

ALT-1

Assembly & Leak Test Machine

Functional Description and Factory Acceptance Test

Field	Value
Document	ALT-1-FD-001
Revision	D (draft)
Date	21 October 2026
Author	Craig Lomax
Status	Complete

Revision history

Rev	Date	Description
A	15 Oct 2026	First draft: scope, architecture, I/O list, run control logic, initial tests.
B	16 Oct 2026	Project renamed ALT-1. Cap press logic, timeouts, sensor check and fault reset added. Fault summary rung. Tests FAT-11 to FAT-18.
C	18 Oct 2026	Machine sequence (FB_Sequence) added as rung 6; later rungs renumbered. Timer presets use CapPress_Timeout constant. Sequence tests FAT-19 to FAT-25. Ladder diagrams drawn graphically. Rung 6 shown without EN/ENO. Automatic heading numbering.
D	21 Oct 2026	I/O addresses assigned. Runtime and plant simulator described (3.1, 3.4). New outputs Lamp_Fault, Lamp_Reset, Lamp_LeakTest and Part_Serial_Out, rungs 14 to 16. Sequence: Fault_Active input, Leak_Testing output, part removal aborts the cycle. Tests FAT-26 to FAT-30.

Table of Contents

1. Introduction.....	3
1.1 Purpose.....	3
1.2 Scope.....	3
1.3 Definitions.....	3
2. Machine overview.....	3
2.1 Product.....	3
2.2 Stations.....	4
3. Control system architecture.....	4
3.1 Platform.....	4
3.2 Program structure.....	4
3.3 Timing.....	4
3.4 Plant simulator.....	5
4. Safety.....	5
5. Variables and I/O.....	5
5.1 Inputs.....	6
5.2 Outputs.....	6
5.3 Internal variables.....	6
6. Logic description.....	7
6.1 Run control.....	7
6.2 Machine sequence.....	9
6.3 Cap press.....	10
6.4 Status outputs and serial number.....	12
6.5 Fault list.....	13
7. Factory acceptance test.....	13
7.1 Run control.....	13
7.2 Cap press.....	14
7.3 Machine sequence.....	15
7.4 Status outputs and serial number.....	15
8. Assumptions and limitations.....	16
Open items.....	16

1. Introduction

1.1 Purpose

This document describes the intended behaviour of the ALT-1 control system and defines the tests used to prove that behaviour. It is written before and alongside the PLC program, and is updated as each function is added.

1.2 Scope

The ALT-1 is a portfolio project: a simulated special purpose machine that assembles and leak tests a small cast valve body. There is no physical machine. The PLC program runs on a soft PLC, and the machine itself is represented by a software simulator.

In scope for version 1:

- Run control, permissives and fault handling
- Load and identify station (serial number assignment)
- Cap fit and press station
- Leak test station, interfacing with a simulated leak test instrument
- Unload and sort station (good parts and locked reject bin)
- Recipe handling for three product variants, including changeover

Deferred to version 2:

- Vision inspection station
- Supervisory HMI and OEE / downtime reporting
- Robot handling

1.3 Definitions

Term	Meaning
Permissive	A condition that must be true before an action is allowed.
Interlock	Logic that prevents an action when it would be unsafe or damaging.
Seal-in	A rung that keeps its own output on after the initiating input is released.
Scan	One PLC cycle: read inputs, execute program, write outputs.
FAT	Factory Acceptance Test: tests run before a machine leaves the builder.
Recipe	A named set of parameters for one product variant.

2. Machine overview

2.1 Product

A small cast aluminium valve body of the kind used in automotive, heating and white goods assemblies. A cap is pressed onto the body, and the assembly is pressure decay leak tested before release.

2.2 Stations

Station	Function
1. Load and identify	Operator loads a part onto a pallet and presses Start. The machine assigns a serial number simulating a laser-marked Data Matrix code.
2. Cap fit and press	A pneumatic cylinder presses the cap onto the body. Extended and retracted positions are confirmed by sensors, with a timeout fault.
3. Leak test	The machine starts a test on the leak test instrument, waits for the result, and handles pass, fail, timeout and instrument fault.
4. Unload	Passed parts go to the good lane. Failed parts go to a locked reject bin.

