

NANOARC

Control System and Human-Interface Architecture Manual

Revision A.2 - Operator Display Semantics

AGC-inspired architecture with contextual Program / Verb / Noun display semantics

NanoArc layered control architecture

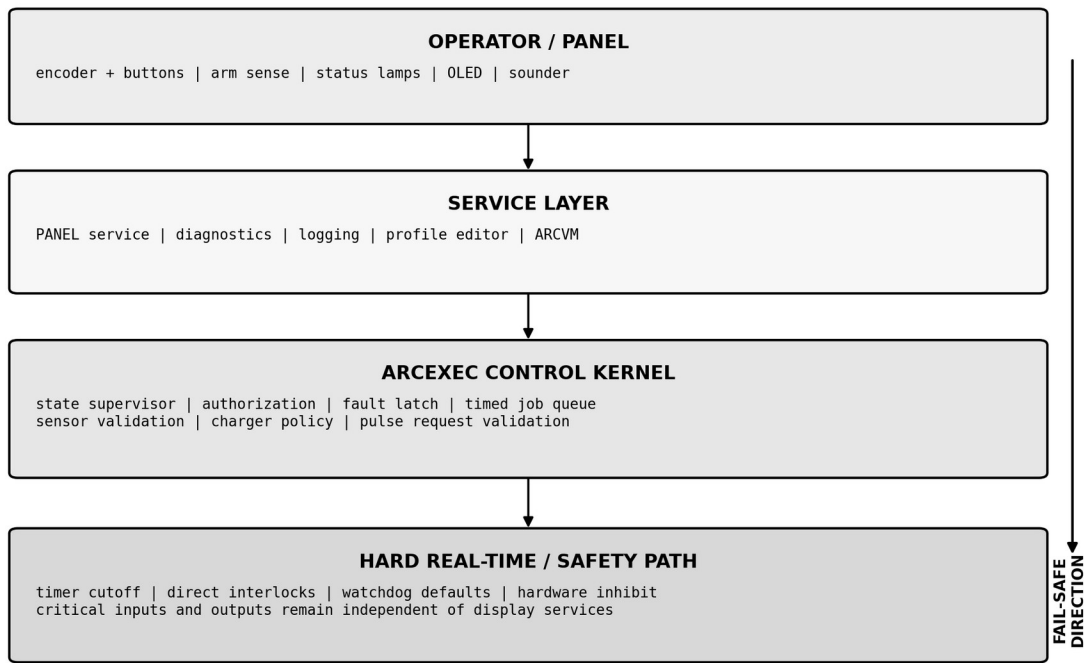


Figure 1. System-level division between operator services and the safety-critical path.

Document identifier	NA-CSHID-001
Revision	A.2
Date	2026-06-22
Status	Operator display semantics integrated

This manual defines control and interface architecture only. It is not an energy-stage construction or operating procedure.

Document Control

Revision	Date	Status	Summary
0.1 / A	2026-06-19	Baseline	Initial control-system, scheduler, interpreter, memory, I/O and HID architecture.
A.1	2026-06-22	Integrated	Artemis 72 Executive, Waitlist, PINBALL, display, alarm and restart principles incorporated into ACE.
A.2	2026-06-22	Display semantics	Replaced fixed DSKY register presentation with typed noun component fields and contextual OLED templates; restored body-level command and display arbitration sections.

Contents

1. Purpose and Scope
2. Design Premise
3. System Partitioning
4. Hardware Platform and Resource Envelope
5. Data Representation
6. Planned Memory Map
7. ARCEXEC Executive
8. Interrupt Processing and Timing Ownership
9. Authoritative Machine-State Model
10. Virtual I/O Channels
11. ARCVM Compact Profile Interpreter
12. Human-Interface Architecture
13. Display Service
14. Input and Output Assignment Policy
15. Fault, Alarm and Inhibit Architecture
16. Restart and Recovery
17. Persistent Storage
18. Diagnostics and Testability
19. Coding and Verification Rules
20. Implementation Sequence
21. Artemis 72 Source-Derived Architecture
- Appendix A. Summary Tables
- Appendix B. Preliminary Fault Catalogue

Appendix C. Open Decisions

References

DESIGN NOTE: The document intentionally uses a manual-like structure similar to historical flight-computer documentation: architecture, data, memory, interrupts, instruction semantics and I/O are specified separately so each can be reasoned about and tested in isolation.

1. Purpose and Scope

NanoArc requires a compact control computer that supervises stored energy, validates operator intent, sequences controlled actions, reports status and fails safe under reset or peripheral failure. The selected controller, an Arduino Nano Every based on the ATmega4809, offers enough capability for this role but not enough memory or processor margin to support undisciplined software architecture.

This manual establishes the baseline architecture for the control system and human-interface device (HID). It specifies the boundary between safety-critical control and convenience services, the scheduler model, memory ownership, state transitions, virtual I/O channels, panel behavior, persistent data and the compact profile interpreter.

- Included: firmware structure, logical timing, control-state authorization, panel controls, display behavior, software-visible I/O and memory budgets.
- Excluded: energy-stage component values, mechanical construction, electrode geometry, detailed charger design and operating procedures.
- Baseline controller: Arduino Nano Every / ATmega4809.
- Baseline panel: 128x64 monochrome I2C OLED, rotary encoder, Back control, physical Arm control, hard Abort/Inhibit, status indicators and sounder.

2. Design Premise

The architecture deliberately borrows the most relevant idea from the Apollo Guidance Computer: scarce resources are managed by dividing software into a small, fast native layer and a compact, more expressive service layer. The historical AGC used native instructions for direct machine work and an interpreter for denser, higher-level calculations. NanoArc applies the same philosophy without imitating AGC syntax or hardware.

DESIGN NOTE: Historical inspiration: the supplied Virtual AGC manual describes native AGC4 code, a software interpreter with its own registers and stack, memory banking, timer-driven services and explicit interrupt handling. NanoArc adopts the separation of critical and interpretive work, not the AGC instruction set itself [1].

2.1 Design axioms

1. The display is never part of the authorization or timing path.
2. Operator intent is a request; the control supervisor is the authority.
3. All outputs default inhibited at reset, brownout, watchdog expiry and impossible state.
4. Critical inputs use direct MCU or hardware paths; expanders are reserved for noncritical panel I/O.
5. The pulse window owns its timer and may suspend all lower-priority services.
6. Production firmware uses no heap and no unbounded data structure.
7. Every queue, buffer, profile and task table has a compile-time maximum size.
8. No restart may return directly to Armed, Firing or an unfinished profile action.
9. Profile code cannot directly write hardware registers or bypass limit checks.
10. Failure of any advisory service must degrade operation toward Safe, not toward permissive output.

3. System Partitioning

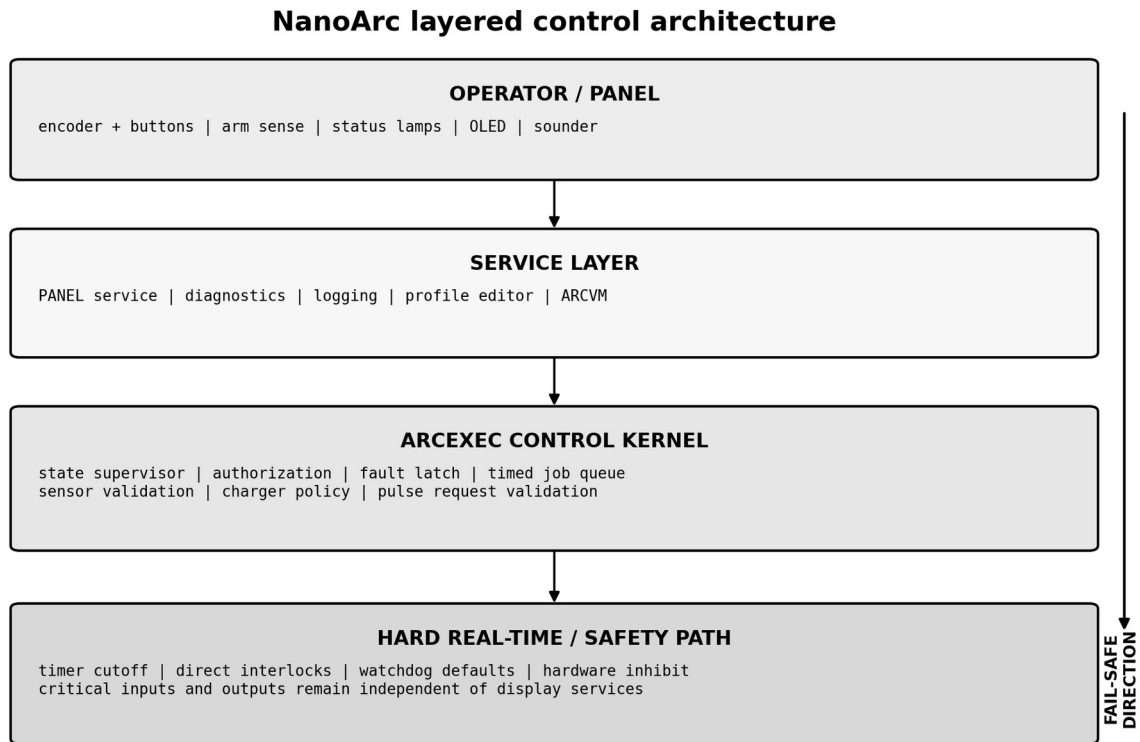


Figure 2. Four-layer partition. Authority increases toward the bottom; convenience increases toward the top.

