
Reliability, Availability, and Bottleneck Analysis

Fastening, Barcode Etching, and Traceability Station

Engineering Review of 26 PLC Status and Interlock Signals

Prepared by: Sam Khalaf

Organization: Acme

Facility: Sterling Heights, Michigan

Revision: 1.0

Issue Date: June 23, 2026

Purpose. This report evaluates the proper reliability and production-performance model for a station (Fastening, Barcode Etching, and Traceability Station). It also examines whether dividing the process into smaller modules can reduce bottleneck impact or improve output.

Contents

1	Executive Summary	1
2	Purpose and Scope	1
3	Station Architecture and Signal Organization	1
4	Critical Modeling Correction	3
4.1	A PLC tag is not automatically a reliability component	3
4.2	Several conditions are mutually exclusive	3
4.3	The conditions are evaluated at different times	3
5	Recommended Phase-Based Cycle Logic	4
6	Reliability, Availability, Quality, and Bottleneck Metrics	5
6.1	Per-cycle functional reliability	5
6.2	First-pass yield	5
6.3	Availability	5
6.4	Bottleneck performance	5
7	Interpretation of the 26-Element Series Calculation	6
8	Effect of Segmentation	6
8.1	Logical segmentation	6
8.2	Physical segmentation with operational independence	7
9	Control and Diagnostic Design Recommendations	7
9.1	Separate raw inputs, derived states, and results	7
9.2	Use result codes and retain measured values	7
9.3	Add contradiction, timeout, and stale-data diagnostics	8
9.4	Keep safety logic distinct	8
10	Recommended Data Collection Plan	8
11	Engineering Recommendations	9
12	Conclusion	9
A	Signal Mapping and Engineering Classification	10
B	Minimum Quantitative Data by Subsystem	12

1 Executive Summary

The station is represented by 26 PLC tags covering part detection, fixture and clamp position, compressed air, safety status, robot readiness and position, screw feeding, fastening results, barcode etching, barcode verification, traceability, and station-level status. These tags are valuable for control, diagnostics, and sequence monitoring, but they are not equivalent to 26 independent physical components connected in series.

The original series calculation,

$$0.99^{26} = 0.770043 \approx 77.00\%,$$

is mathematically correct only under a narrow hypothetical assumption: each item is a distinct and independent physical function, all 26 functions are required for the same mission, and each function has reliability 0.99. The actual tag architecture does not meet those conditions. Several tags describe mutually exclusive positions or results, several are calculated outcomes rather than physical devices, and the tags are evaluated at different times during the cycle.

The number of PLC bits is a rough estimate of the number of reliability blocks. The station should be modeled using physical subsystems, failure modes, and phase-specific success events. The 77.00% result may be retained as an illustrative sensitivity case, but it should not be presented as the station's measured or predicted reliability without supporting component data.

Segmentation alone does not improve the reliability of a true series process. A physical split between fastening and etching/verification can nevertheless improve availability and throughput when it adds a buffer, independent fault recovery, fault containment, and reliable part-identity handshakes. The size of that benefit must be established from actual cycle-time, downtime, repair, retry, and quality data.

2 Purpose and Scope

This report has six objectives:

1. organize the 26 PLC signals into a clear subsystem architecture;
2. distinguish control-state information from physical reliability elements;
3. define an appropriate phase-based cycle-success model;
4. explain the meaning and limitation of the 0.99^{26} calculation;
5. evaluate when segmentation can improve availability or throughput; and
6. recommend data collection and control improvements for a defensible reliability study.

No measured failure history, cycle-time distribution, MTBF, MTTR, reject rate, or buffer-capacity data were provided. The numerical result in this report is therefore illustrative. It is not a validated prediction of field performance.

3 Station Architecture and Signal Organization

The 26 signals are best organized by subsystem and control purpose, as shown in Table 1. Structured names improve readability, filtering, troubleshooting, historian configuration, and future expansion of the PLC program.