3. Control system architecture

3.1 Platform

Item	Choice
Soft PLC	OpenPLC runtime (v4) in a Docker container, host networking
Programming	OpenPLC Editor, IEC 61131-3. The editor compiles, uploads to the runtime and starts it.
Cyclic task interval	20 ms
Plant simulator	plant_sim.py: Python, standard library only. Modbus TCP server on port 5020 with an operator panel.
PLC to simulator link	The PLC is the Modbus client. It polls the simulator as a remote device at localhost:5020 every 20 to 50 ms: 9 discrete inputs read, 5 coils and 1 holding register written.

3.2 Program structure

The program is split into two layers with different jobs:

- **Ladder main program:** physical I/O, permissives, interlocks and run control. Only this layer drives outputs. Ladder is used here because it is what maintenance staff are most likely to read on site.
- **Structured Text function blocks:** the machine sequence, leak test communication, timeouts and serial tracking. These are called from the ladder program.

The ST sequence never drives an output directly. It sets request bits, and the ladder layer turns a request into an output only when the relevant interlocks allow it. The sequence can ask for a cylinder to extend; only the interlock layer decides whether it does.

3.3 Timing

The cyclic task runs every 20 ms. No function depends on timing tighter than 100 ms. The host OS is not real-time so some variation in scan interval is expected and acceptable for this application.

3.4 Plant simulator

The simulator stands in for the machine. It models the cap press cylinder (0.4 s travel each way), the push buttons, guard, e-stop, pallet and part, and exposes them to the PLC over Modbus. It can be driven from an operator panel window or from text commands in the terminal. Both send the same commands through a single queue, so they cannot disagree, and every change is logged.

Panel area	Contents
Operator controls	Start, Stop and Reset (held while pressed) and a latching e-stop. Reset is enabled only while the PLC lights Lamp_Reset.
Machine controls	Open and close the guard, load and unload a part, pallet loaded, and the part serial number written by the PLC. Unloading is disabled while Lamp_LeakTest is on.
Cap press	Animated press and guard, press position, and lamps for both position sensors and the solenoid.
PLC status lamps	Running, Fault, Reset available and Leak testing, as written by the PLC.
Fault injection	Press jammed, press slow (3 s travel), and either position sensor stuck on. Test only; not part of the machine.
Event log	Every signal change with a timestamp, and a command line.

The header shows the Modbus link: green while the PLC is polling, amber if polling stops for more than 0.5 s, red when no PLC is connected. The simulator never decides machine state that belongs to the PLC; lamps and button enables follow the PLC outputs.

4. Safety

In a real machine, the emergency stop and guard door would be wired through a safety relay or safety controller, rated to the required performance level, which removes power from hazardous motion independently of the PLC. The PLC only monitors their state, stops the sequence, and reports the condition.

Fail-safe wiring. The emergency stop and Stop push button are wired normally closed, so the input is TRUE when healthy. A broken wire or lost supply reads as FALSE and stops the machine rather than allowing it to continue.

No automatic restart. After any stop or fault, the machine remains stopped until the cause is cleared and the operator presses Start.

5. Variables and I/O

Addresses are those used with the plant simulator. Discrete inputs map from %IX0.0 upwards, coils from %QX0.0 and holding registers from %QW0, in the order the simulator exposes them. Located addresses are used only in the ladder layer; everything else uses named variables. Inputs now come from the simulator, so their initial values only apply when testing by forcing in the IDE.

5.1 Inputs

Name	Address	Description	Healthy / active state	Initial
------	---------	-------------	------------------------	---------

EStop_OK	%IX0.0	Emergency stop status (monitored)	TRUE = healthy (NC)	TRUE
Guard_Closed	%IX0.1	Guard door closed	TRUE = closed	TRUE
Start_PB	%IX0.2	Start push button	TRUE = pressed	FALSE
Stop_PB	%IX0.3	Stop push button	TRUE = not pressed (NC)	TRUE
Reset_PB	%IX0.4	Fault reset push button	TRUE = pressed	FALSE
Pallet_At_Load	%IX0.5	Pallet in load position	TRUE = present	FALSE
Part_Present	%IX0.6	Part detected on pallet	TRUE = present	FALSE
CapPress_Extended	%IX0.7	Cap press at extended position	TRUE = at position	FALSE
CapPress_Retracted	%IX1.0	Cap press at home (retracted) position	TRUE = at position	TRUE