3.1 Layer 0 - hard safety path

Layer 0 consists of hardware defaults and the smallest possible real-time firmware path. It owns output inhibit, timer cutoff, direct interlocks and watchdog-safe initialization. It must remain effective if the display, I2C bus, profile interpreter or main cooperative loop stalls.

3.2 Layer 1 - ARCEXEC control kernel

ARCEXEC is the authoritative state supervisor. It evaluates sensor validity, current machine state, interlock facts, limits and requests. Only ARCEXEC may authorize charging, arming or a timed output request. It also owns the fault latch and transitions to Safe or Fault.

3.3 Layer 2 - service layer

The service layer contains the panel service, diagnostic formatter, persistent-storage service, event logger and ARCVm profile interpreter. Services communicate with ARCEXEC through bounded requests and snapshots. They never access critical outputs directly.

3.4 Layer 3 - operator interface

The operator interface presents machine facts and accepts requests. Physical indicators carry the most important states. The OLED adds detail but is advisory while Armed. A frozen or disconnected display must not change the actual machine state.

4. Hardware Platform and Resource Envelope

Resource	Nominal capacity	Architecture consequence
Program flash	48 KB	Adequate for a small kernel, drivers, panel service and compact interpreter. Font and diagnostic growth must be controlled.

Resource	Nominal capacity	Architecture consequence
SRAM	6 KB	Primary constraint. Requires page-buffered graphics, fixed-size structures, no heap and a protected stack reserve.
EEPROM	256 bytes	Suitable for compact settings and calibration. High-frequency counters should eventually move to FRAM.
Clock	20 MHz class	More than adequate for supervisory control when blocking display and serial work are excluded from critical paths.
I2C	Shared two-wire bus	OLED, telemetry devices and panel expander may share the bus; bus failure is treated as a service-layer fault.
Hardware timers	Finite, explicitly assigned	One timer is reserved for hard cutoff; remaining timers are assigned by architecture rather than library accident.

The resource envelope is comparable in spirit, though not in exact capability, to other highly constrained embedded computers: the system is practical only when memory and timing are treated as architecture rather than afterthoughts.

4.1 Prohibited hidden costs

- Full-screen color framebuffers.
- Dynamic String objects in persistent or critical code.
- Libraries that claim hardware timers without an explicit assignment review.
- Long synchronous serial formatting in the control loop.
- Unbounded logging or history lists.
- Runtime recursion.
- Implicit floating-point use in the critical path.

5. Data Representation

NanoArc uses fixed-width integer types and explicit engineering-unit scaling. The objective is predictable storage, deterministic arithmetic and simple inspection in diagnostics.

Quantity	Representation	Example unit / rule
Machine state	uint8_t enum	One authoritative state value.
Status facts	uint16_t bit word	Interlocks, readiness, sensor validity and inhibit facts.
Fault code	uint8_t enum	Primary latched cause; supporting flags may coexist.
Voltage	uint16_t	Millivolts or another fixed scale selected per sensor range.
Current	int16_t or uint16_t	Scaled integer; sign only where physically meaningful.
Temperature	int16_t	Tenths of a degree C.
Charge / battery level	uint16_t	Permille, 0-1000, avoiding floating point.
Slow time	uint32_t	Monotonic milliseconds; subtraction used for rollover-safe tests.

Quantity	Representation	Example unit / rule
Fast timing	hardware timer count	Owned by the pulse timer; never derived from display timing.

5.1 Status word

```

bit 15  OUTPUT_INHIBITED
bit 14  FAULT_LATCHED
bit 13  ARM_REQUESTED
bit 12  TRIGGER_ACTIVE
bit 11  GRIP_INTERLOCK_OK
bit 10  ENERGY_READY
bit 9   SENSOR_SET_VALID
bit 8   TEMPERATURE_OK
bit 7   CHARGER_ACTIVE
bit 6   PANEL_AVAILABLE
bit 5   STORAGE_AVAILABLE
bit 4   PROFILE_VALID
bit 3   COOLDOWN_ACTIVE
bit 2   SERVICE_DUE
bit 1   RESERVED
bit 0   RESERVED

```

Status bits describe facts. They do not replace the state machine. For example, ARM_REQUESTED and ENERGY_READY may both be true while the authoritative state remains Ready-Safe because another condition prevents entry to Armed.

5.2 Telemetry snapshot

Services read a coherent telemetry snapshot rather than individual volatile variables. ARCEXEC publishes the snapshot after completing a control evaluation. This prevents the panel from displaying a mixture of old and new values.

```

TelemetrySnapshot
sequence      uint8
machine_state uint8
status_word   uint16
primary_fault uint8
active_profile uint8
energy_level  uint16
battery_level uint16
temperature   int16
shot_count    uint32

```

6. Planned Memory Map

Planned memory ownership

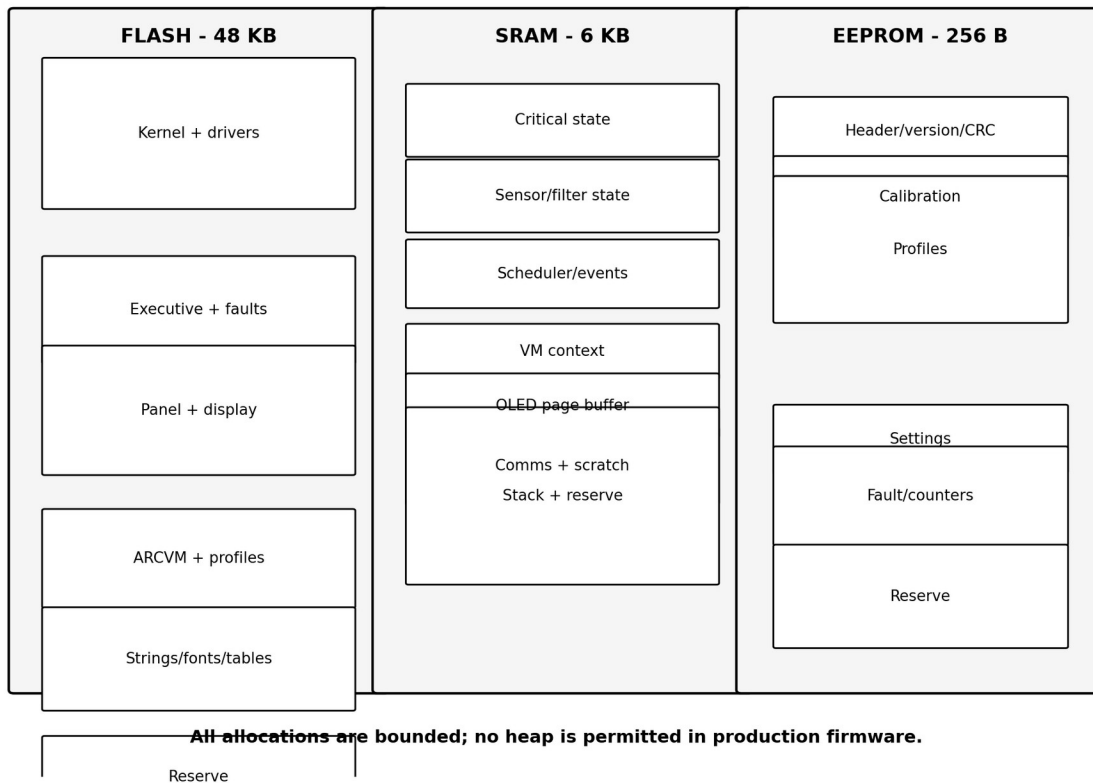


Figure 3. Memory ownership. Exact compiled sizes will replace the planning bands during implementation.

