

Power, avionics & flight software

Keep energy, data and control available together

Starting question — Why can a small electrical or software fault become a mission-level failure?

Intuition. Power, sensors, networks and software form one operational chain. Budgets, isolation, deterministic behaviour and safe recovery must be designed together rather than checked independently.

- Explain the governing idea before calculating.
- Name the units, evidence source and operational boundary of every key quantity.
- Check the result with an inverse, bound, refinement or independent physical estimate.

A spacecraft does not have an ideal wall socket. Generation varies, batteries age, converters dissipate heat and some loads cannot operate simultaneously. Avionics must measure that state, execute flight software, command actuators and remain controllable when a sensor or computer fails. This module therefore connects electrical power, electronics and critical software.

Before you start — Prerequisites: modules 01 to 09 recommended. Every important symbol is defined again at first use.

Mastery objectives

- explain concepts with units and assumptions
- redo a simple calculation by hand before using a tool
- identify at least one failure mode or model limitation
- connect the discipline to a complete Mars architecture

Zero-prerequisite concepts

These four concept cards establish the vocabulary needed before the calculations and mission decisions in this module.

electrical power

Definition. Electrical power is the rate at which electrical energy is delivered or consumed.

Example. A 120 W computer and two 80 W instruments draw 280 W while all three operate simultaneously.

Pitfall. Power and energy are different: a short high-power pulse can use less energy than a long low-power load.

Mental check. Use the mission timeline to turn power states into an energy budget.

Guided exercise — electrical power

In a Mars mission scenario, identify one situation in which “electrical power” changes an engineering or operational decision. State the evidence you would inspect, the mistake you must avoid, and one independent check you would perform before accepting the decision.

Detailed correction — electrical power

Core meaning. Electrical power is the rate at which electrical energy is delivered or consumed.

Mission example. A 120 W computer and two 80 W instruments draw 280 W while all three operate simultaneously.

Error to reject. Power and energy are different: a short high-power pulse can use less energy than a long low-power load.

Independent check. Use the mission timeline to turn power states into an energy budget.

Quantification

Use the physical unit that belongs to electrical power when it is quantitative; if it is qualitative, do not invent a numerical unit.

Verification

Compare the conclusion with the mission example, the stated pitfall and the mental check before using electrical power operationally.

electrical bus

Definition. An electrical bus is a shared distribution path that delivers power and defines voltage, protection and fault boundaries.

Example. A spacecraft may isolate a failed branch while keeping essential loads on a protected bus.

Pitfall. Redundant equipment on one unprotected bus can still share a single-point failure.

Mental check. Trace source, conversion, breaker or switch, wiring and return path for each critical load.

Guided exercise — electrical bus

In a Mars mission scenario, identify one situation in which “electrical bus” changes an engineering or operational decision. State the evidence you would inspect, the mistake you must avoid, and one independent check you would perform before accepting the decision.

Detailed correction — electrical bus

Core meaning. An electrical bus is a shared distribution path that delivers power and defines voltage, protection and fault boundaries.

Mission example. A spacecraft may isolate a failed branch while keeping essential loads on a protected bus.

Error to reject. Redundant equipment on one unprotected bus can still share a single-point failure.

Independent check. Trace source, conversion, breaker or switch, wiring and return path for each critical load.

Quantification

Use the physical unit that belongs to electrical bus when it is quantitative; if it is qualitative, do not invent a numerical unit.

Verification

Compare the conclusion with the mission example, the stated pitfall and the mental check before using electrical bus operationally.

avionics

Definition. Avionics includes onboard sensing, data acquisition, computing, networking and control electronics.

Example. An inertial sensor can feed navigation software, which sends attitude commands through the data network to actuators.

Pitfall. Avionics is not only the flight computer; interfaces and timing are part of the chain.

Mental check. Follow one critical measurement from sensor to decision to actuator and identify where it can be lost or corrupted.

Guided exercise — avionics

In a Mars mission scenario, identify one situation in which “avionics” changes an engineering or operational decision. State the evidence you would inspect, the mistake you must avoid, and one independent check you would perform before accepting the decision.