5.2 Outputs

Name	Address	Description
Lamp_Running	%QX0.0	Machine running indicator
CapPress_Extend	%QX0.1	Cap press extend solenoid. Single solenoid, spring return: off means retract.
Lamp_Fault	%QX0.2	Any fault active
Lamp_Reset	%QX0.3	A fault is active and a reset will clear it
Lamp_LeakTest	%QX0.4	Leak test in progress (sequence step 40)
Part_Serial_Out	%QW0	UINT. Serial number of the current part, low 16 bits of Part_Serial

5.3 Internal variables

Name	Type	Description
Fault_Active	BOOL	Any fault present. OR of all fault bits.
Machine_Healthy	BOOL	All permissives satisfied.
Running	BOOL	Machine in run mode (sealed in).
Load_Ready	BOOL	Running, with a pallet and part at the load station. Handed to the ST sequence.
CapPress_Timeout	TIME (constant)	Cap press position timeout, T#2s. Used by both cap press timers.
FB_SEQUENCE	FB_Sequence	Machine sequence instance, called in rung 6.
Part_Serial	DINT	Serial number of the current part.
CapPress_Req_Extend	BOOL	Extend request, driven by the sequence.
CapPress_ExtTimer	TON	Extend timeout timer. Preset CapPress_Timeout.
CapPress_RetTimer	TON	Retract timeout timer. Preset CapPress_Timeout.
CapPress_Ext_TO	BOOL	Extend timer done (Q output).
CapPress_Ret_TO	BOOL	Retract timer done (Q output).

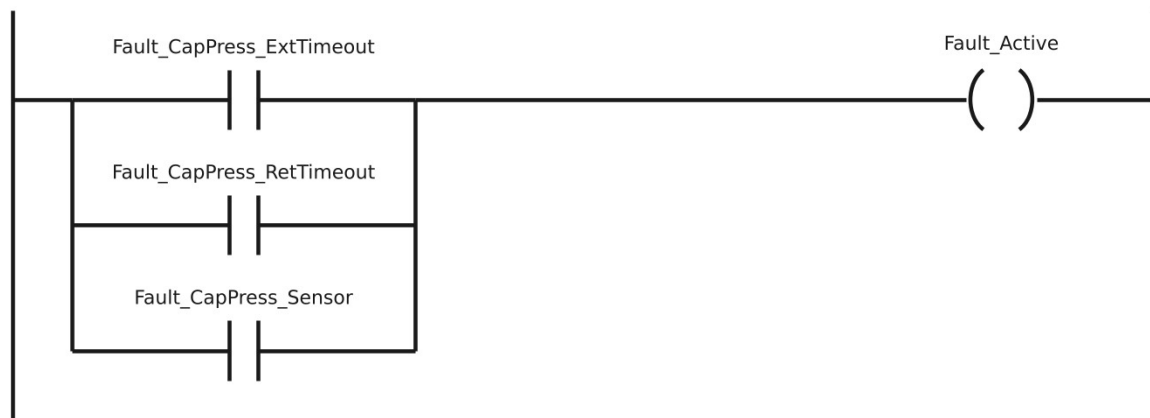
Name	Type	Description
Reset_Allowed	BOOL	Fault causes cleared: press home and sensors agree.
Fault_CapPress_ExtTimeout	BOOL	Latched: press did not reach extended in time.
Fault_CapPress_RetTimeout	BOOL	Latched: press did not return home in time.
Fault_CapPress_Sensor	BOOL	Latched: extended and retracted sensors on at the same time.

6. Logic description

Rungs are listed in program order.