6.1 Preliminary SRAM budget

Allocation	Planning budget	Notes
Critical state and snapshots	256 B	State, flags, telemetry, timer ownership and fault context.
Sensor/filter state	256 B	Small fixed windows or IIR state only.
ARCEXEC and ARCLIST tables	256 B	Fixed task descriptors and deadlines.
Input event queue	64-128 B	Bounded queue; overflow is reported and noncritical events may coalesce.
ARCVM context and profile cache	128-256 B	One active profile; no nested interpreter instances.
OLED page buffer	128 B	Eight-pixel page for a 128-pixel-wide monochrome display.
I2C/UART/library buffers	256-512 B	Must be measured from the final linked image.
Formatting scratch	256 B maximum	No large printf buffers.
Stack and interrupt reserve	2048 B target	Protected headroom for nested calls and library behavior.
Future reserve	1024 B target	Architecture should not begin with zero growth margin.

DESIGN NOTE: The SRAM table is a planning allocation, not a promise of exact linker output. The release criterion is measured free stack margin under worst-case execution, not merely successful compilation.

6.2 Flash placement policy

- Constants, menu labels, icons and default profiles remain in program flash.
- Only the active profile and current display page need reside in SRAM.
- Diagnostic text uses identifiers and compact formatters rather than duplicating long messages.
- Optional development diagnostics are compiled out of production builds where practical.
- Font choice is an architecture decision because font tables may exceed the control logic in size.

7. ARCEXEC Executive

ARCEXEC is a small cooperative executive combined with an authoritative state supervisor. It is not a general-purpose RTOS. The objective is a completely bounded execution model that can be inspected in one source module and exercised with deterministic tests.

7.1 Executive cycle

```
loop:
  consume critical flags published by ISRs
  acquire a coherent input/sensor snapshot
  evaluate hard faults and inhibit conditions
  run one state-supervisor step
  release due ARCLIST jobs within the remaining budget
  idle or sleep until the next event
```

The state-supervisor step always runs before lower-priority scheduled jobs. A due display update may be skipped. A due safety evaluation may not.

7.2 ARCLIST timed-job queue

ARCLIST is the NanoArc equivalent of a small waitlist: a fixed array of timed service jobs. Each entry holds a function identifier, next release time, period, enable flag and priority class. The table is sorted or scanned without allocation.

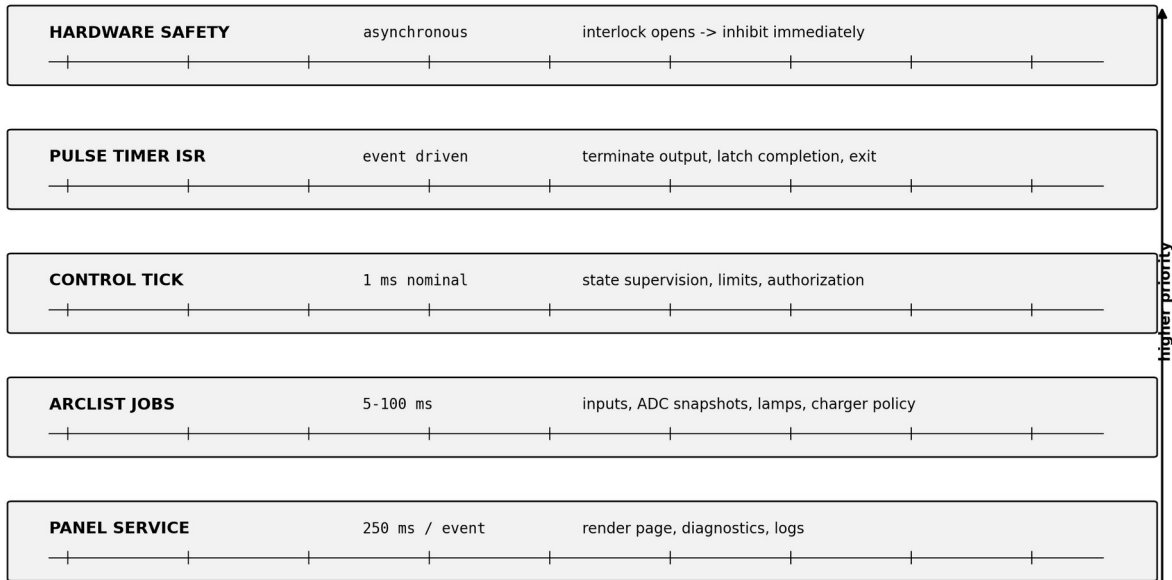
Job	Nominal release	Class	Miss policy
Read panel controls	5 ms	service-high	Run late; coalesce repeated edges where safe.
Acquire telemetry snapshot	10 ms	control-support	Run late; never block on display.
Evaluate charger policy	10 ms	control-support	Run before UI services.
Update physical indicators	50 ms	service-high	Run late; physical fault indication may also be driven immediately.
Check temperatures	100 ms	control-support	Run late within specified safety margin.
Refresh OLED	250 ms or event	service-low	Skip while critical window is active.
Commit settings	event / delayed	background	Cancel or postpone while Armed.

7.3 Priority rules

- A higher layer never waits for completion of a lower layer.
- Only one service job runs at a time; every job has a measured maximum execution time.
- Long I2C transactions are broken into pages or postponed.
- No service job may disable interrupts for an unbounded period.
- If the control budget is exhausted, remaining service jobs stay pending or are skipped according to policy.

8. Interrupt Processing and Timing Ownership

Timing ownership and execution lanes



Rule: lower lanes may be delayed or skipped; upper lanes may not wait on lower lanes.

Figure 4. Timing lanes. The panel is intentionally the lowest-priority periodic service.

8.1 Interrupt policy

11. Capture the event or perform the minimum immediate safety action.
12. Publish a volatile flag or bounded datum.
13. Clear the interrupt source.
14. Return immediately.
15. Perform interpretation, formatting and policy decisions in ARCEXEC, not in the ISR.

8.2 Timer ownership

Timer / source	Owner	Permitted work
Dedicated pulse cutoff timer	Layer 0	Output termination, completion latch and minimum bookkeeping.
Control tick source	ARCEXEC	Release supervisor step and high-priority service work.
Watchdog	Layer 0	Reset on missed executive service; output defaults must already be inhibited.
ADC completion, if interrupt-driven	Telemetry service	Store sample and mark channel complete.
Pin-change / external input	Input service	Capture edges for trigger, grip or encoder as assigned.

8.3 Critical-window behavior

When the system enters the brief critical window, ARCEXEC marks PANEL_SUSPENDED and rejects new noncritical bus work. The OLED remains on its last valid frame. Physical indicators and hard inhibit paths continue to operate.

on qualified request:
 suspend panel transactions
 arm dedicated timer
 issue validated hardware request
 await cutoff/completion flag
 enter Cooldown
 publish telemetry snapshot
 resume panel service

9. Authoritative Machine-State Model

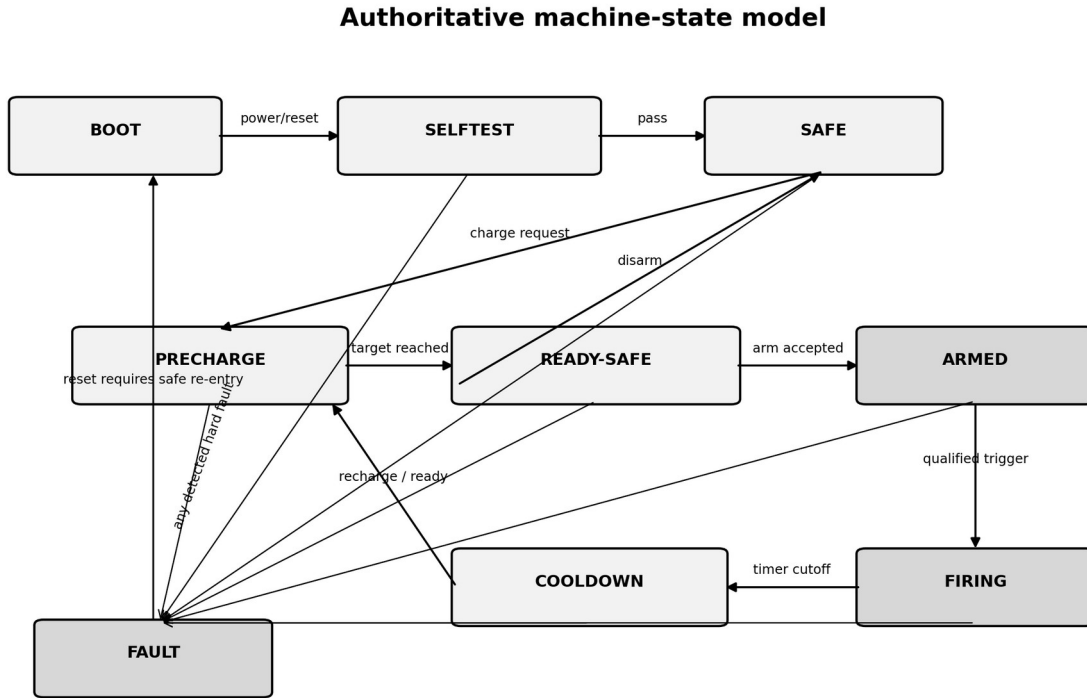


Figure 5. Baseline state transitions. Any hard fault may force Fault from any active state.