Detailed correction — avionics

Core meaning. Avionics includes onboard sensing, data acquisition, computing, networking and control electronics.

Mission example. An inertial sensor can feed navigation software, which sends attitude commands through the data network to actuators.

Error to reject. Avionics is not only the flight computer; interfaces and timing are part of the chain.

Independent check. Follow one critical measurement from sensor to decision to actuator and identify where it can be lost or corrupted.

Quantification

Use the physical unit that belongs to avionics when it is quantitative; if it is qualitative, do not invent a numerical unit.

Verification

Compare the conclusion with the mission example, the stated pitfall and the mental check before using avionics operationally.

flight software

Definition. Flight software implements mission logic, timing, state management, monitoring and recovery on the onboard computers.

Example. After a communications loss, software may enter safe mode, reduce loads and wait for recovery commands.

Pitfall. More automation is not automatically safer if states, priorities and recovery paths are ambiguous.

Mental check. For each automated transition, identify trigger, allowed state, timeout, evidence of success and fallback behaviour.

Guided exercise — flight software

In a Mars mission scenario, identify one situation in which “flight software” changes an engineering or operational decision. State the evidence you would inspect, the mistake you must avoid, and one independent check you would perform before accepting the decision.

Detailed correction — flight software

Core meaning. Flight software implements mission logic, timing, state management, monitoring and recovery on the onboard computers.

Mission example. After a communications loss, software may enter safe mode, reduce loads and wait for recovery commands.

Error to reject. More automation is not automatically safer if states, priorities and recovery paths are ambiguous.

Independent check. For each automated transition, identify trigger, allowed state, timeout, evidence of success and fallback behaviour.

Quantification

Use the physical unit that belongs to flight software when it is quantitative; if it is qualitative, do not invent a numerical unit.

Verification

Compare the conclusion with the mission example, the stated pitfall and the mental check before using flight software operationally.

Calculation laboratory — formula, units, inverse check and limits

Each relation is taught from zero prerequisites: what it asks, what every symbol means, why the operation is valid, how to calculate it and how to catch a wrong result independently.

Canonical quantitative mini-lessons

This block is the module quantitative reference; exercise values are pedagogical and must not be confused with mission data.

Electrical power on a DC bus

$$P = U \times I$$

1 — Concrete question

What does “ $P = U \times I$ ” compute in “Electrical power on a DC bus”?

2 — Intuition without symbols

Instantaneous electrical power is voltage times current on a DC bus.

3 — Quantities

P: power; U: voltage; I: current

4 — Formula

$$P = U \times I$$

5 — Read aloud

Read “ $P = U \times I$ ” by naming every operation, subscript and grouping explicitly.

6 — Symbols and meaning

P: power; U: voltage; I: current

7 — Pronunciation

Say subscripts, powers, Greek letters and parentheses explicitly so the relation can be reconstructed without seeing the page.

8 — Units

P in W; U in V; I in A

9 — Convention

Keep the same frame, time basis, system boundary and sign convention throughout the substitution.

10 — Why this operation

Multiplication converts a rate or specific quantity into a total over the chosen interval or population.

11 — Assumptions

Factors must describe the same scenario and their units must be compatible.

12 — Unit check

P in W; U in V; I in A Verify that dimensional reduction reaches the unit of the requested output.

13 — Numerical case

With $U=28\text{ V}$ and $I=10\text{ A}$, $P=280\text{ W}$.

14 — Why the calculation works

Multiplication converts a rate or specific quantity into a total over the chosen interval or population.

15 — Independent check

Dividing the result by any non-zero factor must recover the other factor.

16 — Mental estimate

Estimating each factor separately predicts the order of magnitude of the product.

17 — Interpretation

The instantaneous result does not yet describe energy over time or conversion losses.

18 — What the result does not prove

The calculation proves neither that the inputs are correct nor that the model remains valid outside the declared scenario; those questions require separate evidence.

19 — Sensitivity

With independent factors, a relative change in one factor transfers directly to the product.

20 — Guided and autonomous exercises

Guided exercise. $U=24\text{ V}$, $I=5\text{ A}$.

Detailed guided correction — open after trying

$U=24\text{ V}$, $I=5\text{ A}$. $P=120\text{ W}$.