6.1 Run control

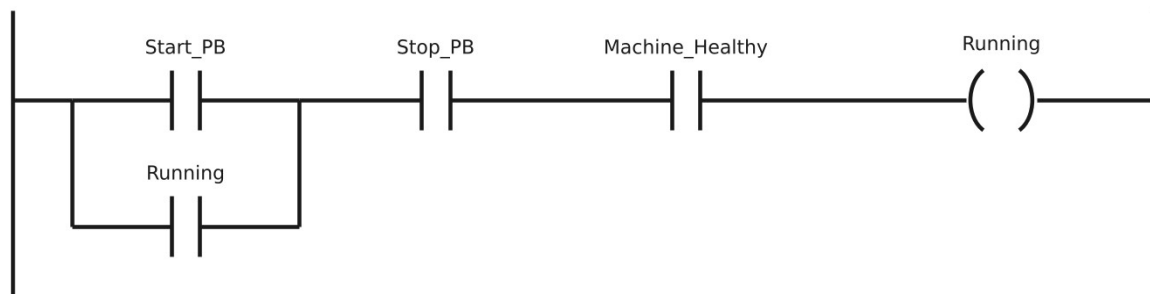
Rung 1, fault summary. Fault_Active is TRUE if any fault bit is set.



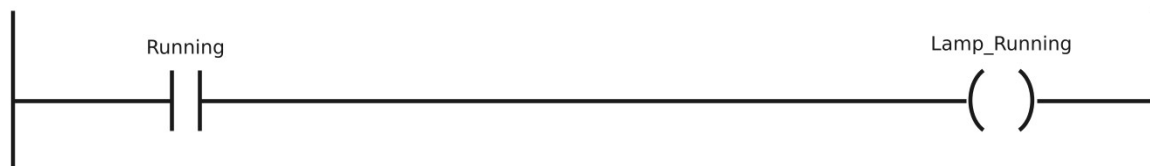
Rung 2, machine healthy. Machine_Healthy is TRUE when the emergency stop is healthy, the guard is closed and no fault is active.



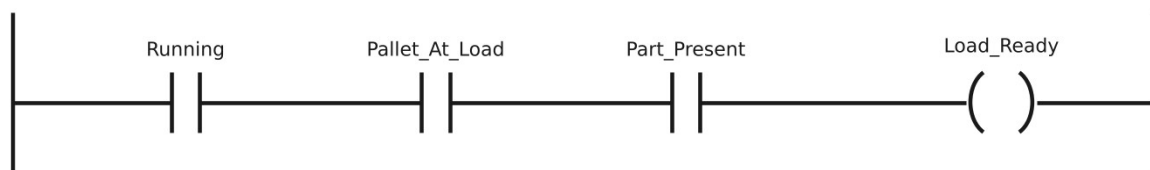
Rung 3, run latch. Pressing Start sets Running, which then holds itself on through the parallel branch. Pressing Stop or losing Machine_Healthy drops Running. It does not come back on when the cause clears; Start must be pressed again.



Rung 4, running lamp. Lamp_Running follows Running.



Rung 5, part ready. Load_Ready is TRUE while the machine is running and a pallet with a part is at the load station.



6.2 Machine sequence

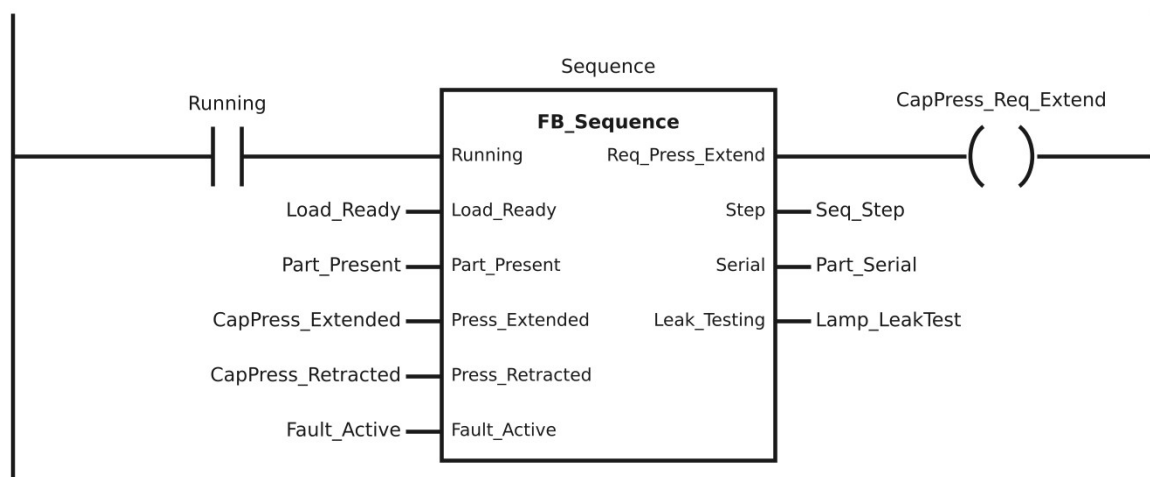
The sequence is a Structured Text function block, FB_Sequence, called once per scan from rung 6 (instance Sequence). It never drives an output. It sets request bits, and the interlock rungs decide whether each request is acted on.

