distance registers L12:0 and L12:1, and the distance-message counter N7:59.

2. Auto hold position (rungs 3–7). When no move is in progress, L12:2 tracks the encoder of the axis whose cam position is active (B3:5/9 -> HSC:0; B3:5/10 -> HSC:3). If hold is enabled (B3:8/0) and the drift is between 50 and 250 counts in either direction, a hold correction (B3:3/4) is latched and the speed command is preset to 400. There are two entry paths:

- Idle path: sequence not active, and N11:39 is 0 or ≥ 9000.
- In-move path: sequence active, cam not moving, and one of the axis flag pairs is not fully made (B3:5/10 with N7:10/7 and flags N14:70/0, /7; B3:5/9 with N7:10/6 and flags N14:70/6, /13).

While the correction is active, N7:12/8 is on (consumed by LAD_3_POWER), and rung 14 ramps N7:4 up by 1 per scan (to a ceiling of ~1000) whenever measured velocity F39:28 is below 2.0, so the wheels overcome stiction. When both flags of the active axis are seen (or T4:16 is already timing), the encoders and L12:2 are zeroed on the first scan and T4:16 holds this condition for 5 s. The current drift is copied to L12:23 while holding.

The hold correction is released (B3:3/4 unlatched) when any of: system not OK; idle and drift back within ±15 counts (with the same N11:39 condition); sequence active, T4:16 timing and drift ≥ 20 counts; T4:48 20-s timeout expires (T4:48 runs whenever the sequence is active); a wheel move starts; neither cam position bit is on; or drift exceeds ±150 counts. Rung 7 also sets B3:6/12 / B3:6/13 when the cam position is within 2700–3300 / 700–1300 during a hold correction.

3. Motion detection support (rung 8). While moving, L12:0 = |HSC:0.ACC| (axis N7:10/7) or |HSC:3.ACC| (axis N7:10/6). When not moving, L12:1 is cleared. The comparison of L12:0 against L12:1 that detects loss of motion is not in this routine; L12:1 is written and both values are read in LAD_13_FAULTS.

4. Distance-moved message (rungs 9–10). N7:57 = F8:0 × N7:56 (counts per message interval), N7:58 = L12:0 ÷ N7:57 (intervals completed). Each time N7:58 exceeds the last reported value N7:59 during a move (and B9:15/14 is off), the routine latches message triggers B9:2/3 and B9:2/1 (consumed by LAD_10_RADIO_OUT) and stores the new count.

5. Drive reset and drive enable (rungs 11–13). When the system is OK and the radio command is not 152, T4:30 times 1 s, then T4:7 times a further 1 s. Drive enables are latched outputs: O:0.0/2 (axis of N7:24–N7:25 cam window) and O:0.0/3 (axis of N7:49–N7:48 cam window). Each rung latches its own enable and unlatches the other. A rung is true when T4:30 is enabled, T4:7 is not timing, and either: (auto) the cam position N7:1 is inside that axis's window; (Mobile View) the corresponding manual buttons are pressed (B3:10/5 or /7 for O:0.0/2; B3:10/6 or /8 for O:0.0/3); or T4:7 has not yet finished (start-up window). See Interlocks for the start-up toggle behavior and latching consequences.

6. Wheel velocity profile (rungs 14–20). During a move in normal mode (B3:3/2 off), rung 15 first selects the acceleration/deceleration shaping parameters F8:20/F8:21: F8:25/F8:26 when B3:5/13 (lift state, from

LAD_8_MOTOR_LIFT) is on; otherwise F8:27/F8:28 for axis N7:10/6 or F8:29/F8:30 for axis N7:10/7. N7:4 is then computed from distance travelled L12:0:

- L12:0 ≤ L12:13: accelerate — N7:4 = (F8:2−F8:1)·F8:20·(L12:0/L12:14)² + F8:1; B9:15/0 on.
- L12:13 ≤ L12:0 ≤ L12:14: finish acceleration — N7:4 = (F8:2−F8:1)·(1 − 1.125·((L12:14−L12:0)/L12:14)^F8:21) + F8:1; B9:15/1 on.
- L12:14 < L12:0 < L12:15: no update — speed holds at the last value (cruise).
- L12:15 ≤ L12:0 ≤ L12:16: decelerate — N7:4 = (F8:1−F8:2)·1.125·((L12:0−L12:15)/L12:11)^F8:21 + F8:2, minimum 1000; B3:12/10 on.
- L12:16 ≤ L12:0 ≤ L12:18: same decel formula, minimum 1000; B3:12/11 on; while L12:0 ≤ L12:10, N7:86 captures the current speed and N7:85 is set to 1000 (used by rung 18).

B9:15/0, /1, B3:12/10, /11 are phase flags consumed by LAD_14_VELOCITY and LAD_9_RADIO_IN. Overrides: in fixed-speed mode (B3:3/2 on) N7:4 = F8:13 (rung 16); short moves (L12:18 ≤ 450 counts) run at 1000 (rung 17). Rung 18: with a stop type selected, once L12:0 ≥ L12:10 (or N11:39 > 0 and L12:0 ≥ L12:29), N7:4 ramps linearly from N7:86 down toward N7:85 between L12:10 and L12:18 (minimum 1000) and the creep phase flag N7:11/2 is latched. Rung 19: in fixed-speed mode the creep flag is latched when the commanded direction's stop sensor (N14:70/12, /8, /5, /1) is not on. Rung 20 sets N7:16 to 4 and N7:74 = N7:75 when N7:10/10 is selected, otherwise 5 and N7:76.

7. Stop-type completion (rungs 21–22, 28). Once creep is active, T4:21 runs (3.2 s). After 30 ms of creep (or immediately in fixed-speed mode), when the commanded direction's track flag is made — or, when N11:39 > 0, the opposite flag is not made — N7:4 drops to 5 and the stop-done flag for the selected stop type is latched (N7:10/8 -> N7:11/3, N7:10/10 -> N7:11/5, N7:10/9 -> N7:11/4). Rung 28 then latches move-complete N7:11/6 (consumed by LAD_5_SEQ_MOV) and clears the move-in-progress bit N7:10/12.

8. Centering under a pallet (rungs 23–27). Enabled by B3:8/1 and active during a move with N7:11/7 on. Pallet-edge sensors N14:70/2 and N14:70/9 latch "seen" flags N7:13/0 and N7:13/1. Rung 25 completes the centering (latches N7:11/5, N7:4 = 10) when lift-related conditions are met (B3:3/10 off, or B3:5/14 on, or B3:3/3 with B3:5/13 off, or B3:3/3 with B3:5/13 on and both edges seen). Rung 26, with B3:5/13 on, sets the search direction: only sensor 2 seen -> N7:11/9; only sensor 9 seen -> N7:11/8; speed N7:4 = N7:26; T4:19 times 12 s. Rung 27 is intended to reverse the search direction when T4:19 expires (see Review items).

9. Mobile View manual speed (rungs 29–30). In Mobile View, N7:68 starts at 200 with no button pressed and increases by 5 every ~25 ms (T4:23 self-resetting) while a direction button is held, up to N7:33.

10. Analog output control (rungs 31–34). B3:5/7 is the direction bit: during a move it is on for directions N7:10/2 and /3 (unless overridden by N7:11/8 or N7:12/13), or when N7:11/9 or N7:12/14 is on. During a hold correction it is chosen from the enabled drive and the flag state, with seal-in. Rung 32 (auto, not Mobile View, system OK, moving or hold-correcting with a cam position active): the axis of N7:10/7/B3:5/9 receives N7:4 on O:1.0 (B3:5/7 on) or O:1.1 (off), with the other channel zeroed; the axis of N7:10/6/B3:5/10 receives N7:4 on O:1.2 (B3:5/7 on) or O:1.3 (off). Rung 33 (Mobile View, system OK): each button writes N7:68 to its channel (B3:10/5 -> O:1.0, /7 -> O:1.1, /6 -> O:1.2, /8 -> O:1.3) and zeroes the paired channel. Rung 34 zeroes all four channels when the system is not OK, or (not in Mobile View) when the move is complete or when neither moving nor hold-correcting.

11. Direction lights (rungs 35–36). While moving, a 500 ms on / 500 ms off flasher (T4:8/T4:10) drives B3:6/4–/7 for directions N7:10/2–/5. These bits are mapped to physical outputs in LAD_4_IO_MAP.

Interlocks and permissives

- **B3:4/0 (system OK, inferred)** is the master permissive for analog speed outputs (rungs 32, 33), forces all four analog outputs to zero when off (rung 34), zeroes the encoders when off (rung 2), and gates the drive-reset timers (rung 11), hold seal-in, distance messaging, hold ramp and direction lights.
- **Drive enables O:0.0/2 and O:0.0/3 are latched (OTL/OTU) and are written only in this routine.** Loss of B3:4/0, the fault summary B30:0/0, or the 152 stop command resets T4:30 and makes rungs 12–13 false — which leaves both enables in their last state. Nothing in this routine de-energizes the drive enables on a fault or stop. **VERIFY:** confirm that the drive enable outputs are dropped by hardware (E-stop circuit) or by another

mechanism, and that this is acceptable for the migration.

- **Cam position window** (N7:1 within N7:24–N7:25 or N7:49–N7:48) is required to enable the corresponding drive in automatic mode. The two rungs are mutually exclusive: enabling one axis unlatches the other.
- **Mobile View (B3:10/2)** blocks automatic speed output (rung 32) and enables manual output (rung 33). In Mobile View with no button pressed, no rung writes the analog outputs (rung 34 excludes Mobile View), and the drive enable latched by the last button press remains latched. **VERIFY:** confirm how the analog outputs and drive enables are brought to zero when a Mobile View button is released (e.g., LAD_12_MBLVIEW clearing B3:10/2, or HMI behavior). This is safety-relevant.
- **Hold correction** requires enable B3:8/0, no move in progress, cam not moving (in-move path), a cam position bit active, and drift within ±250 counts; it is released on move start, cam position lost, 20 s timeout, or drift beyond ±150 counts.
- **Centering** requires enable B3:8/1 and N7:11/7.
- **Distance messages** are inhibited by B9:15/14.
- **Radio command 152** holds the drive-reset timers reset (rung 11).

Alarms and faults

None generated in this routine. The routine supplies motion data (L12:0, L12:1) that LAD_13_FAULTS uses for motion detection, and hold-correction status N7:12/8 that LAD_3_POWER reads. Fault summary B30:0/0 (from LAD_13_FAULTS) only resets the encoders here; it does not directly zero the analog outputs or drop the drive enables in this routine.

Review items

- VERIFY: Drive enable outputs O:0.0/2, O:0.0/3 are latched and never unlatched on fault, E-stop, or stop command by this routine (see Interlocks). Confirm hardware de-energization path.
- VERIFY: In Mobile View, releasing a button leaves the analog outputs and latched drive enable at their last values unless B3:10/2 is cleared elsewhere. Confirm LAD_12_MBLVIEW / HMI behavior.
- VERIFY: Start-up "toggle HIGH -> LOW -> programmed state" (rungs 12–13). During the first second both rungs are true because T4:7/DN is off; rung 13 executes last, so the net result is O:0.0/3 latched on and O:0.0/2 off. Both outputs are then frozen while T4:7 times, and follow the cam window after. The comment's described toggle of both enables does not appear to be realized. Confirm the AMC drive fault-clear requirement is met.
- VERIFY: Rung 13 LIM uses N7:49 as low limit and N7:48 as high limit. If N7:49 > N7:48 the test is true *outside* the window (wrap-around). Confirm the configured values and the intended window.
- VERIFY: Rung 5 pairs B3:5/10 with N7:10/7 and B3:5/9 with N7:10/6, whereas rungs 3, 8 and 32 pair B3:5/9 with N7:10/7/HSC:0 and B3:5/10 with N7:10/6/HSC:3. If both axis conditions are true at once, rung 32 writes speed to both axes. Confirm whether the rung 5 cross-pairing is intentional.
- VERIFY: Rung 5 uses `LIM 50 L12:2 -50` (true when |drift| ≥ 50) inside `LIM -250 L12:2 250`, while rung 6 releases the hold when |drift| > 150. Effective hold band is 50–150 counts; confirm intent.
- VERIFY: Rung 27 direction reversal. Both branches execute in the same scan, so whenever either N7:11/8 or N7:11/9 is set the result is always N7:11/9 on / N7:11/8 off. It does not alternate. T4:19/DN is also true for only one scan because the TON in rung 26 is gated by XIO T4:19/DN and self-resets. Confirm intended centering-search behavior.
- VERIFY: N7:11/7 (centering permissive), N7:12/13 and N7:12/14 (fine-positioning direction overrides) are only written by whole-word writes of N7:11 / N7:12 in LAD_5_SEQ_MOV. Confirm they can actually be set.
- VERIFY: B3:8/0, B3:8/1, B9:15/14, B3:10/5–/8, N7:24–N7:26, N7:33, N7:48, N7:49, N7:56, N7:75, N7:76, F8:1, F8:2, F8:13, F8:25–F8:30 are not written by any routine. They are treated here as configuration/HMI values; obtain their configured values and HMI mappings.
- VERIFY: Rung 9 executes unconditionally. If N7:57 (= F8:0 × N7:56, float into integer) is zero the DIV faults with divide-by-zero; LAD_2_MAIN clears S:5/0, which may be masking this. Also check for integer overflow of N7:57.
- VERIFY: N7:4 is an integer computed from floating-point CPT expressions with fractional exponents (F8:20, F8:21). Confirm the analog output module scaling (O:1.0–O:1.3) and the meaning of the 1000 minimum clamp

versus the 400/5/10 values used for hold and stop.