Autonomous exercise. $U=50\text{ V}$, $I=8\text{ A}$.

Autonomous correction — open after trying

$U=50\text{ V}$, $I=8\text{ A}$. $P=400\text{ W}$.

21 — Mission decision

Size wiring and converters using current, power, transients and appropriate margins.

Energy consumed over time

$$E = P \times t$$

1 — Concrete question

What does “ $E = P \times t$ ” compute in “Energy consumed over time”?

2 — Intuition without symbols

Power becomes an amount of energy when integrated over operating time.

3 — Quantities

E : energy; P : average power; t : duration

4 — Formula

$$E = P \times t$$

5 — Read aloud

Read “ $E = P \times t$ ” by naming every operation, subscript and grouping explicitly.

6 — Symbols and meaning

E : energy; P : average power; t : duration

7 — Pronunciation

Say subscripts, powers, Greek letters and parentheses explicitly so the relation can be reconstructed without seeing the page.

8 — Units

E in Wh when P is W and t is h; or J when t is s

9 — Convention

Keep the same frame, time basis, system boundary and sign convention throughout the substitution.

10 — Why this operation

Multiplication converts a rate or specific quantity into a total over the chosen interval or population.

11 — Assumptions

Factors must describe the same scenario and their units must be compatible.

12 — Unit check

E in Wh when P is W and t is h; or J when t is s. Verify that dimensional reduction reaches the unit of the requested output.

13 — Numerical case

With $P=420\text{ W}$ and $t=3\text{ h}$, $E=1,260\text{ Wh}=1.26\text{ kWh}$.

14 — Why the calculation works

Multiplication converts a rate or specific quantity into a total over the chosen interval or population.

15 — Independent check

Dividing the result by any non-zero factor must recover the other factor.

16 — Mental estimate

Estimating each factor separately predicts the order of magnitude of the product.

17 — Interpretation

The real profile must be integrated when power varies strongly instead of using an arbitrary average.

18 — What the result does not prove

The calculation proves neither that the inputs are correct nor that the model remains valid outside the declared scenario; those questions require separate evidence.

19 — Sensitivity

With independent factors, a relative change in one factor transfers directly to the product.

20 — Guided and autonomous exercises

Guided exercise. $P=800\text{ W}$, $t=2.5\text{ h}$.

Detailed guided correction — open after trying

$P=800\text{ W}$, $t=2.5\text{ h}$. $E=2,000\text{ Wh}=2.0\text{ kWh}$.

Autonomous exercise. $P=150\text{ W}$, $t=8\text{ h}$.

Autonomous correction — open after trying

$P=150\text{ W}$, $t=8\text{ h}$. $E=1,200\text{ Wh}=1.2\text{ kWh}$.

21 — Mission decision

Compare mission energy with energy actually usable after losses and reserves.

Usable battery energy

$E_{\text{usable}} = V \times C_{\text{Ah}} \times \text{DoD} \times \eta$

1 — Concrete question

What does “ $E_{\text{usable}} = V \times C_{\text{Ah}} \times \text{DoD} \times \eta$ ” compute in “Usable battery energy”?

2 — Intuition without symbols

Nominal capacity must not be confused with energy actually accessible after reserve and losses.

3 — Quantities

E_{usable} : usable energy; V : average voltage; C_{Ah} : nominal capacity; DoD: allowed depth of discharge; η : overall efficiency

4 — Formula

$E_{\text{usable}} = V \times C_{\text{Ah}} \times \text{DoD} \times \eta$

5 — Read aloud

Read “ $E_{\text{usable}} = V \times C_{\text{Ah}} \times \text{DoD} \times \eta$ ” by naming every operation, subscript and grouping explicitly.

6 — Symbols and meaning

E_{usable} : usable energy; V : average voltage; C_{Ah} : nominal capacity; DoD: allowed depth of discharge; η : overall efficiency

7 — Pronunciation

Say subscripts, powers, Greek letters and parentheses explicitly so the relation can be reconstructed without seeing the page.

8 — Units

E_{usable} in Wh; V in V; C_{Ah} in Ah; DoD and η dimensionless

9 — Convention

Keep the same frame, time basis, system boundary and sign convention throughout the substitution.