State	Outputs	Operator actions	Exit conditions
Boot	Inhibited	None	Minimum hardware initialized.
Selftest	Inhibited	Abort always honored	Sensors and stored-energy state validated.
Safe	Inhibited	Edit settings, select profile, request charge	Valid request and no fault.
Precharge	Firing inhibited; charger policy active	Cancel, abort, view status	Target reached, canceled or fault.
Ready-Safe	Firing inhibited	Arm request or return Safe	Arm accepted, energy falls or fault.
Armed	Conditional authorization active	Trigger request, disarm, abort	Qualified trigger, disarm, timeout or fault.
Firing	Timer-controlled action only	Abort path remains effective	Hard cutoff or fault.
Cooldown	Inhibited	Abort, view result	Cooldown satisfied; then Ready-Safe, Precharge or Safe.
Fault	Inhibited and latched	Acknowledge; perform required safe reset	Fault cleared through defined recovery path.

9.1 Transition authorization

Every state transition is implemented as a single explicit predicate. UI code may request a transition but cannot modify the state variable. Transition functions record both the previous state and a reason code for diagnostics.

```
authorize_arm =
    current_state == READY_SAFE
    AND no_fault_latched
    AND sensor_set_valid
    AND energy_within_window
    AND temperature_ok
    AND arm_control_asserted
    AND output_inhibit_chain_healthy
```

10. Virtual I/O Channels

To keep hardware details out of the control logic, NanoArc defines virtual I/O channels. A channel is a compact software-visible value with one owner. Drivers publish channel data; ARCEXEC reads facts and writes requests. The scheme is inspired by the separation between AGC memory and I/O channels but uses ordinary C++ structures.

Range	Class	Examples
0x00-0x0F	Input facts	Arm sense, grip, trigger, abort, connector detect.
0x10-0x1F	Output requests	Charger enable request, output inhibit request, buzzer pattern.
0x20-0x2F	Telemetry	Energy level, battery, temperature, current snapshot.
0x30-0x3F	Fault and health	Sensor validity, bus health, timeout, watchdog reason.
0x40-0x4F	Panel events	Encoder delta, Select, Back, menu timeout.
0x50-0x5F	Profile / service	Active profile, requested level, service mode flags.

10.1 Channel rules

- Each channel has exactly one writer and any number of readers.
- Critical channels are updated directly or atomically.
- Service-layer channels may be stale; their age is part of the data contract.
- No profile instruction receives a raw MCU pin number.
- Diagnostics may display channels but cannot change ownership.

11. ARCV M Compact Profile Interpreter

ARCV M provides compact, table-driven profiles without duplicating control code. It is deliberately non-Turing-complete: no arbitrary jump, recursion, raw memory access or raw I/O instruction is provided. A profile is a bounded linear sequence of declarative operations. ARCEXEC validates every request.

11.1 Interpreter context

Register / field	Size	Purpose
PC	uint8_t or uint16_t	Offset within the active profile.
PROFILE	uint8_t	Active profile identifier.
FLAGS	uint16_t	Interpreter condition and result flags.

Register / field	Size	Purpose
ARG0 / ARG1	uint16_t each	Decoded operands for the current operation.
WAIT_UNTIL	uint32_t	Release time for a bounded wait.
RESULT	uint8_t	Last ARCEXEC request result.

11.2 Baseline instruction set

Opcode	Operands	Meaning
END	-	Terminate profile execution and return control to the panel.
SET_TARGET	u16	Request a target level within kernel-defined limits.
SET_LIMITS	u8 profile class	Select a prevalidated limit set; cannot create new limits.
REQUIRE	u16 status mask	Pause or reject unless required facts are present.
WAIT_READY	u16 timeout	Wait for Ready-Safe; timeout produces a profile result, not a bypass.
REQUEST_ACTION	u8 sequence id	Ask ARCEXEC to execute a prevalidated action sequence.
SET_COOLDOWN	u16	Request a cooldown class or duration capped by the kernel.
UI_HINT	u8	Select an advisory prompt or icon; no control effect.
LOG_MARK	u8	Append a compact diagnostic marker if logging is available.

DESIGN NOTE: ARCVm does not contain an unrestricted FIRE opcode. REQUEST_ACTION identifies a kernel-defined sequence. ARCEXEC checks state, limits and interlocks at the moment of execution.

11.3 Encoding model

The baseline encoding uses a one-byte opcode followed by zero, one or two little-endian operands. Profiles are length-prefixed and include a version and checksum. Unknown opcodes invalidate the profile before it can run.

```

ProfileRecord
  magic      2 bytes
  version    1 byte
  length     1 byte
  profile_id 1 byte
  flags      1 byte
  bytecode   N bytes
  crc16      2 bytes

```

12. Human-Interface Architecture

Proposed control-panel organization

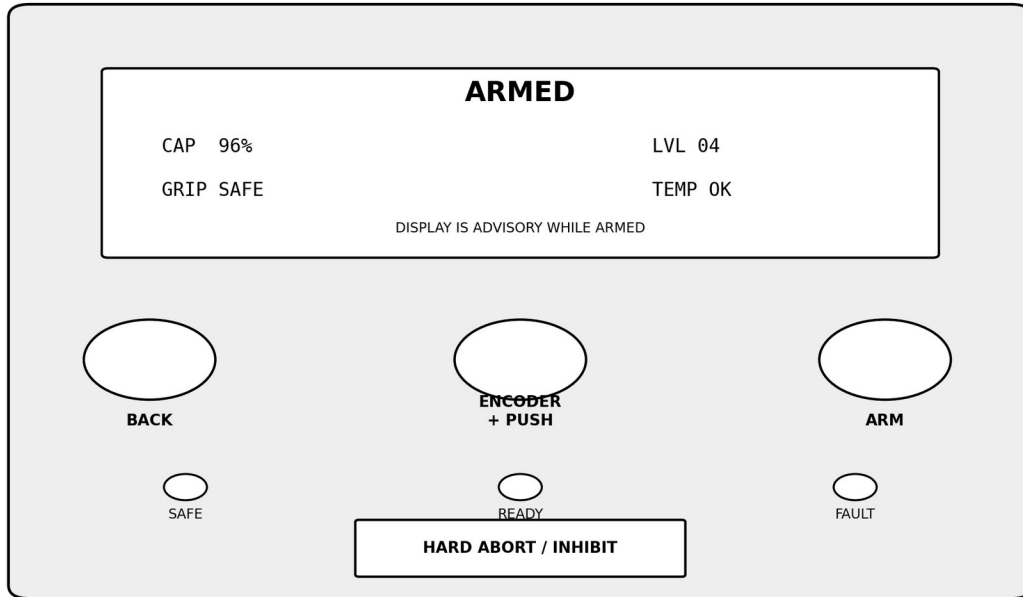


Figure 6. Proposed panel. The OLED explains state; physical controls and indicators carry critical meaning.

12.1 Controls

Control	Electrical treatment	Software role
Rotary encoder A/B	Direct MCU inputs	Navigation and bounded value changes.
Encoder push	Direct or expander input	Select / acknowledge noncritical prompts.
Back	Expander acceptable	Cancel edit, return one level, request Safe where permitted.
Arm control	Direct sense plus independent hardware effect where practical	Request entry to Armed; actual state is reported separately.
Grip interlock	Direct critical input	Required fact for action authorization.
Trigger	Direct critical input	Edge captured as a request only.
Abort / Inhibit	Hard path plus direct sense	Immediately removes authorization; software records event.

12.2 Indicators

Indicator	Meaning	Failure behavior
SAFE	Hardware inhibit is confirmed active; output authorization is absent.	Must remain meaningful with OLED unavailable.
CHARGE	Charger-enable feedback is active, not merely requested.	Loss or mismatch becomes a charger or telemetry alarm.
READY	Stored energy is inside the validated	Never inferred solely from elapsed

Indicator	Meaning	Failure behavior
	window for the selected profile.	charge time.
ARMED	Final ACE authorization is active after all interlocks are satisfied.	Must report actual state rather than Arm-switch position.
OUTPUT	A pulse command or verified gate-driver event occurred; indication may be held briefly.	Advisory after the immediate event; never used for timing.
FAULT	A program or machine alarm is latched.	Assert independently of OLED display ownership.
ENERGY	Stored energy remains above the defined service-safe threshold.	Independent threshold/comparator drive is preferred.
Sounder	Acknowledgement, warning and fault patterns.	Never the sole indication of a critical state.