- VERIFY: Rung 18 integer expression `(N7:85 − N7:86) | (L12:18 − L12:10)` truncates when the numerator is smaller than the denominator, giving a zero ramp. Because N7:86 ≥ 1000 and N7:85 = 1000 the result is always ≤ N7:86 and clamps to 1000; confirm intended behavior.
- Likely dead or diagnostic-only: B3:4/1 (rung 4) is only read by its own seal-in; B3:6/12, B3:6/13 (rung 7), L12:23 (rung 6), N7:89 (rung 14), N7:16 and N7:74 (rung 20) are written but not read by any routine in the program. They may be HMI display values. VERIFY: confirm HMI usage before dropping in migration.
- Rung 15 gap: between L12:14 and L12:15 the speed command is not updated. This is presumably the cruise segment; confirm the breakpoint values written by LAD_5_SEQ_MOV always satisfy L12:13 ≤ L12:14 ≤ L12:15 ≤ L12:16 ≤ L12:18 and L12:16 ≤ L12:10 ≤ L12:18 (needed for N7:86 to be captured).
- Timer accumulator values in the export (T4:30 ACC 1001, T4:7 ACC 1000, T4:48 ACC 7895) are runtime snapshots, not configuration; ignore for migration.

Routine: LAD_7_MOTOR_CAM

Purpose

LAD_7_MOTOR_CAM positions a single motor-driven axis (named "CAM" in the routine name) using a high-speed-counter position feedback, an automatic move-to-window sequence with pulsed slow approach, and manual moves from the operator interface. It drives the CAM run output O:0.0/8 and the CAM direction output O:0.0/9.

VERIFY: The physical device is not identifiable from the export (camera positioner, mechanical cam, or other). No tag descriptions exist for any address in this routine; every purpose below is inferred from name and usage.

Functional description

Position feedback and homing (rungs 0–1)

- Input I:0.0/6 is inferred to be the CAM home/reference switch. While it is on, B3:2/7 (inferred "CAM at home") is set and the high-speed counter accumulator HSC:1.ACC is forced to 0 every scan.
- While the switch is off, HSC:1.ACC is copied to N7:1 (inferred "CAM position"). N7:1 is not refreshed while the switch is on.
- N7:1 is compared against ten LIM windows whose bounds are setpoints in N7:24, N7:25 and N7:40–N7:49. The resulting flags are used throughout the routine and by the sibling routines LAD_5_SEQ_MOV, LAD_6_MOTOR_WHL, LAD_13_FAULTS and LAD_14_VELOCITY:

Flag	Window (low..high)	Inferred meaning
B3:5/9	N7:40..N7:41	At target position A
B3:7/6	N7:24..N7:40	Approach band, low side of A
B3:7/5	N7:41..N7:25	Approach band, high side of A
B3:3/13	N7:24..N7:25	Anywhere in the A region (target + both approach bands)
B3:5/10	N7:42..N7:43	At target position B
B3:7/8	N7:49..N7:42	Approach band, low side of B
B3:7/7	N7:43..N7:48	Approach band, high side of B
B3:3/14	N7:49..N7:48	Anywhere in the B region
B3:5/11	N7:44..N7:47	End-of-travel window D (stops manual and park moves)
B3:5/12	N7:45..N7:46	End-of-travel window C (stops manual and park moves)

VERIFY: Setpoint values N7:24, N7:25, N7:40–N7:49 are not in the export. The table assumes each LIM low bound is less than or equal to its high bound (N7:24 < N7:40 < N7:41 < N7:25; N7:49 < N7:42 < N7:43 < N7:48). If any pair is reversed, the SLC LIM instruction inverts and the flag is true *outside* the window. VERIFY: HSC:1 configuration (count mode, direction sense, scaling) is not in the export.

Command bits from the sequencer (inferred)

The following bits are written by LAD_5_SEQ_MOV and consumed here: N7:10/7 (inferred "move CAM to position A"), N7:10/6 (inferred "move CAM to position B"), N7:10/11 (inferred "automatic CAM move active"), N7:10/14 (inferred "CAM park/return-to-end move"), N7:10/1 (inferred system run/auto permissive, read by almost every routine) and N7:10/12 (inferred system stop/fault, also written by LAD_3_POWER and LAD_6_MOTOR_WHL). See the LAD_5_SEQ_MOV section for their true definitions.

Automatic move enable (rung 3)

N7:12/9 (inferred "CAM auto-move enable") latches when all of the following hold for 0.5 s (T4:35):

- N7:10/12 is off;
- one of: at B with wheel status N14:70/0 and N14:70/7; at A with N14:70/6 and N14:70/13; at B with B command N7:10/6; at A with A command N7:10/7; or B3:7/1 (inferred "CAM reposition request", written by LAD_5_SEQ_MOV and LAD_13_FAULTS);
- N11:39 is between 1 and 8999 (N11:39 is produced by LAD_11_BARCODE);
- N7:10/11 or N7:10/14 is on, and N7:10/1 is on.

N7:12/9 is unlatched (rungs 11 and 12) when the move completes (N7:11/1), when N7:10/1 drops, or when N7:10/12 is set. N14:70 bits are produced by LAD_4_IO_MAP and LAD_6_MOTOR_WHL.

Automatic run request (rung 5)

B3:4/15 (inferred "CAM auto run request") is true when (B3:4/0 or B3:10/0) is on, manual mode B3:10/2 is off, the lift motor is not running (B3:6/8 and B3:6/9 off, from LAD_8_MOTOR_LIFT), and either:

- N7:10/11 is on, N7:11/1 (done) is off, N7:12/9 is latched, and: A commanded while outside region A (B3:3/13 off), or B commanded while outside region B (B3:3/14 off), or the run output O:0.0/8 is already on (seal-in through the output); or
- B3:8/4 and N7:10/14 are on and the axis is not in either end window (B3:5/11, B3:5/12 off) — the park move.

Run output (rung 10)

O:0.0/8 (inferred "CAM motor run") requires B3:4/0 and not T4:11/DN, and then:

- Auto (B3:10/2 off): B3:4/15 on, or B3:4/15 off with the slow-approach pulse B3:6/10 on. When B3:4/15 is on, the output is additionally blanked while T4:11.ACC is 51–129 or 8001–8499 (two LIM instructions with inverted bounds, ANDed). The slow-approach path bypasses the blanking.
- Manual (B3:10/2 on): any manual move latch B3:11/4, B3:11/5, B3:11/6 or manual jog bit B3:11/10, B3:11/11.

T4:11 is the CAM move timer. It is reset here (rungs 7, 8, 9, 18) but timed in a sibling routine (written by LAD_5_SEQ_MOV and LAD_13_FAULTS); its DN bit stops the motor.

Slow approach / inching (rung 6)

Once the run request B3:4/15 drops and N7:10/12 is off, if the axis is commanded to A and sits in an A approach band (B3:7/5 or B3:7/6), or commanded to B and in a B approach band (B3:7/7 or B3:7/8), T4:14 runs for 150 ms. B3:6/10 (inferred "approach pulse") is on for the first 100 ms and B3:6/11 (inferred "slow approach active") for the full 150 ms. When T4:14 times out it is removed from the rung, resets, and re-arms. The result is a repeated 100 ms on / 50 ms off motor pulse that inches the axis until it enters the target window and the approach-band flags clear. B3:6/11 is read by LAD_13_FAULTS.

Direction output (rung 17)

O:0.0/9 (inferred "CAM direction") is:

- Auto: on while the motor runs with B3:6/10 off and B3:7/1 off; during slow approach (B3:6/10 on) it is on only when the axis is on the high side of the target (B3:7/5 without B3:7/6, or B3:7/7 without B3:7/8), i.e., it selects the correction direction.
- Manual: follows jog bit B3:11/10 directly (B3:11/11 jogs the opposite direction).

Move complete (rungs 4, 11, 12, 18)

- N7:11/1 (inferred "CAM move complete") latches after a 15 ms confirm (T4:9) when N7:10/11 is on, N7:10/12 and N7:10/14 are off, and the axis is at the commanded target (A with N7:10/7, B with N7:10/6) or N7:11/0 is set.
- For the park move (N7:10/14), reaching either end window with B3:4/0 on unlatches N7:10/14 and latches both N7:11/1 and N7:11/0 (inferred "CAM at end/park position"). Rung 12 repeats this action and also unlatches N7:12/9.
- N7:11/1 unlatches B3:7/1 and issues a one-shot reset of T4:11 (rung 18). N7:11/1 is read by LAD_5_SEQ_MOV, LAD_6_MOTOR_WHL and LAD_13_FAULTS.
- N7:10/14 also holds T4:33 reset (rung 2); T4:33 is timed and evaluated in LAD_13_FAULTS.

Wrong-end / overtravel indication (rungs 13–16)

While N7:10/11 is on and the move is not complete: B3:5/4 seals in if commanded to A but end window C (B3:5/12) is reached; B3:5/5 seals in if commanded to B but end window D (B3:5/11) is reached; B3:5/6 seals in if commanded to B while at position A. B3:4/14 = (B3:5/4 or B3:5/5) and not B3:5/6. All clear when N7:11/1 latches. Inferred as mis-travel indicators.

Manual moves (rungs 7–9, 19–20)

Manual command bits B3:11/1, B3:11/2, B3:11/3 (with qualifiers B3:5/2, B3:5/3) start sealed manual moves, each mutually exclusive with the others, each resetting T4:11 on start:

- B3:11/4: started by B3:11/3 with B3:5/2 or B3:5/3; ends at either end window.
- B3:11/5: started by B3:11/1 with B3:5/3; ends at position A.
- B3:11/6: started by B3:11/2 with B3:5/2; ends at position B.

Rung 20 sets B3:11/8 as the inverse of B3:11/7. Rung 19 derives B3:11/9 from B3:11/7, B3:11/8 and its own state in manual mode; the construct toggles B3:11/9 every scan while B3:11/7 is held (see Review items).

Status mirror (rung 21)

N7:10/11 is copied to N14:70/14. N14:70 is transmitted as a word by LAD_10_RADIO_OUT.

Interlocks and permissives

- **Run permissive B3:4/0** (produced by LAD_4_IO_MAP): required for the run output O:0.0/8 in both auto and manual, and for the park-move completion logic. B3:10/0 (produced by LAD_12_MBLVIEW) is an alternate permissive only for the auto run request B3:4/15.
- **Manual/auto select B3:10/2** (produced by LAD_12_MBLVIEW): selects the manual command set for run and direction; blocks the auto run request.
- **Lift-motion interlock:** B3:6/8 and B3:6/9 (produced by LAD_8_MOTOR_LIFT) block the auto run request. They do **not** block the manual run path in rung 10.
- **System stop N7:10/12:** blocks the auto-move enable, the done latch, and slow approach; unlatches the auto-move enable.
- **Move timeout T4:11/DN:** stops the run output in all modes. Timer is run in a sibling routine (see LAD_5_SEQ_MOV and LAD_13_FAULTS).
- **Auto-move enable N7:12/9** must be latched (0.5 s qualification, barcode value N11:39 in range) before an automatic move can start.
- **Move complete N7:11/1** blocks the auto run request.
- **Software travel windows:** automatic moves stop on entering the target region; park and manual sealed moves stop at the end windows. Manual jog (B3:11/10, B3:11/11) has **no** position window check in this routine — only T4:11/DN and B3:4/0 apply.
- No hardwired overtravel or end-of-travel limit switch is read in this routine; I:0.0/6 is used only as a home reference.

Alarms and faults

No fault bits are set in this routine. Fault detection for this axis is in LAD_13_FAULTS, which reads the position flags (B3:5/9–5/15, B3:7/5–7/8, B3:3/13, B3:6/10, B3:6/11), the manual move latches (B3:11/4–11/6), T4:11/DN and T4:33/DN. This routine contributes by resetting T4:11 on move start/complete and holding T4:33 reset during the park move.