Table 1: Recommended functional grouping of the PLC tags

Subsystem	Recommended tags	Primary purpose
Part and fixture	Part.Present, Fixture.At_Home, Fixture.Clamped, Fixture.Clamp_Extended, Fixture.Clamp_Retracted	Detect the workpiece and confirm the required fixture or clamp state at the correct sequence step.
Utilities and safety	Utilities.Air_Pressure_OK, Safety.Circuit_OK	Provide global permissives. These shared conditions can affect several subsystems simultaneously.
Robot	Robot.Ready, Robot.At_Home, Robot.At_Fastening_Position	Confirm robot availability and phase-specific position. Home and fastening position are normally mutually exclusive.
Fastening	Fastening.Screw_Present, Fastening.Feeder_Ready, Fastening.Tool_Ready, Fastening.Torque_Within_Limits, Fastening.Torque_Out_Of_Limits, Fastening.Angle_Within_Limits, Fastening.Complete	Control screw delivery and fastening and record whether the fastening result meets the required torque and angle limits.
Etching	Etcher.Ready, Etcher.Data_Ready, Etcher.Complete	Confirm marker readiness, valid data availability, and completion of the etching operation.
Verification and traceability	Scanner.Ready, Scanner.Barcode_Read_Successful, Traceability.Record_Complete	Verify the applied barcode and confirm that the production record was stored for the correct part.
Station supervision	Station.Clear, Station.Cycle_Complete, Station.Fault_Active	Summarize station state, cycle completion, and active fault status for line control and HMI display.

Figure 1 shows the functional sequence. Safety, compressed air, controls, and part identity support multiple process stages and therefore create dependencies that a simple independent-series model does not capture.

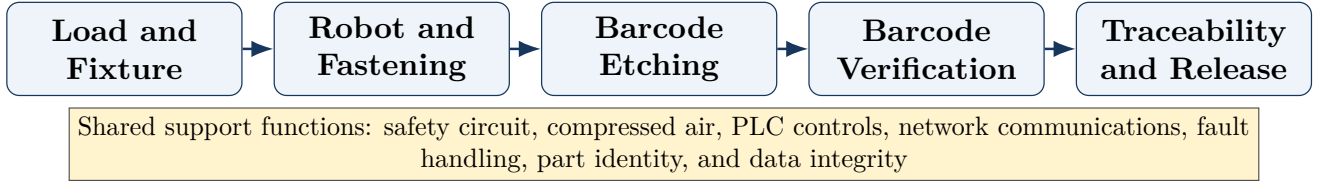


Figure 1: Functional process flow and shared support functions

4 Critical Modeling Correction

4.1 A PLC tag is not automatically a reliability component

A PLC tag may represent a raw input, a communication state, a calculated interlock, a latched result, a command acknowledgement, or an HMI summary. One physical device may generate several tags, while one tag may depend on several devices and software conditions. Counting tags therefore does not establish the number of statistically independent failure opportunities.

For example, `Fixture.Clamp_Extended` and `Fixture.Clamp_Retracted` may be generated by two sensors on one actuator. Their behavior also depends on the same cylinder, valve, compressed-air source, wiring, input module, and PLC logic. Treating these two bits as independent components ignores their shared failure causes.

4.2 Several conditions are mutually exclusive

A successful cycle does not require all 26 bits to be true at the same time. Important examples are:

- `Fixture.Clamp_Extended` and `Fixture.Clamp_Retracted` normally represent opposite end positions;
- `Robot.At_Home` and `Robot.At_Fastening_Position` normally cannot be true simultaneously;
- `Fastening.Torque_Within_Limits` and `Fastening.Torque_Out_Of_Limits` should be mutually exclusive after one fastening attempt;
- `Station.Fault_Active` must normally be false for a successful cycle; and
- `Station.Clear` may describe an empty station and therefore may be false while `Part.Present` is true.

An “AND all 26 signals” expression therefore does not describe the station sequence.

4.3 The conditions are evaluated at different times

Readiness and position tags must be satisfied during the phase in which they are relevant. The robot may begin at home, move to the fastening position, perform the fastening operation, and return home. The correct success condition is that each required transition occurs within its time limit and produces the correct result, not that every position bit remains true throughout the cycle.