Step	Name	Action	Moves on when
0	Idle	Waits for a part	Load_Ready; serial number assigned
10	Press extend	Requests cap press extend	CapPress_Extended
20	Press dwell	Holds the press down for 500 ms	Dwell timer done
30	Press retract	Request removed; spring returns press	CapPress_Retracted
40	Leak test (stub)	Placeholder: waits 3 s, always passes. Leak_Testing on.	Stub timer done
50	Unload	Waits for the operator to remove the part	Part_Present FALSE
90	Aborted	Machine stopped mid-cycle, or part removed during steps 10 to 40	Part_Present FALSE and Fault_Active FALSE

Design decisions

- **Steps are numbered in tens** so that steps can be inserted later without renumbering.
- **Requests are derived from the step**, in assignments at the end of the block, rather than set and cleared inside individual steps. This applies to Req_Press_Extend and the Leak_Testing status output alike. Neither can be left on by an unexpected jump.
- **Timers are called outside the CASE statement**, every scan, with IN tied to the step. A timer only resets when called with IN FALSE, so calling it inside a step would leave it holding its old value after the step is left.
- **Stopping mid-cycle aborts the part.** A part interrupted partway through cannot be assumed good, so the sequence goes to step 90 and returns to idle only once the part is removed and no fault is active. Resuming mid-cycle was rejected.
- **A part removed mid-cycle also aborts.** If Part_Present drops during steps 10 to 40, the sequence goes to step 90. In version 1 this is not a latched fault; see section 8.

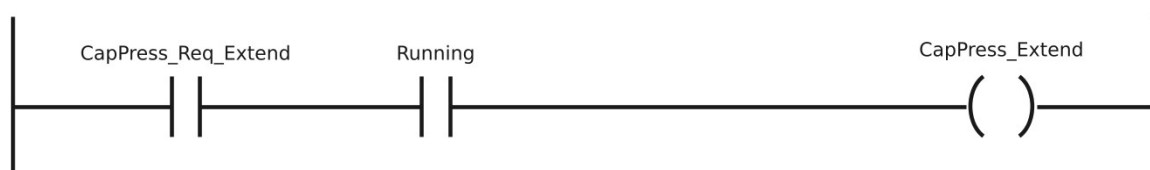
Rung 6, sequence call. FB_Sequence has no EN input, so it runs every scan, including while the machine is stopped; that is what lets it see a stop and abort the cycle. The Running contact only passes a value into the Running input. The Req_Press_Extend output drives CapPress_Req_Extend, which rung 7 turns into an output only when the interlocks allow. CapPress_Extend is written in rung 7 only. Leak_Testing drives Lamp_LeakTest directly, since a status lamp needs no interlock.



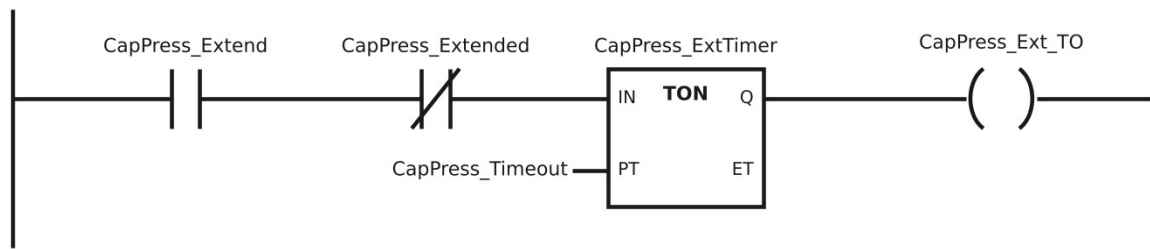
6.3 Cap press

Valve type. The cap press uses a single solenoid, spring return valve. CapPress_Extend on extends the cylinder; off lets the spring return it home. A stop, fault or loss of power therefore always retracts the press. That is acceptable here because the press only applies force to the cap and holds nothing in place. A clamp or lift would need a different choice.