Review items

- VERIFY: No tag descriptions exist for any address in this routine; all functional names above are inferred. Confirm at minimum I:0.0/6 (home switch), O:0.0/8 (run), O:0.0/9 (direction), N7:10/6, N7:10/7 (move to B / move to A), N7:10/14 (park move) and N7:11/1 (move complete).
- VERIFY: Setpoints N7:24, N7:25, N7:40–N7:49 are not in the export. Confirm actual values and that each LIM low bound ≤ high bound; reversed bounds invert the window flag.
- VERIFY: HSC:1 configuration and I/O wiring for I:0.0/6, O:0.0/8, O:0.0/9.
- VERIFY: B3:8/4, B3:11/1, B3:11/2, B3:11/3, B3:11/7, B3:11/10, B3:11/11, B3:5/2 and B3:5/3 have no writer in any routine digest. They are inferred to be written directly by the MobileView HMI (LAD_12_MBLVIEW) or another external device. Confirm the source; if none exists, the park-move path (B3:8/4) and all manual moves are dead.
- VERIFY: N7:11/0 is latched in rungs 11/12 and is never unlatched by any routine per the digests. After the first park move it stays on until a download or power cycle, and it then permanently satisfies one start condition of the done timer T4:9 (rung 4). Confirm intent.
- VERIFY: N7:11/1 is only latched in this routine; the unlatch must be in LAD_13_FAULTS (the only other writer). Confirm.
- Rung 10: the run output is blanked while T4:11.ACC is 51–129 and 8001–8499. Intent is not evident from the logic (possibly a brake/friction pulse or a deliberate break of the O:0.0/8 -> B3:4/15 seal-in). VERIFY: confirm the intended behavior and where T4:11 is timed.
- Rung 5: B3:4/15 seals through the run output O:0.0/8, and O:0.0/8 seals through B3:4/15 (rung 10). The loop is broken only by the blanking windows above, N7:11/1, N7:12/9 dropping, B3:4/0/B3:10/0 dropping, manual mode, lift motion or T4:11/DN. A move that enters its target region will not stop at that instant; it continues until the seal is broken. VERIFY: confirm this is acceptable stopping behavior.
- Rung 12 duplicates rung 11's first branch (plus OTU N7:12/9, which rung 11's second branch also performs once N7:11/1 is set). Redundant logic; candidate for consolidation in migration.
- Rungs 19–20: B3:11/9 toggles every scan while B3:11/7 is held in manual mode (B3:11/8 lags B3:11/7 by one scan). B3:11/9 has no reader outside this routine per the digests. Likely vestigial or dead code. VERIFY: confirm whether B3:11/7/8/9 are used by the HMI.
- B3:4/14 (rung 16) has no reader in any routine digest. VERIFY: confirm whether the HMI reads it; otherwise dead.
- Manual jog bits B3:11/10 and B3:11/11 run the motor with no software travel limit, and the manual run path does not apply the lift-motion interlock. Safety-relevant; confirm with plant engineer.
- Rung 4: T4:9 preset 15 ms; export shows ACC = 16 (timer expired at upload). Informational only.

Routine: LAD_8_MOTOR_LIFT

Purpose

LAD_8_MOTOR_LIFT positions a lift to one of three preset positions using a single analog position input (I:0.7). It selects the target position (from the sequencer in auto or from HMI buttons in manual), drives the lift motor run/direction outputs (O:0.0/10, O:0.0/11), reports "at position" status bits to the rest of the program, and performs a small drift-correction function when the lift is idle.

Functional description

No tag descriptions exist in this export. All names below are **inferred** from usage and are marked as such in the review list.

1. Position feedback (rung 2) Analog input I:0.7 is copied to N7:2 each scan. N7:2 is treated throughout as the lift position value. Rungs 0 and 1 execute before this MOV, so they act on the value of N7:2 from the previous scan (or from LAD_4_IO_MAP, which also writes N7:2 — see Review items).

2. Position windows and "at position" status (rung 0) Three status bits are produced from N7:2 against fixed setpoints stored in N7:34–N7:37 (no routine writes these; they are data-table values):

Status bit	Condition on N7:2	Inferred meaning
B3:5/13	N7:2 ≤ N7:34	Lift at Position 1
B3:5/14	N7:35 ≤ N7:2 ≤ N7:37	Lift at Position 2
B3:5/15	N7:2 ≥ N7:36	Lift at Position 3

Each status bit also requires settle timer T4:27 to be done. T4:27 (150 ms) runs while N7:2 is inside any of the three windows. The Position 1/3 check uses `LIM N7:36 N7:2 N7:34` with the low limit greater than the high limit, which by LIM definition is true when N7:2 is *outside* the range N7:34…N7:36 — i.e. at or beyond either end position. When the lift leaves all windows the timer resets and the status bits drop immediately. These status bits are consumed by LAD_5_SEQ_MOV, LAD_6_MOTOR_WHL, LAD_7_MOTOR_CAM and LAD_13_FAULTS.

3. Target position selection (rungs 3–5) Three mutually exclusive selection bits are produced: B3:3/9 (Position 1), B3:3/10 (Position 2), B3:3/11 (Position 3). All three are forced off while either manual jog button (B3:10/14, B3:10/15) is held.

- *Auto mode* (B3:10/2 off): selection follows a two-bit code from the sequencer in N10:9 (written by LAD_5_SEQ_MOV): N10:9/0 only = Position 1; N10:9/1 only = Position 2; both = Position 3; neither = no selection. There is no seal-in; the selection follows the code directly.
- *Manual mode* (B3:10/2 on, written by LAD_12_MBLVIEW): momentary HMI buttons B3:10/10, /11, /12 select Position 1/2/3. Each selection seals in and is cleared when another position button is pressed or a jog button is pressed.

4. Motion command generation (rungs 6–7) Two direction commands are produced. B3:6/8 drives the lift in the direction that *decreases* N7:2 (toward Position 1); B3:6/9 drives in the direction that *increases* N7:2 (toward Position 3). Both require the global permissive B3:4/0 (written by LAD_4_IO_MAP).

Positioning path — active when either the sequencer lift step N7:10/13 is on (auto) or manual mode B3:10/2 is on, and the in-position acknowledge N7:10/15 is not latched:

- B3:6/8 is commanded when Position 1 is selected and not yet reached, or when Position 2 is selected and the lift is above the window (N7:2 > N7:37). Once started toward Position 2 it seals in until B3:5/14 (at Position 2) is true.
- B3:6/9 is commanded when Position 3 is selected and not yet reached, or when Position 2 is selected and the lift is below the window (N7:2 < N7:35), with the same seal-in to B3:5/14.

Correction path — B3:7/12 forces B3:6/8 and B3:7/11 forces B3:6/9 (see item 6). This path bypasses the N7:10/13 / B3:10/2 / N7:10/15 conditions but still requires B3:4/0.

Travel-end checks — B3:6/8 is only permitted while N7:2 ≥ N7:34; B3:6/9 only while N7:2 ≤ N7:36. Both command bits are also read by LAD_7_MOTOR_CAM and LAD_13_FAULTS.

5. Motor outputs and manual jog (rungs 8–9)

- O:0.0/10 (inferred: lift motor run) energizes when either (B3:6/8 or jog B3:10/15) and not at Position 1, or (B3:6/9 or jog B3:10/14) and not at Position 3.
- O:0.0/11 (inferred: lift direction) energizes when (B3:6/8 or jog B3:10/15) and not at Position 1. It is off for the B3:6/9 direction.

HMI jog buttons B3:10/15 (toward Position 1) and B3:10/14 (toward Position 3) act directly on the outputs. They are stopped only by the corresponding "at position" bit; they are **not** gated by B3:4/0, by manual mode B3:10/2, or by any other permissive in this routine.

6. Idle drift correction (rung 1) When B3:8/2 (inferred: correction enable; written by no routine) and N7:10/1 (inferred: system ready/run status from LAD_5_SEQ_MOV) are on, the system is not in manual mode, and the sequencer lift step N7:10/13 is not active, the routine compares N7:2 against intermediate thresholds N7:14 and N7:15 (inferred: decision points between adjacent positions):

- B3:7/11 (correct in the increasing direction) when N7:14 ≤ N7:2 ≤ N7:35 or N7:15 ≤ N7:2 ≤ N7:36.
- B3:7/12 (correct in the decreasing direction) when N7:34 ≤ N7:2 ≤ N7:14 or N7:37 ≤ N7:2 ≤ N7:15.

Intent appears to be returning a lift that has drifted out of a position window to the nearest window. Timer T4:32 (250 ms) runs while a correction bit is on and its done bit removes the correction; however, because rung 1 is then false the timer resets on the next scan and correction resumes. As written, T4:32 produces a one-scan interruption every 250 ms rather than a hard limit on correction time (see Review items).

7. Sequencer handshake (rung 10) While the sequencer lift step N7:10/13 is on and the selected position matches the corresponding "at position" bit, N7:10/15 is latched (OTL). N7:10/15 then blocks further positioning moves (rungs 6–7). It is not unlatched in this routine; LAD_5_SEQ_MOV writes the whole N7:10 word and reads N7:10/15 — see the LAD_5_SEQ_MOV section for when it is cleared.

8. B3:11/12–14 (rungs 11–12) Rung 12 sets B3:11/13 = NOT B3:11/12 every scan. Rung 11, in manual mode, uses B3:11/12 with the previous-scan value of B3:11/13 to produce B3:11/14. Traced scan by scan, B3:11/14 toggles on and off every scan while B3:11/12 is held. Neither B3:11/13 nor B3:11/14 is read by any other routine per the cross-reference. B3:11/12 is not written by any routine (likely an HMI button). This construct has no effect on lift motion.

Rung 13 is the subroutine return.

Interlocks and permissives

Listed in priority order for the plant engineer.

1. **Manual jog bypasses permissives.** HMI jog bits B3:10/14 and B3:10/15 drive O:0.0/10 and O:0.0/11 directly (rungs 8–9). The only stop condition is the analog-derived "at position" bit at the respective end (B3:5/13 or B3:5/15). Global permissive B3:4/0 and manual-mode bit B3:10/2 are not checked for jog. No hardwired limit switch appears in this routine.
2. **Global permissive B3:4/0** (from LAD_4_IO_MAP; inferred as system OK/run enable) is required for all automatic, manual-positioning and drift-correction motion commands (B3:6/8, B3:6/9).
3. **Software travel limits.** B3:6/8 is blocked when N7:2 < N7:34; B3:6/9 is blocked when N7:2 > N7:36. Additionally the outputs are blocked at Position 1 (decreasing direction) and Position 3 (increasing direction) by B3:5/13 / B3:5/15. All limits depend on the single analog input I:0.7 and the 150 ms settle timer; loss or drift of the analog signal defeats them.
4. **Mode gating for positioning.** Positioning moves require either the sequencer lift step N7:10/13 (auto) or manual mode B3:10/2. Positioning is blocked while N7:10/15 (in-position acknowledge) is latched.
5. **Drift correction permissives.** Requires B3:8/2, N7:10/1, not manual mode, and sequencer lift step not active.
6. **Selection lockout.** Any jog press clears all target-position selections (rungs 3–5).

Alarms and faults

None generated in this routine. Lift status bits B3:3/9, B3:5/13–15, B3:6/8, B3:6/9 and N7:10/13/15 are read by LAD_13_FAULTS — see that section for lift-related fault detection.

Review items

- VERIFY: No tag descriptions were exported for any address in this routine. All functional names (positions 1/2/3, run/direction outputs, jog buttons, mode bits) are inferred from logic structure and must be confirmed against drawings/HMI.
- VERIFY: Direction sense. The logic is written in terms of N7:2 increasing/decreasing. Confirm which physical direction (raise/lower) corresponds to increasing I:0.7, and whether O:0.0/11 is "direction" or a second contactor.
- VERIFY: Setpoint values and ordering of N7:34, N7:35, N7:37, N7:36 and N7:14, N7:15. The logic assumes ascending order N7:34 < N7:14 < N7:35 ≤ N7:37 < N7:15 < N7:36. The rung-0 LIM (low limit N7:36, high limit N7:34) relies on N7:36 > N7:34; if this is not true the settle timer T4:27 behaves differently and the "at position"

bits will not set as described. None of these setpoints are written by any routine; confirm whether they are HMI-entered or fixed.