Let S_k be the event that phase k completes successfully within its required limits. The cycle-success event is

$$S_{\text{cycle}} = S_1 \cap S_2 \cap \cdots \cap S_m, \quad (1)$$

and the cycle reliability is

$$R_{\text{cycle}} = P(S_1 \cap S_2 \cap \cdots \cap S_m). \quad (2)$$

Only when the phase-success events are independent may this be simplified to

$$R_{\text{cycle}} = \prod_{k=1}^m P(S_k). \quad (3)$$

Shared air, power, controls, communications, recipes, and part condition can make the phase events dependent.

5 Recommended Phase-Based Cycle Logic

Table 2 organizes the tags into phase-specific conditions. The final machine sequence should match the functional specification and safety design, but this structure prevents unrelated or mutually exclusive states from being treated as simultaneous requirements.

Table 2: Example phase-based success criteria

Phase	Conditions evaluated during the phase	Successful outcome
Ready / idle	<code>Safety.Circuit_OK</code> ; <code>Utilities.Air_Pressure_OK</code> ; <code>Robot.Ready</code> ; <code>Robot.At_Home</code> ; <code>Fixture.At_Home</code> ; no station fault	The station accepts a new cycle request.
Load and clamp	<code>Part.Present</code> ; clamp command issued; expected clamp-position feedback received within the allowed time	<code>Fixture.Clamped</code> is latched and the clamp-position feedback is valid.
Position and fasten	<code>Robot.At_Fastening_Position</code> ; <code>Fastening.Screw_Present</code> ; <code>Fastening.Feeder_Ready</code> ; <code>Fastening.Tool_Ready</code>	<code>Fastening.Complete</code> ; torque and angle pass; no fastening fault or timeout.
Etch	<code>Etcher.Ready</code> ; <code>Etcher.Data_Ready</code> ; correct part identity associated with the request	<code>Etcher.Complete</code> with no marker fault.
Verify barcode	<code>Scanner.Ready</code> ; valid trigger; expected barcode data available for comparison	<code>Scanner.Barcode_Read_Successful</code> .
Record and release	Required fastening, etching, and scan results latched; traceability service available; safe release state	<code>Traceability.Record_Complete</code> ; <code>Station.Cycle_Complete</code> ; part released without an active fault.

A practical implementation latches the result of each phase for the current part. The final cycle disposition can then be defined as follows:

Derived result	Illustrative logic
Fastening pass	<code>Fastening.Complete AND Fastening.Torque_Within_Limits AND Fastening.Angle_Within_Limits AND NOT Fastening.Torque_Out_Of_Limits</code>
Cycle pass	<code>Fastening pass AND Etcher.Complete AND Scanner.Barcode_Read_Successful AND Traceability.Record_Complete AND NOT Station.Fault_Active</code>

The actual implementation should also include phase timeouts, abort conditions, retry limits, result reset rules, and confirmation that every latched result belongs to the current part.

6 Reliability, Availability, Quality, and Bottleneck Metrics

Reliability, availability, quality, and bottleneck performance are related but different. Separating the metrics prevents equipment faults, quality rejects, and data delays from being combined into one ambiguous percentage.

6.1 Per-cycle functional reliability

A practical production estimate is

$$\hat{R}_{\text{cycle}} = \frac{N_{\text{cycles completed without an unplanned fault, retry, or recovery}}}{N_{\text{cycles started}}}. \quad (4)$$

The definition must state whether an automatic retry is considered normal operation. First-attempt success and eventual success should be reported separately.

6.2 First-pass yield

Quality performance is better represented by

$$\text{FPY} = \frac{N_{\text{parts passing fastening, marking, scan, and traceability on the first attempt}}}{N_{\text{parts entering the station}}}. \quad (5)$$

This distinguishes a product-quality failure from an equipment-availability failure.

6.3 Availability

For a repairable subsystem, a common steady-state estimate is

$$A = \frac{\text{MTBF}}{\text{MTBF} + \text{MTTR}}, \quad (6)$$

where MTBF is mean time between failures and MTTR is mean time to repair and restore production. Segmentation may increase availability by reducing fault-isolation and recovery time even when it does not change component failure probability.