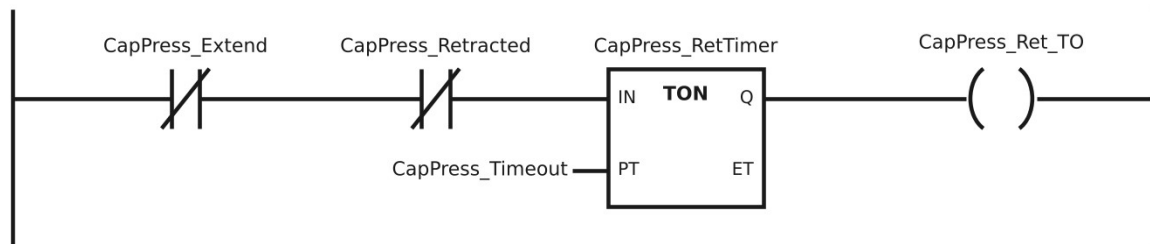
Rung 7, cap press output (interlock layer). The sequence request only reaches the solenoid while the machine is running.



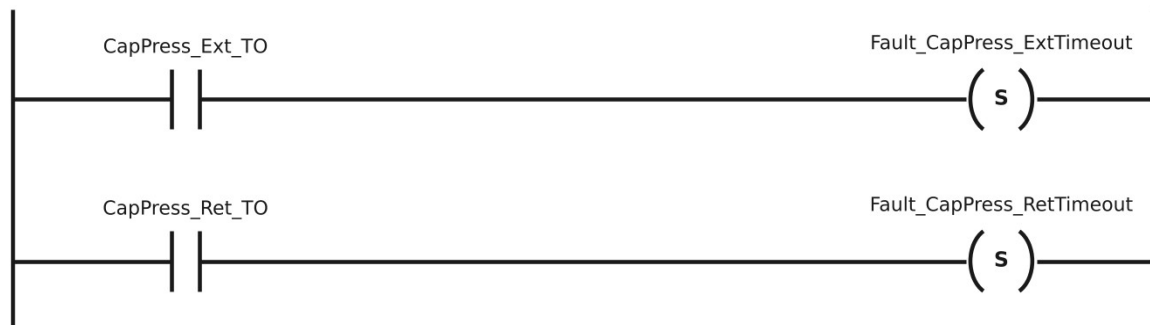
Rung 8, extend timer. Runs while the press is commanded out but not yet extended.



Rung 9, retract timer. Runs while the press is not commanded out but not yet home. This also catches a press that is not home at power-up.



Rung 10, latch timeout faults.



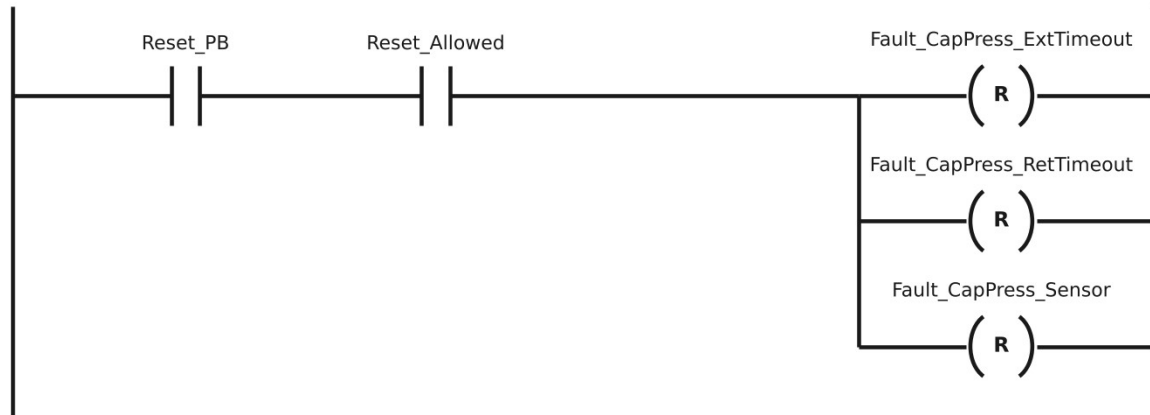
Rung 11, sensor plausibility. Both position sensors on at once means a failed or misplaced sensor.