- VERIFY: N7:2 is also written by LAD_4_IO_MAP. If IO_MAP writes a scaled or filtered value, rungs 0–1 (executed before this routine's raw MOV) and rungs 6–7 (after it) compare against different values. Confirm the two writes are identical or reconcile them during migration.
- VERIFY: B3:8/2 (drift-correction enable) is not written by any routine. Confirm whether it is set by the HMI, forced, or permanently off (in which case rung 1 is dead logic).
- VERIFY: B3:10/10, /11, /12, /14, /15 and B3:11/12 are not written by any routine; confirm they are MobileView/HMI-written and confirm HMI momentary vs. maintained behavior for the jog bits.
- VERIFY: Overtravel protection. This routine provides only analog-derived software limits, and the manual jog path is not interlocked by B3:4/0. Confirm hardwired limit switches / drive limits exist outside the PLC, or add this to the migration risk register.
- VERIFY: Drive behavior when O:0.0/10 and O:0.0/11 are both on (this is the normal state for the decreasing direction), and when B3:6/8 and B3:6/9 are simultaneously true (possible during Position 2 overshoot, or if the sequencer changes the target mid-move while a seal-in is held).
- VERIFY: T4:32 (250 ms) does not limit total correction duration as written; it interrupts correction for one scan every 250 ms. Confirm intended behavior with the plant before replicating.
- VERIFY: N7:10/15 (in-position acknowledge) is latched here and is cleared only by LAD_5_SEQ_MOV writing the N7:10 word. In manual mode a stale latch from a prior auto step blocks manual positioning (but not jog) until the sequencer clears it. Confirm this is acceptable.
- Suspicious construct: rungs 11–12 produce B3:11/14 as a scan-rate toggle while B3:11/12 is held in manual mode. B3:11/13 and B3:11/14 are not consumed by any routine. This appears to be an abandoned or mis-built one-shot. VERIFY whether the HMI reads B3:11/13 or B3:11/14; if not, treat as dead code.
- Note: rungs 0–1 use the previous-scan value of N7:2 (rung 2 refreshes it afterward). One-scan latency; no functional impact expected but should be preserved or consciously changed during migration.
- Note: B3:7/11 and B3:7/12 are consumed only within this routine; B3:3/10 and B3:3/11 are consumed only here (B3:3/9 is also read by LAD_13_FAULTS).

Routine: LAD_9_RADIO_IN

Purpose

LAD_9_RADIO_IN validates each incoming radio command message, checks that it is addressed to this rover and carries a new transaction ID, and copies the accepted command and its data fields into the working registers that the motion sequencer (LAD_5_SEQ_MOV) and the response logic (LAD_10_RADIO_OUT) act on. It also handles the Status Query and Stop/Cancel commands directly.

Functional description

Message source and buffer. The raw incoming message occupies N11:0–N11:9 (written as a block by the LAD_12_MBLVIEW routine). Field usage inferred from the rung-2 comment and from how each word is copied:

Word	Inferred meaning
N11:0	Destination rover ID (compared to N10:2, set in LAD_2_MAIN)
N11:1	Copied to N11:21; purpose not stated (possibly source/sender ID)
N11:2	Transaction ID (1–9999, rolls over)
N11:3	Command code
N11:4	Data 1 – distance (inches) / CAM direction / lift height, depending on command
N11:5	Data 2 – stop type
N11:6	Data 3 – mode
N11:7	Data 4 – lift height

Word	Inferred meaning
N11:8	Data 5 – direction
N11:9	Data 6 – fractional distance, 1/1000 inch

Startup. On first scan the transaction-ID reference N11:19 is zeroed so the first message is accepted regardless of its transaction number.

Step 1 – Command validation (produces B3:3/1, "valid command received", one scan). A message is considered valid when N11:3 is non-zero and matches one of these patterns:

- 2 (Move Complete ACK) or 4 (Status Query): no data required.
- 5 (Change CAM position) or 8 (Change height): Data 1 must be non-zero.
- 19 (Move then Lift) or 20 (Lift then Move): Data 2, Data 4 and Data 5 must be non-zero. Data 1 (whole inches) and Data 3 (mode) are *not* checked.
- 21 (Extend Move): Data 1 must be non-zero. An extend of less than one whole inch (fractional only) therefore fails validation.
- 22 (Creep and Find): Data 1, Data 2 and Data 5 must be non-zero.
- 99 (E-Stop), 152 and 252 (Stop/Cancel): no data required.

Any other command code, or a listed code with missing required data, is silently ignored: nothing is stored and no reject is generated. Command 98 (Save Logs) is listed in the rung comment but is not in the validation list, so it is never accepted.

Step 2 – Housekeeping on the following scan. One scan after B3:3/1 is set (and also on first scan), the routine triggers data-log queue 0 (DLG 0), clears the command word N11:3 so the same message is not processed twice, and clears the error-code word N11:41 (which LAD_13_FAULTS writes). Clearing N11:3 also drops B3:3/1, making it a single-scan pulse.

Step 3 – Save header and command (rung 3). On B3:3/1: N11:20 <- rover ID, N11:21 <- N11:1, N11:23 <- command code (always). The transaction ID is copied to N11:22 only if it is non-zero and the command is not 2 or 4, so queries and ACKs do not disturb the transaction sequence. Note N11:23 retains the last command indefinitely; sibling routines (LAD_3_POWER, LAD_5_SEQ_MOV, LAD_6_MOTOR_WHL, LAD_10_RADIO_OUT, LAD_13_FAULTS) read it as a level, not an event.

Step 4 – Command acceptance (rung 4, produces B9:0/7 "command accepted", one scan). Requires all of:

1. B3:3/1 (valid message this scan).
2. Addressed to this rover: N11:20 = N10:2.
3. New transaction: N11:22 > N11:19, or rollover (N11:22 < 20 and N11:19 > 9000). A retransmitted message with the same ID is silently dropped.
4. Command-specific permissive (see Interlocks): commands 5, 8, 19, 20, 22 require no sequence in progress; command 21 has its own conditions.

Commands 2, 4, 99, 152 and 252 never set B9:0/7.

Step 5 – Latch command data (rung 9). On B9:0/7 (and command not 2, 4 or 99): N11:19 <- N11:22 (transaction now consumed), N11:24 <- Data 1 and F8:22 <- Data 1 as float, N11:25 <- Data 2, N11:26 <- Data 3, N11:27 <- Data 4, N11:28 <- Data 5, F8:23 <- Data 6 as float. These are the registers LAD_5_SEQ_MOV reads to build the move.

Step 6 – Side effects of a new non-extend command (rung 8). On B9:0/7 with command ≠ 21: latch N7:12/7 (read only by LAD_10_RADIO_OUT; inferred to be "new command, acknowledge/respond"), unlatch B3:9/15 (inferred transmit/retry flag shared with LAD_10_RADIO_OUT), and **clear all 43 words of the B30 fault file (B30:0–B30:42) to zero**.

Extend Move reject (rung 5). Latches B9:2/4 (inferred "send reject response"; unlatched in LAD_10_RADIO_OUT) when N11:23 = 21, a sequence is in progress (N7:10/1), and the rover is in a decel/creep condition — in normal mode (B3:16/0 off): F8:7 < 9.0 or B3:12/10 or B3:12/11 or N7:11/2; in alternate mode (B3:16/0 on): B3:16/1 on. This rung is not gated by B3:3/1 and evaluates every scan (see Review items).

Status Query (rung 6). On B3:3/1 with command 4, latch B9:2/3 (shared "send status message" request, also set by LAD_3, 5, 6, 10, 11, 13 and presumably served/cleared in LAD_10_RADIO_OUT). If N7:10/12 is on, the response is only requested while position L12:0 is within the window L12:13…L12:16; if N7:10/12 is off it is requested immediately. N7:10/12 is written by LAD_5_SEQ_MOV and LAD_6_MOTOR_WHL; I infer it is a moving/positioning state and the window is a position band written by LAD_5_SEQ_MOV. No rover-ID or transaction check is applied to queries.

Stop/Cancel Automation (rung 7, command 152 only). On B3:3/1 with command 152: clear all 43 words of the B30 fault file, unlatch B3:9/15, and reset N11:19 to 0 so the next message of any transaction ID is accepted. No rover-ID check is applied. The actual stopping of motion is not done here; LAD_5_SEQ_MOV and LAD_3_POWER read N11:23 and B3:3/1 — see those sections. Command 252 is validated and stored to N11:23 but has no further action in this routine.

E-Stop command (99). Validated and stored to N11:23 only. This routine takes no action on it; see the LAD_3_POWER and LAD_13_FAULTS sections, which read N11:23.

Interlocks and permissives

- **New motion sequence commands (5, 8, 19, 20, 22) are blocked while N7:10/1 is on.** N7:10/1 is written by LAD_5_SEQ_MOV and LAD_3_POWER; from the rung comment I infer it means "motion sequence in progress". A new command received during a sequence is silently dropped (no reject message).
- **Extend Move (21) is permitted only while a move is active** (N7:12/3 or N7:12/4 on, both written by LAD_5_SEQ_MOV — inferred move-phase states) **and the rover is not decelerating or creeping**: in normal mode (B3:16/0 off) F8:7 must equal 9.0 and B3:12/10, B3:12/11, N7:11/2 must all be off; in alternate mode (B3:16/0 on) N7:10/1 must be on and B3:16/1 off. F8:7 is written by LAD_5_SEQ_MOV; B3:12/10, B3:12/11 and N7:11/2 by LAD_6_MOTOR_WHL — inferred to be profile-state/decel/creep flags.
- **Addressing:** motion commands act only if N11:20 = N10:2 (this rover's ID, inferred). Queries (4), Stop/Cancel (152) and E-Stop (99) are *not* filtered by rover ID in this routine.
- **Duplicate suppression:** a command whose transaction ID is not greater than the last accepted one (with 9999->1 rollover) is ignored.
- **Safety-relevant:** the radio "E-Stop" (99) is a software command carried over the radio link. It is not a safety-rated function and is not acted on in this routine. The hardwired E-stop path is documented in the LAD_3_POWER / LAD_13_FAULTS sections.
- **Safety-relevant:** every accepted motion command (5, 8, 19, 20, 22) and every Stop/Cancel (152) clears the entire B30 fault file. This is a remote fault reset with no local acknowledgement.

Alarms and faults

- B9:2/4 (latched here) — Extend Move rejected because the rover is in a decel/creep profile. Cleared in LAD_10_RADIO_OUT.
- B30:0–B30:42 fault words are cleared (not set) by commands 152 and by any accepted motion command. Fault detection itself is in LAD_13_FAULTS.
- N11:41 error code is cleared here one scan after each valid message; it is set in LAD_13_FAULTS.
- No faults are generated by this routine for invalid, mis-addressed or out-of-sequence commands — they are dropped silently.

Review items

- VERIFY: Meaning of message words N11:0–N11:9 as tabulated above is inferred from the rung-2 comment and copy targets; no tag descriptions exist in the export. Confirm against the radio protocol specification.
- VERIFY: N10:2 is assumed to be this rover's ID (written in LAD_2_MAIN). Confirm.
- VERIFY: B3:16/0 and B3:16/1 are read here but not written by any routine in the program. Confirm whether they are HMI/configuration bits, forced, or always zero. If always zero, the alternate extend-move path in rungs 4 and 5 is dead code.

- VERIFY: F8:7 = 9.0 is used as an exact floating-point equality test to mean "not in decel profile". Confirm F8:7 is only ever assigned discrete constants in LAD_5_SEQ_MOV; otherwise this compare is fragile and a migrated platform may evaluate it differently.
- VERIFY: DLG 0 dataset contents and trigger expectations are not in the export. It fires once per accepted command (one scan later, after N11:3 has been cleared but before N11:23 is overwritten) and once on first scan.
- VERIFY: N11:23 (saved command) is also written by LAD_10_RADIO_OUT. Confirm how that write interacts with sibling routines that read N11:23 as a level (e.g., 99 / 152 detection).
- VERIFY: Inferred purposes of B3:9/15 (transmit/retry flag), N7:12/7 (new command pending response), B9:2/3 (status message request), B9:2/4 (reject response) — all shared with LAD_10_RADIO_OUT.
- Suspicious — rung 5 is not gated by B3:3/1 or a one-shot. Because N11:23 holds 21 after an *accepted* extend, B9:2/4 will latch later in the same move as soon as the rover enters decel/creep while N7:10/1 is still on, i.e. a reject message may be sent for an extend that was actually executed. Rung 5 also does not check the rover ID, so an extend addressed to another rover on the same channel can trigger a reject from this rover. Confirm whether this is intended.
- Suspicious — command 98 (Save Logs) appears in the command list comment but is not accepted by rung 2. Either the comment is stale or the feature was removed.
- Suspicious — command 252 is validated and stored but has no action in this routine (only 152 is handled in rung 7). Confirm whether 252 is consumed elsewhere or is dead.
- Suspicious — commands 99, 152 and 4 bypass the rover-ID and transaction-ID checks. This may be intended (broadcast stop / any-rover query) but should be confirmed, particularly if multiple rovers share the radio channel.
- Suspicious — full B30 fault file clear (43 words) on every accepted motion command and on Stop/Cancel. Confirm the plant accepts remote fault reset without a local acknowledge, and that no latched fault in B30 is expected to persist across commands.
- Note — rung 2 does not require Data 1 (whole inches) or Data 3 (mode) to be non-zero for commands 19/20, but does require Data 1 for command 21. Confirm intentional (e.g., fractional-only moves allowed for 19/20 but not for extend).
- Note — rung 9 excludes commands 2, 4 and 99 redundantly; rung 4 never sets B9:0/7 for those codes. Harmless.
- Note — no reject or NAK is produced for any dropped command except the extend-move decel case; the sender only learns of a dropped command by the absence of a response.