12.3 Input grammar

- Turn: move focus or adjust a bounded value.
- Press: select or acknowledge.
- Back: cancel edit or move to the parent screen.
- Long press: reserved for noncritical shortcuts only; never used as a hidden safety override.
- Arm and Abort remain physical state controls, not menu commands.

12.4 Program / Verb / Noun command model

ACE adopts a DSKY-inspired command grammar without requiring a numeric keypad. PROGRAM identifies the active operating procedure, VERB identifies an action and NOUN identifies the controlled or displayed object. The encoder browses stable numeric codes and readable labels. All command sources - encoder, panel controls, internal programs and service serial - produce the same ArcCommand record and enter one validation path.

```

encoder / panel / internal / service
    -> ArcCommand {program, verb, noun, value}
    -> verb table + noun metadata
    -> supervisor authorization
    -> execute, reject, prompt, or alarm

```

13. Display Service

The baseline display is a 128x64 monochrome OLED on I2C. The renderer uses an eight-pixel page buffer of approximately 128 bytes rather than a full 1024-byte framebuffer. The display service is event-driven with a slow periodic refresh for values that drift.

13.1 Screen hierarchy

Screen	Primary content	Refresh policy
Safe / Setup	Profile, target level, battery, service status	Event driven; up to 8-10 Hz while actively editing.
Precharge	Progress, target, temperature, cancel prompt	2-5 Hz.
Ready-Safe	Readiness, profile, level, arm prompt	2-4 Hz or on state change.
Armed	Large state label, energy, level, grip fact	2-4 Hz; freezes during critical window.
Cooldown	Countdown / readiness facts	2-4 Hz.
Fault	Fault code, plain-language cause, reset condition	Immediate request, then 2 Hz.

Screen	Primary content	Refresh policy
Diagnostics	Raw channels, reset reason, build ID	Manual entry; unavailable while Armed.

13.2 Armed-screen contract

```

+-----+
|      ARMED      |
| CAP 96%   LVL 04 |
| GRIP SAFE  TEMP OK|
+-----+

```

While Armed, the display is tertiary. It shows only stable facts and never requires operator interaction for continued safety. If rendering stops, the physical Armed/Ready/Fault indications and hardware controls remain authoritative.

13.3 I2C failure handling

- Time-limit every bus transaction.
- Recover or reinitialize the bus only outside the critical window.
- Mark PANEL_AVAILABLE false after repeated failure.
- Continue safe supervisory control without the OLED.
- Prevent an I2C fault from extending a timed output action.

13.4 Noun component fields and contextual rendering

ACE does not reproduce the DSKY's three fixed numerical registers as permanent R1, R2 and R3 boxes. The OLED can present the same structured noun data more clearly. A noun may expose up to three logical components internally, but each component is rendered as a named, typed field with an engineering unit, range and access rule.

```

N10 POWER STATUS
  INPUT VOLTAGE  5.02 V
  BATTERY        82%
  TEMPERATURE    31.4 C

```

```

logical components: value[0..2]
display fields: label + type + unit + formatted value

```

The verb selects the rendering template. VIEW presents read-only values, SET exposes bounded editable fields, MONITOR emphasizes live values, PROGRAM prompts show procedure progress and ALARM pages replace ordinary content with recovery information. Program, Verb and Noun codes may remain in a compact header, while the body of the screen is contextual rather than register-shaped.

The three-component ceiling is a software bound, not a visual imitation. It keeps noun values fixed-size, permits generic validation and formatting, and avoids heap allocation. Nouns needing more information use another page, a subordinate noun or a program-specific screen rather than an unbounded record.

13.5 Display ownership classes

The Artemis display interface treats the display as an arbitrated resource rather than a global buffer that any routine may overwrite. ACE keeps one fixed request slot per class and one saved interrupted program display. Physical talkbacks do not participate in this arbitration; they report authoritative state even while the OLED is frozen, failed or owned by another class.

14. Input and Output Assignment Policy

The final pin map remains a hardware-integration decision, but the architectural allocation is fixed: critical signals are direct, shared buses are reserved and noncritical panel signals may use an expander.

Signal group	Preferred connection	Reason
Trigger, grip, arm sense, abort sense	Direct MCU / hardware path	Latency, observability and independence from I2C.

Signal group	Preferred connection	Reason
Pulse cutoff / output inhibit command	Dedicated timer-capable or direct output path	Explicit ownership and safe reset default.
Capacitor, battery, temperature telemetry	Direct analog or dedicated sensor interface	Control supervisor requires age and validity.
OLED and noncritical sensors	I2C	Pin efficiency; service may be suspended.
Back button, menu shortcuts, auxiliary lamps	MCP23017 or equivalent	Noncritical, low-rate I/O.
Encoder rotation	Direct inputs preferred	Avoid lost transitions and expander latency.
UART diagnostics	Reserved D0/D1 or board-defined serial path	Development and fault capture.
SPI	Reserved until memory/logging decision	Avoid accidental pin consumption by early prototypes.

14.1 Expansion connector

```
GND
LOGIC_SUPPLY
SDA
SCL
PANEL_INTERRUPT
UART_TX
UART_RX
SPARE_GPIO / SYNC
```

The expansion connector is a controlled interface, not an invitation for arbitrary peripheral growth. Every added device must state its bus time, SRAM cost, failure mode and timer requirements.

15. Fault, Alarm and Inhibit Architecture

Fault handling is layered. Hardware inhibit provides immediate protection, ARCEXEC latches and classifies the cause, and the panel explains recovery. Fault reporting never substitutes for output inhibition.

15.1 Fault classes

Class	Examples	Required response
F0 - hard immediate	Abort asserted, hard interlock opened, cutoff timeout	Inhibit immediately; latch Fault.
F1 - control invalid	Sensor set invalid, impossible state, timer ownership error	Inhibit; enter Fault.
F2 - operating limit	Temperature or stored-energy limit exceeded	Inhibit; Fault or Cooldown according to policy.
F3 - service degraded	OLED absent, FRAM absent, noncritical expander fault	Continue only if authoritative controls remain available; report degraded state.
F4 - maintenance	Calibration due, log full, optional sensor unavailable	Safe operation may continue under defined restrictions.

15.2 Fault record

```
FaultRecord
  code          uint8
  state_at_fault uint8
  status_word    uint16
  telemetry_seq  uint8
  reset_reason   uint8
  timestamp_low  uint16
  context        uint16
```

The record is intentionally compact. Detailed reconstruction comes from the state, status word, telemetry snapshot sequence and context field rather than a large text log.

15.3 Alarm presentation

- Physical Fault indicator asserts first.
- Sounder emits a bounded pattern unless muted by a hardware or policy condition.
- OLED displays the numeric code and a short plain-language cause.
- Diagnostics may expose raw context only after the machine is Safe.
- Acknowledging a message does not clear the physical cause.

16. Restart and Recovery

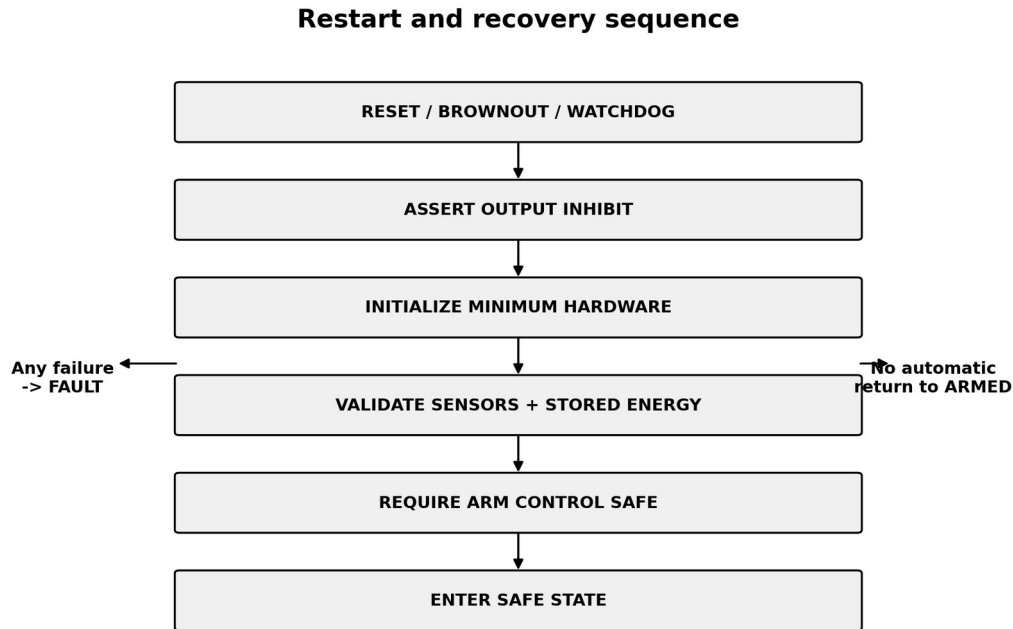


Figure 7. Every restart re-enters through inhibited hardware and Safe-state validation.