6.4 Bottleneck performance

The station should be identified as a bottleneck from production behavior, not from its tag count. Relevant measures include:

- average and 95th-percentile cycle time;
- effective capacity relative to upstream supply and downstream demand;
- blocked, starved, faulted, and planned-stop time;
- fault frequency and downtime by subsystem;
- retry and rework frequency;
- buffer-full and buffer-empty duration; and
- good parts per planned production hour.

A station can have many signals without being the bottleneck. Conversely, one slow or frequently faulted operation can constrain the line even when it uses very few PLC tags.

7 Interpretation of the 26-Element Series Calculation

The original calculation remains useful as a sensitivity example if the following assumptions are explicitly stated:

1. each of the 26 items represents a different physical function;
2. all 26 functions are required for the same mission;
3. each function has reliability $R_i = 0.99$ for that mission;
4. failures are statistically independent; and
5. no bypass, retry, rework, standby equipment, or alternate path exists.

Under these assumptions,

$$R_{\text{series}} = \prod_{i=1}^{26} R_i = 0.99^{26} = 0.770043, \quad (7)$$

so

$$\boxed{R_{\text{series}} \approx 77.00\%}. \quad (8)$$

The probability of at least one mission failure is

$$1 - R_{\text{series}} = 0.229957 \approx 22.996\%. \quad (9)$$

If the 26 hypothetical elements are separated into two groups of 13, each group has reliability

$$R_{13} = 0.99^{13} = 0.877521 \approx 87.75\%. \quad (10)$$

Because both groups remain required,

$$R_{\text{total}} = R_{13}R_{13} = 0.99^{26} \approx 77.00\%. \quad (11)$$

More generally, grouping a true series system into m segments does not change its mission reliability:

$$R_{\text{total}} = 0.99^{n_1} 0.99^{n_2} \dots 0.99^{n_m} \quad (12)$$

$$= 0.99^{n_1+n_2+\dots+n_m} \quad (13)$$

$$= 0.99^{26}, \quad (14)$$

where $n_1 + n_2 + \dots + n_m = 26$.

The arithmetic is correct, describes a hypothetical 26-element independent series system, not the actual phase-dependent control sequence represented by the tag list.

8 Effect of Segmentation

8.1 Logical segmentation

Organizing the PLC program into fixture, robot, fastening, etching, scanning, traceability, and station-level structures improves maintainability and diagnostic clarity. It can reduce programming errors and speed troubleshooting, but naming or software grouping alone does not change physical reliability.

8.2 Physical segmentation with operational independence

A practical physical division is:

- **Module A - Load, fixture, robot, and fastening;** and
- **Module B - Etching, barcode verification, and traceability.**

A controlled buffer between the modules can allow one module to continue temporarily while the other is stopped. This can increase good output by preventing short disturbances from immediately stopping the entire process. The benefit depends on buffer capacity, cycle-time balance, failure and repair distributions, and whether the modules can recover independently.

To create a real benefit, segmentation should include:

1. independent module state machines and fault recovery;
2. explicit Ready, Busy, Complete, Faulted, and Part Available handshakes;
3. reliable transfer of part identity and process results;
4. a defined quarantine path for failed or incomplete parts;
5. buffer-full and buffer-empty handling;
6. safe restart logic that prevents duplicate fastening, duplicate etching, or mismatched records; and
7. analysis of shared air, power, PLC, network, and safety dependencies.

A finite-buffer simulation or production-data study is more appropriate than a simple series equation for estimating the throughput benefit.

9 Control and Diagnostic Design Recommendations

9.1 Separate raw inputs, derived states, and results

The signal architecture should clearly distinguish:

- **raw inputs:** clamp sensors, part-present sensor, screw-present sensor;
- **device states:** robot ready, etcher ready, scanner ready;
- **derived interlocks:** fixture clamped, station ready;
- **quality outcomes:** torque pass, angle pass, barcode verification pass;
- **sequence outcomes:** fastening complete, etching complete, cycle complete; and
- **diagnostics:** active fault, timeout, discrepancy, or communication fault.

This separation prevents a calculated summary bit from being counted as an additional independent physical component.

9.2 Use result codes and retain measured values

For fastening, a result record is preferable to relying only on Boolean torque bits. At minimum, retain:

- result code: not evaluated, pass, fail, aborted, or communication fault;
- actual torque and acceptance limits;
- actual angle and acceptance limits;
- selected program or recipe;
- controller fault code and tool cycle counter; and
- timestamp and part identifier.