Routine: LAD_10_RADIO_OUT

Purpose

LAD_10_RADIO_OUT assembles the Rover's outbound status block (N11:30–N11:49) and sends it to the System PLC (referred to as "TC" in the rung comments) with a single Ethernet MSG write. It also processes the three network stop commands (152, 252, 99) received from the System PLC and sends the corresponding confirmation message.

Functional description

Message data path. Every scan, the routine refreshes the words of the outbound block. A send is requested by latching B9:2/3 (inferred: "send message" request; also latched by LAD_9_RADIO_IN, LAD_11_BARCODE, LAD_13_FAULTS, LAD_5_SEQ_MOV, LAD_3_POWER, LAD_6_MOTOR_WHL). The MSG instruction (MG16:2) fires when B9:2/3 is on and MG16:2/EN is off. When the message completes (MG16:2/DN) or errors (MG16:2/ER), the routine unlatches all four request bits B9:2/1, B9:2/2, B9:2/3 and B9:2/4 together.

Outbound block contents assembled in this routine (word | content | source):

Word	Content	Source
N11:30	Rover number	N10:2 (set in LAD_2_MAIN)
N11:31	Transaction ID	N11:19 (from LAD_9_RADIO_IN), copied only while N11:23 is not 2, 3, 4 or 99
N11:32	Battery status	N7:0 (from LAD_4_IO_MAP)

Word	Content	Source
N11:33/0	Power/battery status bit	follows B3:7/15 (from LAD_3_POWER; meaning inferred from location only)
N11:34	E-stop status: 0 = B3:4/0 on, 1 = B3:4/0 off	B3:4/0 (from LAD_4_IO_MAP)
N11:35/0	Lift status bit	follows B3:3/3 (from LAD_4_IO_MAP)
N11:36	Lift height band: 1 = N7:2 < N7:34, 2 = N7:34 ≤ N7:2 ≤ N7:36, 3 = N7:2 > N7:36	N7:2 lift position (from LAD_4_IO_MAP / LAD_8_MOTOR_LIFT); N7:34, N7:36 are threshold constants not written by any routine
N11:37	Movement status code (see below)	fault and request bits
N11:38	CAM position: 1 = B3:5/9 on, 2 = B3:5/10 on, 3 = neither	B3:5/9, B3:5/10 (from LAD_7_MOTOR_CAM)
N11:42	Distance moved, whole inches	computed here
N11:43	Raw input word I:0.0	direct copy
N11:44, N11:45	Mapped sensor status words	N14:70, N14:71 (from LAD_4_IO_MAP)
N11:46	Distance moved, thousandths of an inch	computed here
N11:49	PLC Ready flag, constant 1	constant

Words N11:39, N11:40, N11:41, N11:47, N11:48 are within the transmitted block but are not written by this routine (N11:39 is written in LAD_11_BARCODE; N11:41 in LAD_9_RADIO_IN and LAD_13_FAULTS).

Transaction ID hold. The transaction ID is copied to N11:31 only when the current command word N11:23 is not 2, 3, 4 or 99. The effect is that N11:31 is frozen while one of these commands is active, so the eventual reply carries the ID of that command. The rung comment describes this as copying "if executing a motion related command", which is the inverse of the logic as written; the behavior above is what the logic does.

Distance moved calculation. Runs only while N7:10/12 is on (inferred: a move/sequence-active flag set in LAD_5_SEQ_MOV and LAD_6_MOTOR_WHL). F8:17 = L12:0 / F8:0 (encoder counts ÷ counts-per-inch; L12:0 from LAD_6_MOTOR_WHL, F8:0 from LAD_5_SEQ_MOV). F8:17 is moved to integer N7:80 (rounded by the MOV), then adjusted to truncate toward zero: if the rounded value overshoots the float it is stepped back by one. Result is N11:42 = whole inches (signed, truncated toward zero) and N11:46 = (F8:17 − N11:42) × 1000 = thousandths remainder. When N7:10/12 is off the last values hold.

Movement status code (N11:37). Branches are evaluated in order; a later true branch overrides an earlier one:

- 0: no fault (B30:0/0 off) and N7:12/7 on
- 1: no fault and (N7:12/7 off or B3:4/2 on)
- 2: fault active (B30:0/0 on, from LAD_13_FAULTS)
- 4: B9:2/2 on and N7:12/7 on (B9:2/2 latched by LAD_11_BARCODE)
- 5: B9:2/1 on and N7:12/7 on (B9:2/1 latched by LAD_6_MOTOR_WHL)
- 6: B9:2/4 on and N7:12/7 on (B9:2/4 latched by LAD_9_RADIO_IN)

N7:12/7 is set in LAD_9_RADIO_IN and read only there and here; its meaning (inferred: a command-received/acknowledge state) is not described in the export. B3:4/2 is set in LAD_5_SEQ_MOV (inferred: move in progress). Code 3 is never produced.

Event-triggered sends. This routine latches B9:2/3 on three events:

- E-stop state change: a one-shot on B3:4/0 turning on (N7:21/4) and a separate one-shot on it turning off (N7:21/5). A message is therefore sent on each transition, and once at first scan.
- Stop command confirmation (see below).
- Periodic status: while B3:9/2 is on, B3:4/0 is on and N7:10/1 is off (inferred: Rover not in a movement sequence), timer T4:5 (1.0 s base, preset 50) runs. When it times out it self-resets and the one-shot N7:22/1 latches B9:2/3, so a status message is sent roughly every 50 s while stationary, plus one when the conditions first become true. B3:9/2 is not written by any routine in the program.

Network stop commands (rung 8). When no send is pending (B9:2/3 off) and N11:23 equals 152, 252 or 99, a one-shot (N7:21/3) performs, in one scan:

- latches N7:11/10 (all three codes; per comment, prevents the X and Y drives from being enabled — consumed in LAD_5_SEQ_MOV and LAD_3_POWER, cleared in LAD_3_POWER / LAD_13_FAULTS);
- code 99 only: latches B3:8/15 (per comment, disables the Safety Power Relay; consumed and cleared in LAD_3_POWER);
- code 252 only: latches B3:9/15 (per comment, puts the Rover in the faulted "Kill Command" state; consumed in LAD_13_FAULTS);
- clears N11:23 to 0;
- latches B9:2/3 to send the confirmation.

Message target addressing (rung 12). N10:11 = N10:2 − 1000 (rover index). If N10:11 ≤ 2 the target file number written to MG16:1.TFN and MG16:2.TFN is N10:11 + 90; otherwise it is N10:11. Only MG16:2 is used by a MSG instruction. The message writes the block starting at N11:30 to the target starting element N26:0 (file number overridden at runtime by TFN) via MultiHop SLC:1 -> ENET "10.162.21.190" -> CLB:0.

Interlocks and permissives

Safety-relevant — read first:

- **Network E-stop (command 99)** latches B3:8/15, which LAD_3_POWER uses to drop the Safety Power Relay (per rung comment). This is a software/communications-mediated stop, not a hardwired safety function. The rung comment states recovery requires the green button. The actual relay logic and recovery are in the LAD_3_POWER routine.
- **Kill command (252)** latches B3:9/15, evaluated as a fault in the LAD_13_FAULTS routine; recoverable by reset from the System PLC per comment.
- **Drive inhibit (152, 252, 99)** latches N7:11/10; consumed as a drive-enable inhibit in LAD_5_SEQ_MOV and LAD_3_POWER.
- **Stop-command processing is gated by B9:2/3 being off.** If a message is in flight or pending, a 152/252/99 command is not acted on until that message completes or errors (rung 14 releases B9:2/3). The stop is delayed by up to the MSG completion/timeout time.

Other permissives:

- MSG MG16:2 executes only when B9:2/3 is on and MG16:2/EN is off.
- Periodic status message requires B3:9/2 on, B3:4/0 on (safety circuit healthy, inferred) and N7:10/1 off.
- Distance calculation updates only while N7:10/12 is on.
- Transaction ID copy is blocked while N11:23 is 2, 3, 4 or 99.

Alarms and faults

No alarms are generated in this routine. It reports fault state (B30:0/0 from LAD_13_FAULTS) as movement status code 2 and reports E-stop state in N11:34. It originates the Kill Command fault latch B3:9/15 (evaluated in the LAD_13_FAULTS routine). A MSG error (MG16:2/ER) is not recorded or annunciated; it only releases the send request bits.

Review items

- VERIFY: Rung 12 comment says Rovers 1 and 2 are addressed to files 101 and 102, but the logic adds 90 (giving N91 and N92). Confirm the intended target file numbers in the System PLC.
- VERIFY: Rung 10 comment refers to N11:49 being "written in the second message", and rung 14 comment says "Write the first 19 values", but only one MSG (MG16:2) exists and its apparent length is 20 words (N11:30–N11:49). MG16:1.TFN is written every scan but MG16:1 is not used by any MSG instruction in any routine. Confirm whether a second message was removed and whether the single message length is 20. The MG16:1.TFN writes are likely dead code.

- VERIFY: MSG parameters (message type, local/target addressing, timeout, and the MultiHop path SLC:1 -> ENET 10.162.21.190 -> CLB:0) cannot be fully decoded from the export. Confirm the target controller type, that N-file writes via PCCC are mapped in the target, and the IP address. This is a hard-coded IP.
- VERIFY: Runtime modification of MG16:2.TFN is relied on to select the target file. Confirm this is supported/behaves as expected on the migration platform.
- VERIFY: No validation on N10:2. If N10:2 is less than 1000 (e.g. at first scan or if unset), N10:11 is negative and an invalid target file number results.
- VERIFY: Rung 1 comment ("if executing a motion related command") is the inverse of the logic (copies transaction ID when the command is NOT 2, 3, 4, or 99). Confirm intent and the meaning of command codes 2, 3, 4.
- VERIFY: Rung 8 clears N11:23 to 0 after acting on a stop command. N11:23 is also written by LAD_9_RADIO_IN. Confirm in the LAD_9_RADIO_IN routine that the received command is not rewritten every scan; otherwise the stop one-shot re-fires each cycle and a confirmation message is sent repeatedly.
- VERIFY: Distance calculation (rung 3) does not update N11:42 when N7:80 equals 0 or when N7:80 exactly equals F8:17 (distance is an exact whole number). N11:42 then holds a stale value and N11:46 is computed from it. Also confirm the float-to-integer MOV rounding behavior on the target platform, since the truncation logic assumes round-to-nearest.
- VERIFY: CAM position (rung 4): if B3:5/9 and B3:5/10 are both on, N11:38 = 2 (last branch wins). Confirm these are mutually exclusive in the LAD_7_MOTOR_CAM routine.
- VERIFY: Lift band thresholds N7:34 and N7:36 are static data values not written by any routine; values are not in the export. The LIM instruction assumes N7:34 ≤ N7:36.
- VERIFY: Movement status precedence (rung 7): codes 4, 5, 6 override fault code 2 when a request bit and N7:12/7 are both on. Confirm this is intended. Code 3 is never produced. Confirm the code definitions against the System PLC protocol.
- VERIFY: Rung 14 unlatches B9:2/1, B9:2/2, B9:2/3 and B9:2/4 on completion OR error. A message that errors is not retried, and any request latched while a message was in flight is cleared without its data being sent (N11:37 codes 4/5/6 can be dropped).
- VERIFY: B3:9/2 (periodic message enable) is not written anywhere in the program. Confirm whether it is set by HMI/programming software or is a fixed data-table value.
- VERIFY: N7:12/7, B3:7/15, B3:3/3, N7:10/1, N7:10/12, B3:4/2 have no descriptions; purposes above are inferred from name/usage and cross-routine digests.
- VERIFY: N11:33/0 and N11:35/0 are driven with OTE on single bits of message words; the remaining bits of N11:33 and N11:35 are not written by this routine.
- Safety review: Network E-stop (99) and Kill (252) are communications-mediated stops with no hardwired backing in this routine, and their processing is delayed while a message is pending. Confirm this is acceptable in the plant's safety assessment for the migrated platform.