The reset path is designed around the assumption that the previous execution context is untrustworthy. No attempt is made to resume an interrupted action. Persistent settings may be restored only after outputs are inhibited and data integrity is verified.

16.1 Reset reasons

Code	Reason	Initial policy
00	Power-on	Normal self-test.
01	External reset	Normal self-test; record event.
02	Brownout	Extended supply and stored-energy validation.
03	Watchdog	Latch software fault until acknowledged or service policy permits reset.
04	Software requested	Normal self-test; reason retained.
05	Unknown / corrupted	Treat as watchdog-class fault.

16.2 Safe re-entry requirements

- Output inhibit asserted before peripheral initialization.
- Arm control observed in the non-armed position before Armed can be requested again.
- Trigger input must return inactive before a new edge is accepted.
- Sensor values must be plausible and fresh.
- Persistent profile and calibration records must pass version and checksum checks.
- Watchdog reset cannot be hidden by a normal-looking panel screen.

17. Persistent Storage

The built-in 256-byte EEPROM is sufficient for a compact baseline but not for high-frequency logging. Writes are transactional: a new record is written with version, length, sequence and checksum, then marked valid. Settings are committed on menu exit or after a delay, never on every encoder detent.

Region	Budget	Contents
Header	8 B	Magic, schema version, active slot and checksum.
Calibration	32 B	ADC scale/offset values and sensor identifiers.
Profiles	96 B	Several compact user profile records or overrides.
Global settings	16 B	Display timeout, sound policy and UI preferences.
Counters / last reset	16 B	Coarsely committed counters and reset reason.
Fault summary	40 B	Recent compact fault records.
Reserve	48 B	Migration and future fields.

17.1 FRAM expansion

A future I2C FRAM device is the preferred expansion for frequent counters and larger fault history. External RAM is not required for the baseline architecture because it would not transparently extend the AVR stack or ordinary variables. Native SRAM pressure is better controlled through page rendering and fixed structures.

18. Diagnostics and Testability

Diagnostics are designed into interfaces rather than added as print statements. Every transition, request and fault has a compact reason code. Host-side tests can drive virtual channels and verify state results without actual power hardware.

18.1 Build modes

Mode	Purpose	Differences
HOST_SIM	Desktop unit tests	Hardware drivers replaced with channel stubs; deterministic virtual time.
BENCH	Low-energy integration	Full panel and telemetry; critical outputs replaced or physically inhibited.
PRODUCTION	Final controller	Bounded diagnostics, watchdog enabled, test commands removed.
SERVICE	Controlled maintenance	Extra read-only diagnostics; entry requires physical Safe condition.

18.2 Required test families

- Every legal and illegal state transition.
- Every fault class from every active state.
- Queue overflow, stale telemetry and I2C timeout.
- Watchdog reset at each phase of the executive cycle.
- Profile validation, unknown opcode and truncated operand.
- Rollover-safe time comparisons.
- Minimum SRAM/stack margin under the deepest legal call path.
- Display failure while Safe, Precharge, Ready-Safe and Armed.

18.3 Diagnostic page

```
BUILD  A.0.1  RESET 02
STATE  READY-SAFE
STAT   C7A0
FAULT  00
VM     ID 03 PC 12 OK
I2C    OLED OK  EXP OK
STACK  MARGIN 1.9K
```

19. Coding and Verification Rules

Rule	Reason
No malloc, new or Arduino String in production firmware.	Prevents fragmentation and hidden failure paths.
Use enum class and fixed-width integer types.	Makes state and storage explicit.
Mark ISR-shared data volatile and access multi-byte values atomically.	Avoids torn reads on an 8-bit MCU.
No display, EEPROM, FRAM or serial formatting inside critical ISRs.	Bounds interrupt latency.
Every function documents maximum blocking time or is proven nonblocking.	Supports scheduler budgeting.
Use fixed-point arithmetic in critical supervision.	Avoids flash and timing surprises.
Drivers expose facts and requests, not global pin manipulation.	Preserves ownership.
Compile-time assertions validate table sizes and record layouts.	Prevents silent schema drift.
A build is rejected if stack reserve, flash reserve or timer ownership falls below policy.	Turns resources into release criteria.

19.1 Naming baseline

Name	Meaning
ARCEXEC	Authoritative executive and state supervisor.
ARCLIST	Fixed timed-job queue.
ARCVM	Compact noncritical profile interpreter.
PANEL	HID input and display service.
SAFECORE	Layer-0 initialization, inhibit and hard timer path.
CHANNELS	Virtual I/O channel ownership layer.

20. Implementation Sequence

16. Create the virtual channel types, machine-state enum and transition tests on the host.
17. Implement SAFECORE startup with all outputs inhibited and capture reset reason.
18. Implement ARCEXEC state supervision using stubbed channels.
19. Assign timer ownership and prove the cutoff ISR independently of the panel.
20. Add direct input drivers for Arm, Grip, Trigger and Abort.
21. Add telemetry snapshots and freshness/validity rules.
22. Implement ARCLIST with fixed jobs and execution-time instrumentation.
23. Add physical indicators and sounder patterns.
24. Add page-buffered OLED and panel navigation.
25. Implement ARCVN only after the native state machine is complete and tested.
26. Add EEPROM schema, checksums and migration policy.
27. Measure final flash, SRAM, stack margin, timer use and worst-case service latency before integrating optional peripherals.
28. Trace the Artemis 72 command, alarm, display and restart paths into explicit ACE interface contracts and host-side transition tests.

20.1 Exit criteria for Revision A firmware

- No unauthorized transition to Armed or Firing exists in exhaustive state tests.
- Panel disconnection cannot extend or initiate an action.
- Watchdog, brownout and manual reset always return through inhibited Safe entry.
- Worst-case stack reserve remains at or above the chosen release threshold.
- All critical timing remains within budget with display updates and diagnostics enabled.
- Every persistent record is versioned and checksum-protected.
- Pin and timer ownership tables match the compiled hardware configuration.

21. Artemis 72 Source-Derived Architecture

This chapter records the first direct architecture pass over the flown Artemis 72 / Colossus 3 symbolic listings, revision 072. The purpose is not to transliterate AGC assembly into C++, but to identify proven boundaries, bounded-resource mechanisms and restart rules that remain useful on the ATmega4809.

21.1 Source set and evidence

The primary evidence for this pass is EXECUTIVE, WAITLIST, KEYRUPT/UPRUPT, PINBALL GAME - BUTTONS AND LIGHTS, PINBALL NOUN TABLES, EXTENDED VERBS, DISPLAY INTERFACE ROUTINES, T4RUPT PROGRAM, ALARM AND ABORT, FRESH START AND RESTART, PHASE TABLE MAINTENANCE, RESTARTS ROUTINE and RESTART TABLES. Mission navigation and guidance mathematics were outside this pass.

The resulting ACE rules are treated as architecture requirements where they reinforce deterministic operation. Mechanisms that depended on AGC-specific banking, one's-complement arithmetic or hand-swapped execution contexts are not copied unless ACE has an equivalent need.