Rung 12, reset allowed. The cause has gone: press home and the extended sensor off.



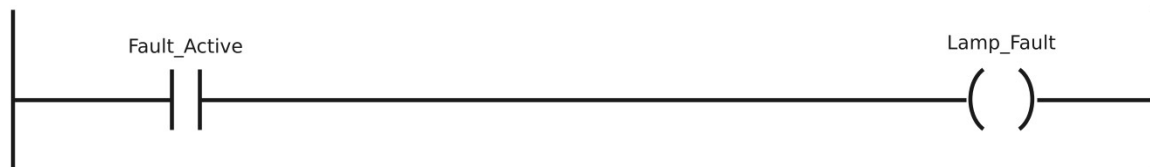
Rung 13, fault reset. Reset only clears faults when the cause has gone. Start is then needed to resume.



Why faults use set/reset coils. A fault must stay latched after the condition that raised it has gone, and clear only on a deliberate reset. The run latch uses a seal-in instead because it must drop the moment any condition fails.

6.4 Status outputs and serial number

Rung 14, fault lamp. Lamp_Fault follows Fault_Active.



Rung 15, reset available lamp. Lamp_Reset is on when a fault is active and Reset_Allowed is TRUE, meaning a reset will clear it. Reset_Allowed alone is TRUE whenever the press is home, so without Fault_Active the lamp would be lit with nothing to reset. The simulator panel enables its Reset button only while this lamp is on.



Rung 16, serial number output. Modbus registers are 16 bits, so the 32-bit Part_Serial is converted to a UINT and sent in one holding register. A plain MOVE cannot be used because the types differ. The value wraps to 0 after 65,535 parts.



6.5 Fault list

Fault	Response	Recovery	Status
Emergency stop	Machine stops	Release e-stop, press Start	Implemented
Guard open	Machine stops	Close guard, press Start	Implemented
Cap press extend timeout	Fault latched, machine stops, press retracts	Return press home, Reset, Start	Implemented
Cap press retract timeout	Fault latched, machine stops	Return press home, Reset, Start	Implemented
Cap press not home at power-up	Raised as retract timeout (intentional)	Return press home, Reset, Start	Implemented
Cap press sensor conflict	Fault latched, machine stops	Fix sensor, Reset, Start	Implemented
Part removed mid-cycle	Part aborted (step 90); no fault in v1	None; next part starts a new cycle	Implemented

7. Factory acceptance test

Tests can be run by forcing variables in the IDE debugger, or against the plant simulator (3.4) using its panel or commands. With the simulator, the cylinder moves by itself, and the fault conditions in FAT-12, FAT-15 and FAT-16 are created with fault injection (press jammed, press slow, sensor stuck on). Unless stated, each test starts with all permissives healthy (EStop_OK, Guard_Closed, Stop_PB and CapPress_Retracted TRUE, no faults) and the machine stopped.

When testing by forcing, the tester plays the part of the cylinder by forcing its sensors. If 2 s is too short to do this by hand, the timer presets may be raised to T#5s for testing. Record the preset used in the Result column, and return it to T#2s afterwards.

7.1 Run control

Test	Action	Expected result	Result	Date / initials
FAT-01	Press and release Start	Running TRUE and remains TRUE after Start released; Lamp_Running TRUE		
FAT-02	Running. Press Stop	Running FALSE; Lamp_Running FALSE		
FAT-03	Running. Open guard	Running FALSE		

Test	Action	Expected result	Result	Date / initials
FAT-04	After FAT-03, close guard	Running stays FALSE until Start is pressed		
FAT-05	Guard open. Press Start	Running stays FALSE		
FAT-06	Running. Pallet and part present	Load_Ready TRUE		
FAT-07	After FAT-06, remove part	Load_Ready FALSE		
FAT-08	Running. E-stop pressed (EStop_OK FALSE)	Running FALSE		
FAT-09	After FAT-08, release e-stop	Running stays FALSE until Start is pressed		
FAT-10	Stopped. Pallet and part present	Load_Ready FALSE		