Boolean HMI tags such as `Fastening.Torque_Within_Limits` can then be derived from the result record.

9.3 Add contradiction, timeout, and stale-data diagnostics

Diagnostics should detect:

- both clamp end-position sensors true after motion completion;
- neither clamp end-position sensor true after the motion timeout;
- robot home and fastening-position confirmations true simultaneously;
- both torque-pass and torque-fail results true for one attempt;
- a completion or result bit remaining latched from the previous part; and
- barcode or traceability data that does not match the current part identifier.

Each event should generate a unique fault code, timestamp, phase identifier, and recommended operator or maintenance action.

9.4 Keep safety logic distinct

`Safety.Circuit_OK` is an indication that the safety system permits production operation. It is not, by itself, evidence of functional-safety performance. Safety functions should remain within the validated safety architecture and should not be reduced to an ordinary production reliability calculation.

10 Recommended Data Collection Plan

For each production attempt, record:

1. part identifier, product variant, recipe, and cycle number;
2. cycle start and end timestamps;
3. start and completion time for each phase;
4. phase result: pass, retry, fail, abort, or bypass;
5. first-out fault code and originating subsystem;
6. recovery start and completion time;
7. fastening torque, angle, limits, and result code;

8. etcher result and retry count;
9. scanner result, read attempts, and decoded value;
10. traceability result and transaction response time; and
11. blocked, starved, buffer-full, and buffer-empty duration when segmentation is used.

These records allow the station to be evaluated by shift, product, recipe, subsystem, and fault mode. A Pareto chart of lost production time is usually more actionable than one aggregate reliability percentage because it identifies the failures that remove the most good-parts capacity.

11 Engineering Recommendations

The recommended next steps are:

1. adopt the structured tag hierarchy and publish a controlled signal dictionary;
2. document the station state machine with phase-specific permissives, timeouts, and completion criteria;
3. define station fault, retry, quality reject, eventual pass, and first-pass success consistently;
4. collect actual cycle, failure, recovery, and quality data by subsystem;
5. construct a reliability block diagram from physical functions and common dependencies, not the number of PLC tags;
6. perform an FMEA focused on clamp sensing, compressed air, robot availability, screw delivery, fastening quality, etcher communication, barcode verification, and traceability integrity;
7. evaluate the fastening-to-etching split using actual buffer capacity and downtime distributions; and
8. prioritize improvements using lost good parts per hour, MTTR, failure frequency, and first-pass yield.

12 Conclusion

The fastening, barcode etching, and traceability station contains 26 useful PLC status and interlock tags, but it is not correctly represented as 26 independent sensors connected in series. Several signals are mutually exclusive, some are required only during specific phases, some are derived from other signals, and several share common physical causes. The calculation $0.99^{26} \approx 77.00\%$ should therefore be presented only as a hypothetical series-system example.

Actual reliability should be estimated from physical subsystem failures and phase-success data. Availability should be measured from uptime and recovery behavior, quality from first-pass yield, and bottleneck impact from effective capacity and lost production time. Segmentation can provide a meaningful benefit when it creates independent modules, a controlled buffer, clear handshakes, part-identity integrity, and localized recovery. Merely dividing the tag list or PLC code into smaller groups does not change physical reliability.

The strongest engineering approach combines structured controls, event-based data collection, subsystem reliability modeling, and throughput analysis. That approach provides a defensible basis for deciding whether to improve individual devices, reduce repair time, add a buffer, redesign the sequence, or provide redundancy.