21.2 Hardware event to Executive job

Artemis event boundary adapted to ACE

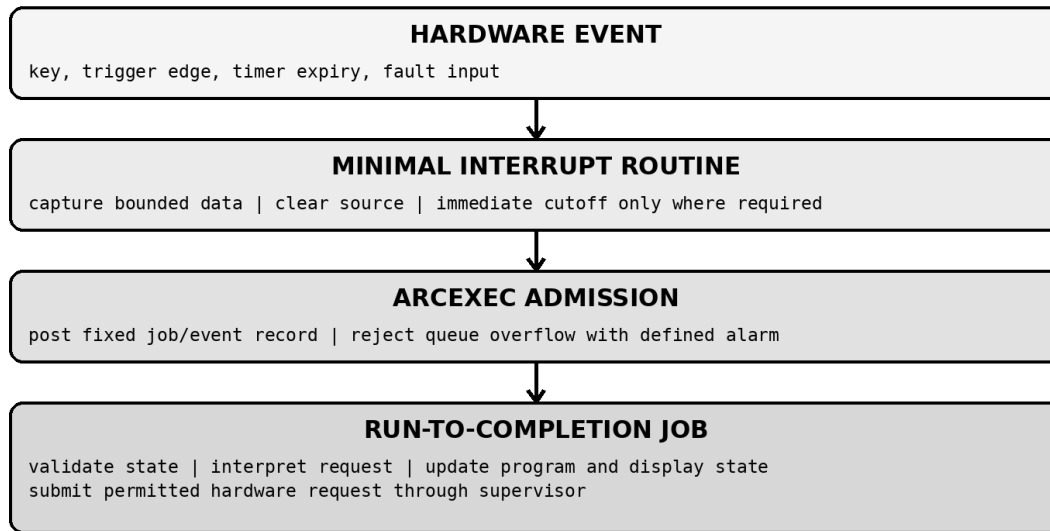


Figure 8. The flown interrupt boundary adapted to ACE: capture and admit first; interpret outside interrupt context.

The central discovery is a repeated boundary: a hardware interrupt captures a small datum, requests a bounded Executive job and returns. Policy, display changes and program flow occur in the admitted job. ACE formalizes this as an event-posting contract shared by trigger, panel, telemetry and service interrupts. Queue-full behavior is defined and alarmed; it is never allowed to corrupt the next record.

21.3 Executive and Waitlist adaptation

Artemis 72 mechanism	Observed bound	ACE adaptation
Executive core sets	Seven fixed job contexts	Eight fixed ArcJob records; explicit phases; no hidden stack suspension.
VAC areas	Five large workspaces	One bounded active program context and one ARCVM context.
NOVAC / FINDVAC	Admission class selected by workspace need	Job descriptor selects native handler or program/VM service.
Waitlist	Nine tasks, 10 ms resolution, 162.5 s maximum	Eight ArcTimedTask records with 32-bit absolute deadlines.
Resource exhaustion	Defined abort codes	Defined ACE program alarm and fail-safe queue policy.

ACE does not emulate the AGC context switch. A handler that cannot finish saves an explicit phase, returns and is later re-posted or released by ARCLIST. This preserves bounded memory and makes restart semantics inspectable.

21.4 Unified command path and noun metadata

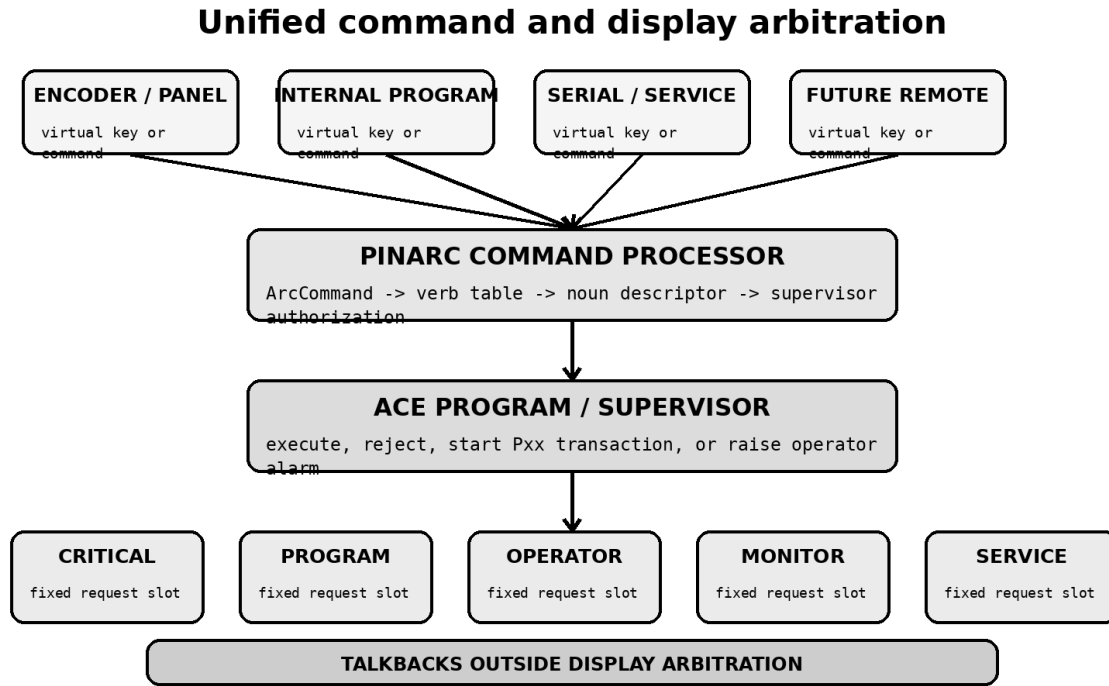


Figure 9. All command sources converge on one verb/noun validator; display classes and talkbacks remain separate.

PINBALL accepts keyboard, uplink and internal NVSUB requests through the same command machinery. ACE generalizes that rule: the physical encoder is only one producer of ArcCommand records. The command processor is intentionally independent of connector, keypad or serial framing, making future service tools unable to bypass ordinary authorization.

Noun descriptors are the data contract. They hold component count, display/load permissions, engineering format, bounds, persistence policy and read/write handlers. Verb descriptors state whether a noun is required and which machine states permit execution. Spare or illegal table entries always produce an operator alarm.

```

NounDescriptor
  code | flags | component_count | format
  minimum | maximum | read_handler | write_handler
  
```

```

VerbDescriptor
  code | flags | handler
  
```

Tables remain in flash; only selected codes and values occupy SRAM.

The AGC DSKY exposed noun components through three physical numeric registers. ACE retains the useful data-model principle but not the hardware constraint: a noun descriptor may define up to three typed components, each with its own label, unit, format, permissions and bounds. The OLED renders those components through the current verb or program template rather than as anonymous R1/R2/R3 fields.

```

NounComponentDescriptor
  label | value_type | unit | format | permissions
  minimum | maximum
  
```

```

NounDescriptor
  code | flags | component_count <= 3
  component[3] | read_handler | write_handler
  
```

21.5 Display arbitration and talkbacks

The flown display system distinguishes priority, extended-verb, normal, miscellaneous and astronaut-initiated displays, with explicit interruption rules. ACE maps those classes to CRITICAL, PROGRAM, OPERATOR, MONITOR and SERVICE. One request slot per class is sufficient; a priority page replaces or suspends lower classes without allocating a queue.

SAFE, CHARGE, READY, ARMED, OUTPUT, FAULT and ENERGY talkbacks report sensed or authoritative state rather than requested state. They remain visible even if the OLED transport is suspended during the critical window or permanently marked unavailable after an I2C failure.

21.6 Alarm and recovery model

Alarm response is split into four ordered actions: immediate inhibit where required, independent talkback annunciation, compact history/context capture and optional priority display. Acknowledging a page clears neither the physical cause nor the latched machine state.

ACE class	Meaning	Typical recovery
NOTICE	Recorded event with no inhibit.	Continue and expose through logs.
OPERATOR	Invalid verb/noun, value or sequence.	Reject request and remain in current program.
PROGRAM	Active Pxx procedure cannot continue.	Abort program and return to a safe parent state.
MACHINE	Machine condition invalid or outside limit.	Return to Safe and require acknowledgement/revalidation.
HARD	Cutoff, inhibit or hardware integrity failure.	Physical reset, power cycle or service investigation.