7.2 Cap press

Test	Action	Expected result	Result	Date / initials
FAT-11	Running. Set CapPress_Req_Extend TRUE, then within the preset set CapPress_Retracted FALSE and CapPress_Extended TRUE	CapPress_Extend TRUE; no fault		
FAT-12	Running. Set CapPress_Req_Extend TRUE, leave CapPress_Extended FALSE	After preset, Fault_CapPress_ExtTimeout set; Running and CapPress_Extend FALSE		
FAT-13	After FAT-12, with press not home, press Reset	Fault stays set		
FAT-14	After FAT-13, set CapPress_Retracted TRUE, press Reset, then Start	Fault clears on Reset; machine runs only after Start		
FAT-15	Running, press extended. Set CapPress_Req_Extend FALSE, leave CapPress_Retracted FALSE	CapPress_Extend FALSE; after preset, Fault_CapPress_RetTimeout set		
FAT-16	Set CapPress_Extended and CapPress_Retracted both TRUE	Fault_CapPress_Sensor set; machine stops		
FAT-17	Stopped. Set CapPress_Req_Extend TRUE	CapPress_Extend stays FALSE		
FAT-18	Start simulation with CapPress_Retracted FALSE	After preset, Fault_CapPress_RetTimeout set; Start has no effect until press is home and Reset pressed		

7.3 Machine sequence

Test	Action	Expected result	Result	Date / initials
FAT-19	Running. Set Pallet_At_Load and Part_Present TRUE	Step 0 to 10; Part_Serial increases by 1; CapPress_Extend TRUE		
FAT-20	Set CapPress_Retracted FALSE, CapPress_Extended TRUE	Step 20; after 500 ms, Step 30 and CapPress_Extend FALSE		
FAT-21	Set CapPress_Extended FALSE, CapPress_Retracted TRUE	Step 40; after 3 s, Step 50		
FAT-22	Set Part_Present FALSE	Step 0; Part_Serial unchanged		
FAT-23	Repeat FAT-19 and FAT-20 to reach step 20, then press Stop	Step 90; CapPress_Extend FALSE		
FAT-24	After FAT-23, with part still present, press Start	Step stays 90; no press movement requested		
FAT-25	After FAT-24, set Part_Present FALSE	Step 0; next part starts a new cycle with a new serial number		

7.4 Status outputs and serial number

Test	Action	Expected result	Result	Date / initials
FAT-26	Create a cap press extend timeout (simulator: press jammed, then run a cycle)	Lamp_Fault on; Lamp_Reset off while the press is not home; panel Reset disabled		
FAT-27	After FAT-26, clear the jam so the press returns home	Lamp_Reset on; panel Reset enabled; Reset clears the fault and both lamps go off		
FAT-28	Run a cycle and watch step 40	Lamp_LeakTest on for 3 s; panel Unload disabled; both released at step 50		
FAT-29	Run two cycles	Part_Serial_Out equals Part_Serial and increases by 1 per part; panel shows the same number		
FAT-30	Remove the part (command unload) during step 20	Step 90, then 0 with no fault active; no further press movement; serial unchanged		

8. Assumptions and limitations

- Simulation only. No physical I/O, actuators or instruments.
- Safety functions are modelled as monitored inputs only. A real machine would require a separately designed and validated safety system.
- The software PLC is not real-time; scan timing varies.
- Leak test behaviour is simulated. The real instrument protocol, test timings and limits are place-holders.
- Pallets do not move; the press acts at the load station (see 2.3).
- Part_Serial_Out is a 16 bit unsigned integer and wraps after 65,535 parts.

Open items

- Exclude step 50 from the stop-abort condition, so a finished part waiting to be unloaded is not marked as aborted.
- Consider a homing routine or "press not home" status in place of the power-up fault (v2).
- Define the simulated leak tester protocol and its fault cases.
- Consider latching a fault for a part removed mid-cycle, so the operator must acknowledge an untested part.
- Model separate load and press stations, with pallet transfer between them.
- Send the serial in two registers if more than 65,535 parts are needed.