Routine: LAD_11_BARCODE

Purpose

LAD_11_BARCODE runs a four-message CIP handshake with an Ethernet barcode reader at 192.168.100.3 (trigger, poll status, fetch result, reset), decodes the returned string into a 4-digit integer in N11:39, and — when the vehicle is inside a position window — latches a "new barcode" flag for the radio transmit routine and records the value in N7:30.

Functional description

Message set (rungs 1–5). Four MSG instructions (MG20:1..MG20:4) target the same device via MultiHop SLC:1 -> "192.168.100.3", all against CIP object class 0x69, instance 1, attribute 1. Their service codes differ:

MSG	Service	Data	Inferred role
MG20:1	0x4B	source N10:10, 2 words	"Trigger / start read" command

MSG	Service	Data	Inferred role
MG20:2	0x0E (Get Attribute Single)	destination N10:30, 8 words	Status poll
MG20:3	0x55	source N10:20 (=32), 4 words; destination N10:40, 40 words	Fetch result string
MG20:4	0x4C	none	Reset / acknowledge

The roles are inferred from their position in the state sequence; the actual device semantics are not in the export.

State sequence. Three latch bits in B3:24 drive the sequence (names inferred from usage):

- B3:24/12 — trigger sent, waiting for status.
- B3:24/13 — result ready, fetch requested.
- B3:24/11 — reset requested.

Flow:

1. With permissive B3:4/0 true and B3:24/12 clear, MG20:1 (trigger) is sent. On MG20:1/DN, B3:24/12 latches.
2. With B3:24/12 set and B3:24/13 clear, MG20:2 (status poll) is sent repeatedly (re-issued each time the previous poll completes). On MG20:2/DN:
   - If N10:31/0 is set (inferred "result available"), B3:24/13 latches.
   - If N10:31/0 is clear and either N10:30/8 is clear or N10:30/0 is set (device status bits, meaning not in export), B3:24/11 latches — a reset is requested.
3. With B3:24/13 set, MG20:3 (fetch result) is sent into N10:40..N10:79.
4. MG20:4 (reset) is sent on a one-shot (N7:21/13) when either B3:24/11 is set, or the sequence status combination N7:10/1 true AND N7:10/12 false becomes true (both bits come from LAD_5_SEQ_MOV; N7:10/12 is also written by LAD_3_POWER and LAD_6_MOTOR_WHL). The same rung unlatches B3:24/11. When MG20:4/EN goes true, B3:24/12 is unlatched so the trigger is re-sent.
5. After the result fetch completes (B3:24/13 AND MG20:3/DN), or whenever B3:4/0 is false, B3:24/13, B3:24/12 and N10:31/0 are all unlatched, restarting the cycle from the trigger.

N10:20 is written to 32 every scan (rung 0). It is the request data for MG20:3, so it is most likely a maximum result length parameter hard-coded to 32.

Decode (rung 7). When the result fetch has completed, B9:2/2 is clear (previous barcode already consumed — see LAD_10_RADIO_OUT), and N10:42 ≠ 21061, the routine decodes four ASCII characters from N10:42 (low byte, high byte) and N10:43 (low byte, high byte):

- N10:60..N10:65 are pre-loaded from N11:50..N11:55 by COP (needed because MVM writes only the masked bits).
- The low nibble of each byte is extracted by MVM (masks 000Fh / 0F00h) and, for high bytes, divided by 256. For ASCII digits '0'..'9' (0x30..0x39) the low nibble is the digit value.
- N11:39 = ((d1×10 + d2)×100) + (d3×10 + d4), i.e. the 4-digit code.

21061 = 0x5245, which is the ASCII pair "E","R" in little-endian word order — inferred to be the start of an "ERROR"/"ER" no-read string. When the result starts with that pair, N11:39 is not updated. N11:39 is consumed by LAD_3_POWER, LAD_6_MOTOR_WHL, LAD_7_MOTOR_CAM and LAD_13_FAULTS.

Clearing the value (rung 6). N11:39 is set to 0 when a status poll completes with no result available (B3:24/12 set, B3:24/13 clear, N10:31/0 clear, MG20:2/DN) and B9:2/2 is clear, or whenever B3:4/0 is false.

New-barcode report (rung 9). When B3:13/0 is true, position L12:0 (from LAD_6_MOTOR_WHL) is within the window L12:13..L12:16 (both from LAD_5_SEQ_MOV), and N11:39 is non-zero and differs from the last accepted value N7:30, the routine latches B9:2/3 (general transmit/update trigger, written by several routines and consumed in LAD_10_RADIO_OUT) and B9:2/2 (new-barcode pending; cleared in LAD_10_RADIO_OUT), and copies N11:39 to N7:30.

Interlocks and permissives

- **B3:4/0** (written in LAD_4_IO_MAP; description not in export, inferred as a system/run enable) is the master permissive for the barcode cycle. When false: no trigger is sent, all sequence latches and N10:31/0 are cleared, and N11:39 is forced to 0.
- **B9:2/2** (new-barcode pending) blocks both the decode (rung 7) and the value-clear (rung 6). A read completing while B9:2/2 is still set is discarded without decode, because rung 8 resets the cycle regardless.
- **B3:13/0** must be true for a barcode to be reported (rung 9). No routine in this program writes B3:13/0 (see Review items).
- **Position window** L12:13 ≤ L12:0 ≤ L12:16 gates the barcode report. (If L12:13 > L12:16 the LIM instruction inverts and passes values outside the window.)
- N7:10/1 AND NOT N7:10/12 (sequence status from LAD_5_SEQ_MOV) forces a reader reset command on its rising edge.
- No barcode-related interlock acts on any physical output in this routine.

Alarms and faults

None identified. The routine does not evaluate any MSG /ER bit and has no communications timeout. An errored message is simply re-issued when its EN bit clears. Downstream use of N11:39 in LAD_13_FAULTS is documented in that routine.

Review items

- VERIFY: Device at 192.168.100.3 — make, model and the meaning of CIP class 0x69 services 0x4B / 0x0E / 0x55 / 0x4C, instance 1, attribute 1. The roles (trigger / status / fetch / reset) are inferred from sequence position only.
- VERIFY: Meaning of status bits N10:31/0 (assumed "result available"), N10:30/8 and N10:30/0 (their combination requests a reset). Confirm with device documentation.
- VERIFY: Contents of N10:10..N10:11 (request data for MG20:1). No routine writes these words; they are data-table constants.
- VERIFY: Contents of N11:50..N11:55. No routine writes them; they pre-load the digit registers. If they are not zero in the data table the decoded barcode is corrupted. Expect all zeros.
- VERIFY: N10:20 = 32 is hard-coded every scan as MG20:3 request data. Confirm it is the maximum result length and whether the migrated system needs it adjustable.
- VERIFY: The constant 21061 (0x5245 = "ER") in rung 7 is assumed to detect a no-read/error string. Confirm the reader's actual no-read response.
- VERIFY: Result string layout — decoding assumes the four barcode characters are bytes 4–7 of the result buffer (N10:42–N10:43) and that N10:40–N10:41 are a header. Confirm.
- VERIFY: MSG control-bit fields differ for MG20:4 ("0 0 0 0 0 1 0 0 0 0") versus the other three ("…0 1 1 0"). Check the MG20:4 configuration (timeout/connection options) in the original project.
- **Suspicious — no writer for B3:13/0.** Rung 9 (the only place a barcode is reported to the radio) requires B3:13/0, which is not written by any routine in the program. If it is not set by an HMI, radio message, or data-table default, the entire barcode function is effectively dead. VERIFY: source and default value of B3:13/0.
- **Suspicious — no error/timeout handling.** No MSG /ER bit is used and there is no watchdog on the read cycle. If the reader never returns a "result available" or a reset condition, MG20:2 polls indefinitely with N11:39 forced to 0, and no fault is raised. The only recovery is the N7:10/1 / N7:10/12 sequence transition or loss of B3:4/0.
- Decode has no digit validation. Any non-numeric ASCII character contributes its low nibble (e.g. 'A' -> 1) with no error indication.
- A completed read is silently discarded if B9:2/2 is still latched from the previous barcode (rung 8 resets the cycle even when rung 7 does not decode). Confirm this is acceptable for the expected read-to-transmit timing.
- Rung 4 triggers MG20:4 through a one-shot while MG20:1–3 may be in flight; there is no mutual exclusion between the four messages. Confirm the device tolerates overlapping requests.

Routine: LAD_12_MBLVIEW

Purpose

LAD_12_MBLVIEW selects the operator-interface control mode from a hardwired input and a mode number, and, in one of those modes, converts four HMI pushbutton bits into command frames that are copied into the radio command buffer (N11:0–15) so the rest of the program treats them as received radio commands. The routine name and the data layout indicate a MobileView (tethered pendant) interface; this is inferred, not stated in the export.

Functional description

Mode selection (rung 0). While discrete input I:0.0/9 is on, the value of N7:3 selects one of two mode bits:

- N7:3 = 1 sets B3:10/2.
- N7:3 = 2 sets B3:10/0.

Both are direct outputs (OTE), so they drop immediately when I:0.0/9 is off or N7:3 holds any other value. The two bits are mutually exclusive. B3:10/2 is not used further in this routine; it is consumed by LAD_2_MAIN, LAD_6_MOTOR_WHL, LAD_7_MOTOR_CAM, LAD_8_MOTOR_LIFT and LAD_13_FAULTS (see those sections). B3:10/0 enables everything else in this routine and is also read by LAD_7_MOTOR_CAM. No routine in the program writes N7:3; it is inferred to be a configuration value or an HMI-written selector.

Command sequence counter (rung 1). In mode 2 (B3:10/0 on), the rising edge of any one of the four button bits B3:8/8, /9, /10 or /11 (single one-shot N7:20/3, shared by all four) increments a sequence counter N17:80, provided N7:10/1 is off. The counter rolls over: when the incremented value reaches 9999 it is reset to 1, so the counter actually cycles 1–9998. The new value is then written into word 2 of each of the four command frame templates (N17:2, N17:22, N17:42, N17:62). Because the four templates are re-stamped every time, each frame carries the latest sequence number when it is sent.

Command frame injection (rungs 2–5). In mode 2, with N7:10/1 off, the rising edge of each individual button bit (one-shots N7:22/8–/11) copies a 16-word frame template into N11:0–15:

- B3:8/8 -> COP N17:0–15 -> N11:0–15
- B3:8/9 -> COP N17:20–35 -> N11:0–15
- B3:8/10 -> COP N17:40–55 -> N11:0–15
- B3:8/11 -> COP N17:60–75 -> N11:0–15

N11:0–9 is the buffer decoded by LAD_9_RADIO_IN (see that section). This routine therefore lets HMI pushbuttons issue the same commands the radio receiver would deliver. The template content (N17:0–79, other than word 2 of each block) is not written by any routine in the program; it is inferred to be pre-loaded by the HMI or as static data. Rung 1 executes before rungs 2–5 in the same scan, so a frame copied on a first button press carries the freshly incremented counter.

Rung 6 is the subroutine return.

Interlocks and permissives

- **I:0.0/9 (hardwired input) is the master permissive for both mode bits.** With it off, B3:10/0 and B3:10/2 are off and no HMI command frames are generated. Its function is not described in the export; given the MobileView context it may be an interface-connected, key-switch or enabling-device contact. See VERIFY below.
- **N7:3 must equal 2** for HMI command injection to be active (B3:10/0). N7:3 = 1 enables B3:10/2 only, which gates behavior in the motor and fault routines rather than here.
- **N7:10/1 must be off** for the counter increment and for every frame copy. N7:10/1 is produced by LAD_3_POWER and LAD_5_SEQ_MOV (see those sections) and is used as a blocking condition throughout the program; its meaning is not described in the export and is inferred to be an inhibit/fault-type condition.
- **Safety relevance:** this routine is a path by which pendant pushbuttons inject motion-related commands into the radio command buffer. Whatever enabling/deadman protection applies to those commands must come from I:0.0/9, N7:10/1, or downstream logic (LAD_9_RADIO_IN and the motor routines); none is implemented within this routine itself.

Alarms and faults

None identified. The routine generates no fault bits, timers or alarm words.