21.7 Restart reconstruction

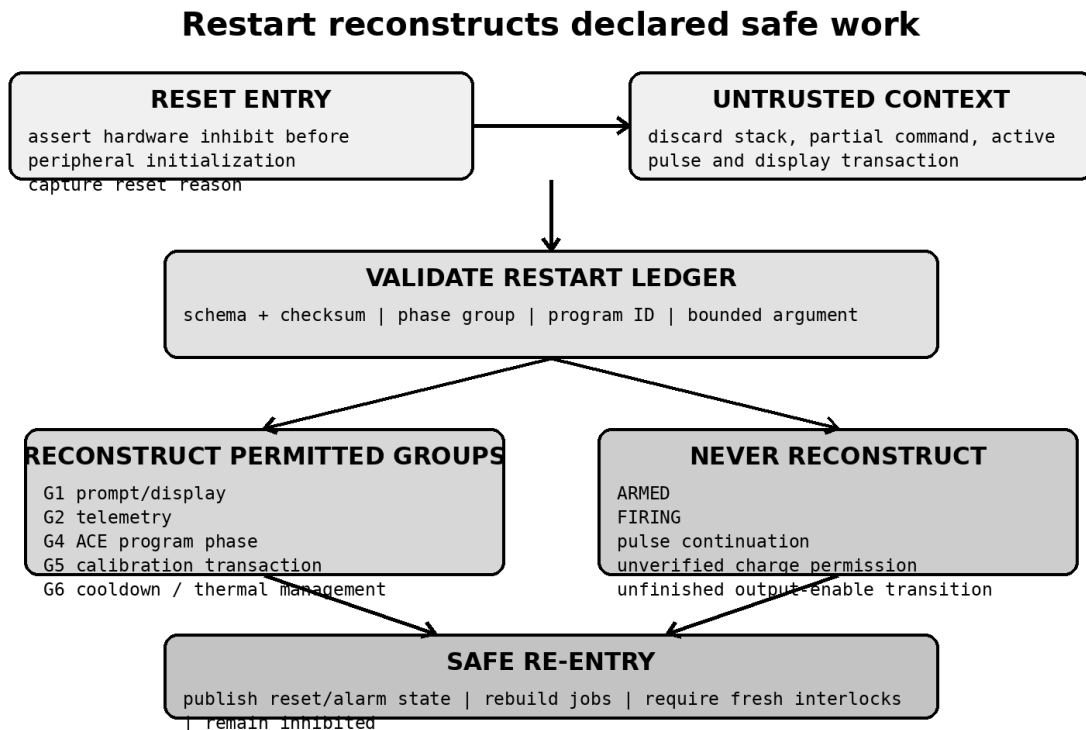


Figure 10. ACE restarts from an inhibited entry and reconstructs declared safe work rather than interrupted execution.

Artemis fresh-start and restart paths rebuild the Executive and Waitlist from restart groups and phase records. ACE uses the same reconstruction principle with stricter output rules. The persistent ledger records what safe work may be recreated, not CPU registers or a return address. Any invalid ledger is discarded and reported.

ARMED, FIRING, pulse continuation, unverified charging and an unfinished output-enable transition are never restartable. Calibration, telemetry, cooldown, logging and selected operator prompts may be reconstructed only after fresh interlock and sensor validation.

21.8 Program selection as a transaction

Artemis Verb 37 validates the requested major mode, rejects prohibited transitions, cleans or preserves selected work, records restart state and only then launches the new program. ACE therefore treats PROGRAM Pxx selection as a transaction: validate the program and hardware prerequisites; terminate or suspend the old program; establish a restart phase; initialize the new context; and publish the actual active program. The display never reports a requested program as active before the transaction commits.

```
REQUEST Pxx
-> validate program and present machine state
-> validate profile, sensors and required hardware
-> terminate or suspend current program
-> write restart phase
-> initialize new program context
-> publish ACTIVE Pxx or raise PROGRAM ALARM
```

21.9 Established source-derived requirements

29. Interrupts capture and acknowledge; ARCEXEC jobs interpret.
30. All queues and contexts are fixed-size and reject overflow explicitly.
31. Jobs yield only through explicit phases, never through hidden suspended stacks.
32. Timed tasks post jobs; the cutoff timer remains a dedicated hard-real-time path.
33. All command sources share one Program/Verb/Noun validation path.
34. Nouns define permissions, units, formats, bounds and persistence behavior.
35. Verb and program dispatch are table-driven; spare entries fail safely.
36. Display ownership is arbitrated and is never machine authority.
37. Alarm annunciation is independent of alarm-detail display.
38. Restarts reconstruct declared safe work and discard unsafe transient phases.
39. Program changes are validated transactions rather than direct jumps.
40. ARMED and FIRING are never restartable phases.
41. Noun components are fixed-size typed fields; ACE does not impose permanent R1/R2/R3 display registers.

21.10 Next analysis pass

The next source pass will trace exact end-to-end paths and convert them into ACE interfaces and testable transition tables: KEYRUPT1 to CHARIN and verb fan-out; NVSUB display sleep/wake; ALARM/POODOO/BAILOUT; PHASCHNG through restart reconstruction; Verb 37 major-mode transition; and DSPTAB through T4RUPT physical display servicing.

Appendix A. Summary Tables

A.1 Priority order

Priority	Owner	Work
0	Hardware / SAFECORE	Abort and physical inhibit.
1	Cutoff ISR	Terminate timed action.
2	ARCEXEC	Fault evaluation and state authorization.
3	Control support	Fresh telemetry and charger policy.
4	Panel indicators	Lamps and sounder.
5	OLED / input menus	Display and configuration.
6	Storage / logging	Deferred persistence.

A.2 Preliminary resource targets

Metric	Target
Flash use	Prefer ≤ 38 KB, leaving approximately 10 KB for growth and boot/library variation.
Static SRAM before stack	Prefer ≤ 4 KB.
Measured worst-case stack margin	At least 1.5 KB; 2 KB preferred.
OLED buffer	128-byte page buffer.
Input event queue	8-16 fixed entries.
ARCLIST jobs	No more than 12 in Revision A.
ARCVM profile length	Bounded; initial target 64 bytes per profile.
OLED refresh while Armed	2-4 Hz, suspended during critical window.

A.3 Preliminary logical pin budget

Function	Count	Policy
I2C SDA/SCL	2	Shared service bus.
Encoder A/B	2	Direct.
Trigger, grip, arm sense, abort/fault sense	4	Direct critical inputs.
Output inhibit / charger / timed request	2-3	Direct controlled outputs; exact hardware interface TBD.
Sounder	1	Direct or timer-capable as required.
Analog telemetry	3-5	Energy, battery, temperature and optional current.
UART	2	Reserved diagnostics.
SPI	3-4 shared	Reserved for later memory or logging decision.
Expander interrupt	1	Optional.

Appendix B. Preliminary Fault Catalogue

Code	Name	Class	Immediate action	Recovery
01	ABORT_ACTIVE	F0	Inhibit	Release physical abort; return through Safe.
02	INTERLOCK_OPEN	F0	Inhibit	Restore interlock; explicit re-arm.
03	CUTOFF_TIMEOUT	F0	Inhibit and latch	Power-cycle/service investigation.
10	SENSOR_INVALID	F1	Inhibit	Validate wiring and fresh samples.
11	IMPOSSIBLE_STATE	F1	Inhibit and watchdog-safe	Software fault investigation.

Code	Name	Class	Immediate action	Recovery
			reset	
12	TIMER_CONFLICT	F1	Inhibit	Build/configuration correction.
20	OVER_TEMPERATURE	F2	Inhibit	Cooldown and explicit acknowledgement.
21	ENERGY_OUT_OF_RANGE	F2	Inhibit charger/action	Return within validated range.
30	PANEL_BUS_FAILED	F3	Keep output inhibited if required controls unavailable	Safe-mode operation or repair.
31	DISPLAY_MISSING	F3	Advisory loss	Continue only under panel policy.
32	PERSISTENCE_FAILED	F3	Use safe defaults	Repair or replace storage.
40	CALIBRATION_DUE	F4	Restrict operation by policy	Service calibration.

Appendix C. Open Decisions

Decision	Current baseline	Closure evidence
OLED controller and physical size	128x64 monochrome I2C; physically large module preferred	Readability test in enclosure.
Full framebuffer vs page buffer	Page buffer	Compiled SRAM and render latency.
MCP23017 inclusion in Revision A	Optional panel daughterboard	Final direct-pin budget.
FRAM inclusion	Deferred	EEPROM wear model and logging requirements.
Exact hardware timer assignment	Reserved by role, numbers TBD	Board core and library timer audit.
Encoder polling vs interrupt capture	Direct inputs; method TBD	Measured detent rate and missed-edge tests.
Profile record count	Several compact profiles	Actual EEPROM/FRAM selection.
Armed indicator color/label	Physical dedicated indicator required	Human-factors panel prototype.

References

- [1] Virtual AGC Assembly-Language Manual, Block II / AGC4 and Interpreter reference, supplied project document. Used as historical architecture inspiration for native-versus-interpreted execution, interrupts, timers, memory discipline and I/O separation.
- [2] Microchip ATmega4808/4809 Data Sheet. Device memory, timers, interrupt and peripheral reference.
- [3] Arduino Nano Every technical documentation and board pinout.
- [4] Microchip MCP23017 Data Sheet. Optional I2C panel I/O expansion.
- [5] Project design discussions: NanoArc power supply, pen/electrode design and control-panel architecture, June 2026.
- [6] Artemis 72 / Colossus 3 symbolic listings, revision 072: Executive, Waitlist, PINBALL, display interface, alarm/abort and restart sources.
- [7] NanoArc ACE - Artemis 72 Architecture Analysis, Pass 1, 2026-06-22.