A Signal Mapping and Engineering Classification

Table 3: Mapping from the original tag names to the recommended structured names

Original tag	Recommended tag	Classification	Engineering interpretation
Part_Present	Part.Present	Input / detection	Confirms a workpiece is present. Its state may be opposite to Station.Clear , depending on the station definition.
Fixture_Home	Fixture.At_Home	Position state	Confirms the fixture is in its defined home or load position. This is a phase-specific state.
Fixture_Clamped	Fixture.Clamped	Derived interlock	Confirms the complete clamped condition, potentially using position, pressure, and command-state checks.
Clamp_Extended	Fixture.Clamp_Extended	Position input	Indicates one clamp end position and should be evaluated with the command and motion timeout.
Clamp_Retracted	Fixture.Clamp_Retracted	Position input	Indicates the opposite end position and is normally mutually exclusive with extended feedback after motion settles.
Air_Pressure_OK	Utilities.Air_Pressure_OK	Global permissive	Represents a shared utility and a possible common-cause event affecting several pneumatic functions.
Safety_OK	Safety.Circuit_OK	Safety indication	Indicates that the safety system permits production motion; keep it separate from ordinary production reliability.
Robot_Ready	Robot.Ready	Device state	Indicates that the robot controller is available and not reporting a blocking fault.
Robot_Home	Robot.At_Home	Position state	Confirms the robot is at home and is normally false while the robot is at the fastening position.
Robot_In_Position	Robot.At_Fastening_Position	Position state	Uses a specific position name and is evaluated before fastening begins.
Screw_Present	Fastening.Screw_Present	Input / device state	Confirms a screw is available at the required location.
Screw_Feeder_Ready	Fastening.Feeder_Ready	Device state	Indicates that the feeder can support the next fastening attempt.
Tool_Ready	Fastening.Tool_Ready	Device state	Indicates that the fastening controller and tool are ready for the selected program.
Torque_OK	Fastening.Torque_Within_Limits	Quality result	Indicates that measured torque met the acceptance limits for the current attempt.
Torque_NOK	Fastening.Torque_Out_Of_Limits	Quality result	Indicates torque failure. Prefer a single result code as the source of truth and derive this Boolean.
Angle_OK	Fastening.Angle_Within_Limits	Quality result	Indicates that measured fastening angle met its acceptance limits.

Continued on next page

Table 3 continued

Original tag	Recommended tag	Classification	Engineering interpretation
Fasten_Complete	Fastening.Complete	Completion state	Confirms that the attempt completed; completion does not necessarily mean the result passed.
Etcher_Ready	Etcher.Ready	Device state	Indicates that the marking device is ready for the next command.
Barcode_Data_Ready	Etcher.Data_Ready	Data permissive	Confirms that valid data for the current part is available before etching.
Etch_Complete	Etcher.Complete	Completion state	Confirms etching completed. A separate result or fault code should identify pass, retry, abort, or communication failure.
Scanner_Ready	Scanner.Ready	Device state	Indicates that the reader is ready for a trigger and has no blocking fault.
Barcode_Read_OK	Scanner.Barcode_Read_Successful	Verification result	Confirms that the barcode was decoded and, where required, matched the expected part data.
Traceability_OK	Traceability.Record_Complete	Transaction result	Confirms that the required production record was written for the correct part.
Station_Clear	Station.Clear	Summary state	Indicates that the station is clear for loading or transfer. Define whether this means no part, no interference, or both.
Cycle_Complete	Station.Cycle_Complete	Completion state	A station-level result asserted after all required process and traceability criteria are complete.
Fault_Active	Station.Fault_Active	Fault summary	Indicates one or more active faults. A successful cycle normally requires this tag to be false.

B Minimum Quantitative Data by Subsystem

Table 4: Minimum data needed for a quantitative model

Subsystem	Failure and recovery data	Process and timing data
Fixture and clamp	Sensor discrepancy, valve or cylinder fault, pressure-related stop, recovery time	Extend and retract time, clamping retries, part-location failures
Robot	Controller faults, communication faults, protective stops, recovery duration	Move time, wait time, position timeout, blocked and starved time
Fastening	Feeder jams, missed screws, tool or controller faults, maintenance duration	Torque and angle results, fastening time, retries, seating failures
Etcher	Device faults, communication failures, restart time	Etch time, data-preparation delay, retry and reject counts
Scanner	Reader or communication faults, cleaning or adjustment downtime	Read time, no-read rate, mismatch rate, retry count
Traceability	Server, network, transaction, or database fault duration	Response time, write failures, duplicate or mismatched records
Utilities and controls	Air, power, PLC, network, and common-cause events	Total station impact and number of affected subsystems