Review items

- VERIFY: Function and wiring of I:0.0/9. It is the sole hardwired permissive for both interface mode bits. Confirm whether it is a MobileView enabling (deadman) switch, a connected/present signal, or a selector switch, and whether it is part of the safety chain.
- VERIFY: Source and expected values of N7:3. No routine writes it. If it is a fixed value in the data table, one of the two mode branches in rung 0 is permanently dead. If the HMI writes it, confirm which screens/conditions set 1 vs. 2.
- VERIFY: Source of B3:8/8–/11. No routine writes these bits; they are inferred to be HMI pushbutton bits. Confirm the HMI tag mapping and whether they are momentary (required for the one-shot logic to behave as intended).
- VERIFY: Content of the four frame templates N17:0–15, N17:20–35, N17:40–55, N17:60–75. Only word 2 of each is written by the PLC. Confirm who loads the remaining words and what command each frame represents (e.g., stop, jog, velocity clear), since these frames are decoded as radio commands by LAD_9_RADIO_IN.
- VERIFY: N11:0–15 is shared with the radio receive path. Confirm how radio data enters N11:0–15 (channel/communication configuration is not in the export) and whether a radio message can overwrite, or be overwritten by, an HMI-injected frame in the same scan.
- Rung comment mismatch: the comment "Clear last velocity command if deadman is released" precedes rung 3, but rung 3 does not clear anything — it copies the N17:20 template on the rising edge of B3:8/9. Either the comment is stale, or the N17:20 frame is itself a "clear velocity" command sent by a button release path that no longer exists. VERIFY which is the case.
- Inconsistent one-shot placement: in rung 1 the one-shot is before XIO N7:10/1, so a button pressed while N7:10/1 is on does not increment the counter and will not increment later when N7:10/1 clears. In rungs 2–5 the one-shot is after XIO N7:10/1, so the frame copy fires when N7:10/1 clears while the button is still held. A frame can therefore be sent without a new sequence number.
- Shared one-shot in rung 1: a single one-shot covers the OR of all four buttons. If a second button is pressed while another is held, rungs 2–5 send the second frame but the sequence counter is not incremented, so two different frames carry the same sequence number. Confirm whether the receiver depends on unique sequence numbers.
- Counter range: the reset-on-9999 executes in the same scan as the increment, so the counter cycles 1–9998 and never transmits 9999. Minor, but note for any receiver-side range checks.
- Template blocks are 20 words apart but only 16 words are copied; words 16–19 of each block are unused by this routine. Likely intentional padding; confirm during migration so no data is lost.

Routine: LAD_13_FAULTS

Purpose

LAD_13_FAULTS detects rover operating faults (charging, motion, encoder, load, lift, cam and index timeouts), latches a single master fault bit with a numeric fault code, triggers the radio fault report, and performs a post-move encoder distance verification. It is called unconditionally from LAD_2_MAIN rung 11.

Functional description

No tag descriptions exist in this export. Every tag meaning below is inferred from name, usage and the routine digests; treat them as inferred unless stated otherwise.

Master fault structure (common to all detectors)

Every detector follows the same pattern: a condition holds for a timer period, then the routine latches the master fault bit `B30:0/0`, writes a numeric code to `N11:41`, and latches one fault-specific bit in the `B30:30`–`B30:37` range. The fault-specific bit position equals (code − 497) counted from `B30:30/0`, consistently for every rung, so the bit is derived from the code actually written (see review item on code 509/510).

Every detector is gated by `XIO B30:0/0`, so the first fault to trip blocks all others: `N11:41` holds one code at a time and the fault timers stop running once any fault is active.

No rung in this routine clears `B30:0/0`, `N11:41` or the `B30:3x` fault bits. Per the digests, the only other writer of `B30:0` is LAD_9_RADIO_IN, which writes the `B30:0` file as a block (`#B30:0`). See the LAD_9_RADIO_IN section for the reset path.

Fault reporting (rungs 18, 19)

While `B30:0/0` is set, the routine latches `N7:11/10` (a "faulted" status bit consumed by LAD_3_POWER and LAD_5_SEQ_MOV; also written by LAD_10_RADIO_OUT). It also runs a 5 s retransmit cycle: on each cycle a one-shot latches `B9:2/3` and increments counter `C5:6`. `B9:2/3` is written by most routines and is inferred to be the radio "send message" request handled in LAD_10_RADIO_OUT. `C5:6` has preset 3, so the fault report is sent at most three times (initially, then twice at 5 s intervals). `B3:3/1` (written by LAD_9_RADIO_IN; inferred to be a remote reset/acknowledge) resets `C5:6`.

Fault detectors

Code written	Fault bit	Condition (inferred meanings)	Delay
502	B30:30/5	Rover is on charging pad (`I:0.0/7` or barcode `N11:39` within `N7:70`..`N7:71`) and state of charge `N7:0` is below the baseline captured in `N7:54`	240 s
505	B30:30/8	In motion (`N7:10/12`), `B3:3/2` off, measured travel `F8:17` exceeds `L12:26` = commanded distance `N11:24` + allowance `N7:77`	none
507	B30:30/10	In motion, lift not moving (`B3:6/8`, `B3:6/9` off), a wheel drive selected (`N7:10/6` or `/7`), encoder count `L12:0` has not exceeded moving threshold `L12:1`	10 s
509	B30:30/12	`B3:12/15` and `B3:8/1` on, in motion, lift raised (`B3:5/14` or `/15`), load sensor (`N14:70/2` or `/9`) on, and an overhang/off-centre sensor (`N14:70/3`, `/4`, `/10`, `/11`) on	5 s
510	B30:30/13	`B3:12/15` on, sequence active (`N7:10/1`), not in motion, `B3:5/13` off, carrying state (`N7:12/3` or `/4`), both load sensors `N14:70/2` and `/9` off	2.5 s
512	B30:30/15	Lift command `N7:10/13` on without lift-complete `N7:10/15`	16 s
514	B30:31/1	Sequence active `N7:10/1` without motion-complete `N7:12/0`	180 s
516	B30:31/3	In motion with cams not confirmed for the selected drive: X drive `N7:10/7` without `B3:5/9`, `B3:7/5`, `B3:7/6`; or Y drive `N7:10/6` without `B3:5/10`, `B3:7/7`, `B3:7/8`	1 s
530	B30:32/1	`B3:4/0` on, sequence active, `B3:5/13` off, a direction bit (`N7:10/2../5`) is set together with its paired input (`B3:12/0`, `/2`, `/3`, `/6`), no index/lift/motion in progress (`N7:10/11`, `/13`, `/12` off), `N7:10/9` on	4 s
540	B30:32/11	`B3:9/15` on and radio word `N11:23` equals 252	1 s
606	B30:36/13	Index command active (`N7:10/11` or `/14`), `N7:11/1` off, index-complete `N7:12/9` off, `B3:3/4` off	5 s
609	B30:37/0	`B3:8/1` on, `N11:26` = 0, either (lift lowered and idle with `N7:11/5`) or (in motion, lift raised, cams in Y), and on each side a load sensor is off or an overhang sensor is on	T4:3, 0.1 s or 0.75 s (see below)
615	B30:37/7, B3:7/1	`B3:4/0` on and cam motor running: in auto (`B3:10/2` off) via `B3:4/15` or `B3:6/10`; in manual (`B3:10/2` on) via `B3:11/4`, `/5`, `/6`, `/10`, `/11`	12 s cumulative (RTO T4:11)

Detail on selected detectors:

- **502 – charging baseline.** When the rover is on the pad and `N7:0` ≤ 80, a one-shot captures `N7:0` into `N7:54`. The fault timer runs only while `N7:0` is strictly less than that baseline.

- **505 – overtravel counters.** When 505 trips, the routine increments one of four diagnostic counters `N7:81`..`N7:84`, chosen by the active direction bit (`N7:10/2`, `/4`, `/3`, `/5`) when its associated sensor (`N14:70/6`, `/13`, `/0`, `/7`) is not seen. These counters are read only within this routine.
- **507 – encoder feedback.** While the encoder count is above `L12:1`, the rung re-arms by setting `L12:1` = `L12:0` + 20 counts and the timer resets. If the count does not advance past `L12:1` within 10 s, 507 trips. `L12:1` is also written by LAD_6_MOTOR_WHL.
- **606 – index timeout counters.** When 606 trips, five diagnostic counters record the sensor pattern at the time of the fault: `C5:0` if barcode `N11:39` is 0; `C5:1`..`C5:4` by combinations of `N14:70/0`, `/1`, `/5`, `/6`, `/7`, `/8`, `/12`, `/13`. Their presets are 0 and their DN bits are not read.
- **609 – dynamic timer preset (rung 14).** While in motion, `T4:3.PRE` is set to 750 ms, or 100 ms while `L12:0` ≤ `L12:13` (a position threshold from LAD_5_SEQ_MOV). The literal preset of 750 in rung 15 is overridden.
- **615 – RTO.** `T4:11` is retentive, so cam run time accumulates across separate run periods until reset elsewhere.

Cam reposition retry (rung 16) – not a fault

If the rover is stopped, `N7:11/1` is set, and the cams are not confirmed for the selected drive (same conditions as fault 516) for 5 s, the routine increments diagnostic counter `C5:7` and, via one-shot `B3:12/14`, unlatches `N7:11/1`. `N7:11/1` is also written by LAD_7_MOTOR_CAM; this is inferred to force a cam re-index rather than raise a fault.

Move-progress flag (rung 2)

While in motion, when encoder count `L12:0` reaches `L12:9` (a distance target written by LAD_5_SEQ_MOV), a one-shot latches `N7:12/2`. No routine reads `N7:12/2` as a bit (see review items).

Encoder distance verification (rungs 20, 21)

While a sequence is active (`N7:10/1`), `F8:49` tracks encoder count `L12:0` as a float; a one-shot zeroes it on the first scan of the sequence. When motion stops (one-shot on `N7:10/12` falling) with no fault active and commanded distance `N11:24` greater than minimum `N7:9`, the routine computes `F8:50` = |`N11:24` − `F8:17`| (commanded minus measured distance). If `F8:50` exceeds tolerance `F8:52`, it computes a counts-per-distance factor `F8:49` / `N11:24` into `F8:47` (X drive: `B3:5/9`, `N14:70/6`, `/13`) or `F8:48` (Y drive: `B3:5/10`, `N14:70/0`, `/7`). In RSLogix 500 CPT syntax `|` is division. `F8:47` and `F8:48` are not read by any routine.

Interlocks and permissives

This routine drives no physical outputs. Its safety-relevant effect is through the master fault bit:

- `B30:0/0` (master fault, latched here) is read by LAD_5_SEQ_MOV, LAD_6_MOTOR_WHL and LAD_10_RADIO_OUT. See those sections for how motion and the sequence are inhibited when it is set.
- `N7:11/10` (latched here on fault) is read by LAD_3_POWER and LAD_5_SEQ_MOV.
- `B3:7/1` (latched on fault 615) is read by LAD_7_MOTOR_CAM.
- LAD_7_MOTOR_CAM and LAD_8_MOTOR_LIFT do not read `B30:0/0`. Any inhibit of the cam and lift motors on a master fault must come through the sequence bits they read (`N7:10/1` etc.), not directly from this routine.
- `XIO B30:0/0` in every detector is an internal permissive: once one fault is latched, no other fault can be detected or coded.

VERIFY: Confirm with the plant engineer that cam and lift motors are stopped on a master fault through LAD_5_SEQ_MOV, since neither LAD_7_MOTOR_CAM nor LAD_8_MOTOR_LIFT reads `B30:0/0` directly.

Alarms and faults

Thirteen latched fault conditions, as tabled above: 502, 505, 507, 509, 510, 512, 514, 516, 530, 540, 606, 609, 615. All set `B30:0/0`, write the code to `N11:41`, latch a bit in `B30:30`–`B30:37`, and trigger up to three radio fault reports via `B9:2/3`. Fault 615 additionally latches `B3:7/1`. Reset is not performed in this routine; see LAD_9_RADIO_IN (`#B30:0` file write) for the clear path.

Review items

VERIFY: Rung comments do not match the logic in several places and appear shifted or copied from another project: rung 6 is commented "510" but writes code 509; rung 11 is commented "550 – Y Drive Faults" but writes 530 and does not test any drive fault input; rung 12 is commented "550" but writes 540; rung 13 is commented "609 – Timed Out Looking for Pallet" but writes 606 and tests index bits; rung 15 is commented "Clear comm fault if message was able to recover" but is the 609 pallet-not-centred detector; rung 16 is commented "552 – X Drive Faults" but raises no fault; rung 17 is commented "553 – X Drive Comm Faults" but writes 615 for cam-motor run time; rung 21 is commented "606 – Unable to index wheels" but is the encoder verification. Several comments refer to a "bridge", which is crane terminology. Confirm the intended fault code list with the plant engineer before the migration reproduces the codes as written.

VERIFY: Rung 6 writes code 509 and latches `B30:30/12`, the bit position for 509. Comment says 510. Confirm whether 509 is the intended code for "load off-centre while in motion" and 510 for "no load with lift raised" (rung 7), or whether rung 6 should be 510 and rung 7 something else.

VERIFY: Fault 502 comment states 5 minutes; timer `T4:24` preset is 240 × 1.0 s = 4 minutes. Also, the timer runs only while charge `N7:0` is strictly below the captured baseline `N7:54`, not when it merely fails to increase. Confirm intended behavior.

VERIFY: Fault 507 assumes the encoder count `L12:0` increases during motion. If the count decreases for a direction, `L12:0` ≤ `L12:1` holds continuously and 507 trips after 10 s. Confirm `L12:0` is an absolute/unsigned count (see LAD_6_MOTOR_WHL, which writes `HSC:0.ACC`/`HSC:3.ACC` and `L12:0`).

VERIFY: Timer `T4:11` (fault 615 RTO) is also written by LAD_5_SEQ_MOV and LAD_7_MOTOR_CAM; timer `T4:33` (fault 606) is also written by LAD_7_MOTOR_CAM. Multiple routines instructing the same timer produces scan-order dependent behavior. Confirm the other writes are RES instructions and not additional TON/RTO uses.

VERIFY: Rung 13 counter `C5:2` branch tests `N14:70/7 OR N14:70/7` (duplicated bit). By symmetry with `C5:1` (`/7 OR /8`) this is likely a typo for `/7 OR /8`. Diagnostic only; confirm.

VERIFY: `B3:8/1` (enables faults 509 and 609), `N7:77` (overtravel allowance), `N7:9` (minimum verify distance), `F8:52` (distance tolerance), `N7:70`/`N7:71` (charging barcode range) and `B3:11/10`, `B3:11/11` are not written by any routine. They are data-table constants or HMI/radio-written values. Their configured values are not in this export and must be captured before migration.

VERIFY: `I:0.0/7` is inferred as a "charger connected / on pad" input; also read by LAD_3_POWER. Confirm wiring.

VERIFY: `B30:0/0` and the `B30:3x` fault bits are latched here and cleared only by the `#B30:0` file write in LAD_9_RADIO_IN. Confirm that this clears every fault bit and `N11:41`, and that no fault can be cleared while its cause persists (the detectors re-trip only after their timers re-expire).

Suspicious / likely dead logic:

- `N7:12/2` (rung 2) is latched but never read as a bit by any routine; LAD_5_SEQ_MOV writes the whole word `N7:12`, which may overwrite it. Likely dead or HMI-only.
- `F8:47`, `F8:48` (rung 21) are computed but never read. `F8:50` is read only within this routine. Diagnostic only.
- Counters `N7:81`–`N7:84`, `C5:0`–`C5:4`, `C5:7` are occurrence counters read nowhere else; `C5:0`–`C5:4`, `C5:7` have preset 0. Diagnostic only, likely viewed via HMI.
- Rung 17 branch `B3:4/15 OR (NOT B3:4/15 AND B3:6/10)` reduces to `B3:4/15 OR B3:6/10`; the `XIO B3:4/15` is redundant.
- Rung 4 (fault 505) is the only detector with no time delay; a single scan over `L12:26` trips it.
- Rung 15 `TON T4:3` literal preset 750 is overridden by rung 14 while in motion; while stopped, `T4:3.PRE` retains its last in-motion value.

Routine: LAD_14_VELOCITY

Purpose

LAD_14_VELOCITY computes wheel-motor velocity (from encoder position and high-speed counter data) and captures phase durations of the movement sequence (acceleration, deceleration, creep, a pre-move step, and lift-motor run time) into five rolling FIFO logs. It drives no field outputs; it produces analysis/display data, of which only one value (F39:28) is consumed by another routine.

Functional description

Tag naming note. No tag in this routine has a description in the export. All roles below are inferred from usage and cross-routine references and are marked as such.

1. Encoder-based speed (counts/s and scaled), rungs 0–1

- Rung 0 selects a high-speed counter accumulator into F39:25: HSC:0.ACC when B3:5/9 is on, HSC:3.ACC when B3:5/10 is on. B3:5/9 and B3:5/10 are written by the LAD_7_MOTOR_CAM routine and are inferred to be direction or axis-select bits (each associated with one of two HSC channels).
- Rung 1 runs a 0.5 s sample timer (T4:29, 1 ms base, preset 500). Each time it completes:
  - F39:27 = change in counts since the last sample (F39:25 − F39:26); F39:26 is then updated with the current count.
  - If the change is within ±2000 counts and the timer accumulator is > 0, F39:28 = |Δcounts / ACC(ms) × 1000|, i.e. counts per second. The ±2000 window is inferred to reject rollover or glitch samples.
  - If F8:0 > 0, F39:29 = F39:28 / F8:0. F8:0 is written by the LAD_5_SEQ_MOV routine and, from its use as a divisor of position counts throughout this routine, is inferred to be a counts-per-engineering-unit scale factor.
  - T4:29 is reset. A second TON T4:29 instruction in the last branch restarts the timer in the same scan.
- F39:28 is read by the LAD_6_MOTOR_WHL routine. F39:29 is not read by any routine.

2. Scaled position, rung 3

- When L12:0 > 0, F39:20 = L12:0 / F8:0. L12:0 is written by the LAD_6_MOTOR_WHL routine and is inferred to be the wheel position in encoder counts; F39:20 is therefore inferred to be position in engineering units. F39:20 is not read by any routine.

3. Time-stamped velocity, two channels, rungs 2, 4–8

- Rung 2 runs two free-running interval timers: T4:40 (10 ms base, preset 500 = 5 s) and T4:41 (10 ms base, preset 416 = 4.16 s). Each is reset only by its velocity rung below.
- Rung 4 (channel A) runs when B3:12/15 is on, AND (N7:10/6 OR B3:5/10) is on, AND T4:40 is done. Rung 5 (channel B) runs when (N7:10/7 OR B3:5/9) is on AND T4:41 is done. N7:10/6 and N7:10/7 are written by the LAD_5_SEQ_MOV routine and are inferred to be sequence-step bits; B3:12/15 is written by the LAD_2_MAIN routine (role inferred as a global run/enable bit).
- Each channel performs the same calculation using its own working registers (A: F39:0–F39:6, F40:0–F40:1; B: F39:10–F39:16, F40:5–F40:6):

1. Read the controller high-speed clock (RHC) and compute the elapsed time since the previous sample (TDF), then divide by 100,000 to obtain seconds.
2. Convert position: L12:0 / F8:0, then × 25.4. This implies the scaled position is in inches and is converted to millimetres.
3. Velocity = Δposition (mm) / Δtime (s), written to F39:0 (channel A) or F39:10 (channel B). If Δtime ≤ 0 the velocity is forced to 0.0.
4. Store current position and clock value for the next sample, reset the interval timer, and energise a one-scan marker bit (B3:20/0 or B3:20/1).

- Rungs 6–7 take the absolute value: F39:5 = |F39:0|, F39:15 = |F39:10|.
- Rung 8 converts to display units: F39:8 = F39:5 / F39:7 and F39:18 = F39:15 / F39:17. The rung comment identifies this as a conversion to ft/min; F39:7 and F39:17 are constants held in the data table and are not written

by any routine.

- None of F39:0, F39:5, F39:8, F39:10, F39:15, F39:18, B3:20/0 or B3:20/1 is read by any routine.

4. Sequence-phase duration logging (five FIFOs), rungs 9–24 Five RTO timers (1 ms base) accumulate time while a phase condition is true. At the end of the phase the accumulated value is pushed into an N41 FIFO (FFL) and the timer is reset. Each FIFO wraps to position 0 when full (rungs 10, 13, 16, 19, 22). The FFL executes on the false-to-true transition of its rung; the RES in the same branch then holds the timer at zero until the phase condition next permits timing.

FIFO	Timer / preset	Times while…	Pushed when…	Inferred phase
#N41:0 / R6:0 (len 8)	T4:42 / 9 s	B9:15/0 OR B9:15/1 on (both written by LAD_6_MOTOR_WHL)	N7:10/12 turns off (rung 12)	Acceleration
#N41:10 / R6:1 (len 10)	T4:43 / 15 s	(B3:12/10 OR B3:12/11) on AND B9:15/4 off (B3:12/10–11 written by LAD_6_MOTOR_WHL)	N7:10/12 turns off (rung 15)	Deceleration
#N41:20 / R6:2 (len 10)	T4:44 / 9 s	N7:11/2 on AND N7:11/3, /4, /7 off (N7:11 bits written by LAD_6_MOTOR_WHL)	N7:10/12 turns off (rung 18)	Creep
#N41:30 / R6:3 (len 10)	T4:45 / 9 s	N7:10/11 on AND N7:10/12 off	N7:10/12 turns **on** (rung 21)	Pre-move step ("Time Index" per comment)
#N41:40 / R6:4 (len 10)	T4:46 / 9 s	N7:10/12 off AND O:0.0/10 on (lift output, written by LAD_8_MOTOR_LIFT)	O:0.0/10 turns off (rung 24)	Lifting lid

- N7:10/12 is written by the LAD_5_SEQ_MOV and LAD_6_MOTOR_WHL routines. From its use as the "end of movement" trigger it is inferred to be a "movement in progress" step bit.
- Because RTO timing stops at the preset, captured durations saturate at 9.000 s (15.000 s for deceleration).
- Rung 9: when B3:24/14 is on, all five FIFO positions are reset to 0 and B3:24/14 is unlatched. No routine sets B3:24/14; it is inferred to be an HMI "clear logs" command.
- No N41 word is read by any routine.

Interlocks and permissives

None identified. This routine writes no field outputs and no bits used as permissives elsewhere. The following conditions only gate internal calculations:

- Channel A velocity (rung 4) requires B3:12/15 (from LAD_2_MAIN) plus a sequence-step or direction bit; channel B (rung 5) has no B3:12/15 gate.
- Encoder speed sample (rung 1) is only scaled when F8:0 > 0 and only computed when |Δcounts| ≤ 2000.

Alarms and faults

None identified. Note the divide-by-zero exposure listed under Review items; a zero divisor sets the processor math overflow flag S:5/0, which the LAD_2_MAIN routine writes (inferred to be a clear of that flag).

Review items

- VERIFY: Rung comments in this export are offset by two rungs from the logic they describe (e.g. "Generate Absolute Value of Velocity" precedes rung 4 but describes rungs 6–7; "Lifting Lid Timer" precedes rung 20 but describes rungs 22–24). The phase names in the table above assume this two-rung offset. Confirm phase assignments with the plant engineer.
- VERIFY: Roles of B3:5/9, B3:5/10, N7:10/6, N7:10/7, N7:10/11, N7:10/12, N7:11/2–/7, B9:15/0–/1, B3:12/10–/11, B3:12/15 are inferred only; no descriptions exist in the export.
- VERIFY: F8:0 (scale factor), F39:7 and F39:17 (ft/min conversion constants) are data-table values not written by any routine. Confirm their values. F39:7 and F39:17 are used as divisors with no zero check (rung 8); F8:0 is used as a divisor without a zero check in rungs 3, 4 and 5 (rung 1 does check it). A zero value produces a math overflow.

- VERIFY: RHC/TDF clock units. The logic divides the TDF result by 100,000 to obtain seconds, implying a 10 µs tick. Confirm against the target platform when migrating; the equivalent instruction may differ.
- VERIFY: Consumers of computed values. Only F39:28 is read by another routine (LAD_6_MOTOR_WHL). F39:0, F39:5, F39:8, F39:10, F39:15, F39:18, F39:20, F39:29, B3:20/0, B3:20/1 and all N41 FIFO arrays are not read by any routine. Confirm which are displayed or trended by the HMI; otherwise they are candidates for removal.
- VERIFY: B3:24/14 (FIFO clear) is never set by any routine. Confirm it is an HMI pushbutton; if not, rung 9 is dead code.
- VERIFY: B9:15/4 (rung 14) is not written by any routine. If it has no HMI source the XIO is always true and can be dropped.
- VERIFY: N7:11/7 (rung 17) is not written as an individual bit by any routine (only via whole-word writes of N7:11 in LAD_5_SEQ_MOV). Confirm whether this condition is meaningful.
- Asymmetry: rung 4 (channel A) is gated by B3:12/15; rung 5 (channel B) is not. Confirm whether this is intentional.
- Unusual construct: rung 1 contains two TON instructions for the same timer T4:29 (first and last branch). The second restarts the timer in the same scan after the RES. Functional, but should be implemented as a single self-resetting timer on the new platform.
- Inconsistency: the acceleration FIFO (R6:0, rung 12) has length 8 while the other four have length 10, although the N41:0 block has room for 10. Rung 10 wraps it at 8, not at "spot 10" as the comment states.
- Timing saturation: RTO presets (9 s / 15 s) cap the logged durations. Confirm these limits exceed the longest expected phase.
- Data snapshot: T4:45 shows a residual accumulator (349 ms) and the FFL position fields show live values; these are runtime data, not configuration.