10 — Why this operation

Multiplication converts a rate or specific quantity into a total over the chosen interval or population.

11 — Assumptions

Factors must describe the same scenario and their units must be compatible.

12 — Unit check

E_{usable} in Wh; V in V; C_{Ah} in Ah; DoD and η dimensionless Verify that dimensional reduction reaches the unit of the requested output.

13 — Numerical case

With $V=28\text{ V}$, $C_{\text{Ah}}=100\text{ Ah}$, $\text{DoD}=0.70$ and $\eta=0.90$, $E_{\text{usable}}=1,764\text{ Wh}$.

14 — Why the calculation works

Multiplication converts a rate or specific quantity into a total over the chosen interval or population.

15 — Independent check

Dividing the result by any non-zero factor must recover the other factor.

16 — Mental estimate

Estimating each factor separately predicts the order of magnitude of the product.

17 — Interpretation

Temperature, ageing and discharge rate can reduce available energy further.

18 — What the result does not prove

The calculation proves neither that the inputs are correct nor that the model remains valid outside the declared scenario; those questions require separate evidence.

19 — Sensitivity

With independent factors, a relative change in one factor transfers directly to the product.

20 — Guided and autonomous exercises

Guided exercise. $V=26\text{ V}$, $C_{\text{Ah}}=90\text{ Ah}$, $\text{DoD}=0.70$, $\eta=0.95$.

Detailed guided correction — open after trying

$V=26\text{ V}$, $C_{\text{Ah}}=90\text{ Ah}$, $\text{DoD}=0.70$, $\eta=0.95$. $E_{\text{usable}}\approx 1,556\text{ Wh}$.

Autonomous exercise. $V=48\text{ V}$, $C_{\text{Ah}}=50\text{ Ah}$, $\text{DoD}=0.80$, $\eta=0.90$.

Autonomous correction — open after trying

$V=48\text{ V}$, $C_{\text{Ah}}=50\text{ Ah}$, $\text{DoD}=0.80$, $\eta=0.90$. $E_{\text{usable}}=1,728\text{ Wh}$.

21 — Mission decision

Size against end-of-life capacity and mission thermal conditions.

Captured solar power

$P_{\text{solar}} = G \times A \times \eta \times \cos_{\text{theta}}$

1 — Concrete question

What does " $P_{\text{solar}} = G \times A \times \eta \times \cos_{\text{theta}}$ " compute in "Captured solar power"?

2 — Intuition without symbols

Generation depends on incident flux, area, efficiency and panel orientation.

3 — Quantities

P_{solar} : electrical power produced; G : irradiance; A : active area; η : efficiency; $\cos_{\text{theta}}$: incidence factor

4 — Formula

$P_{\text{solar}} = G \times A \times \eta \times \cos_{\text{theta}}$

5 — Read aloud

Read " $P_{\text{solar}} = G \times A \times \eta \times \cos_{\text{theta}}$ " by naming every operation, subscript and grouping explicitly.

6 — Symbols and meaning

P_{solar} : electrical power produced; G : irradiance; A : active area; η : efficiency; $\cos_{\text{theta}}$: incidence factor

7 — Pronunciation

Say subscripts, powers, Greek letters and parentheses explicitly so the relation can be reconstructed without seeing the page.

8 — Units

P_{solar} in W; G in W/m^2 ; A in m^2 ; η and $\cos_{\text{theta}}$ dimensionless

9 — Convention

Keep the same frame, time basis, system boundary and sign convention throughout the substitution.

10 — Why this operation

Multiplication converts a rate or specific quantity into a total over the chosen interval or population.

11 — Assumptions

Factors must describe the same scenario and their units must be compatible.

12 — Unit check

P_{solar} in W; G in W/m^2 ; A in m^2 ; η and $\cos_{\text{theta}}$ dimensionless. Verify that dimensional reduction reaches the unit of the requested output.

13 — Numerical case

With $G=590\text{ W}/\text{m}^2$, $A=20\text{ m}^2$, $\eta=0.30$ and $\cos_{\text{theta}}=0.8$, $P_{\text{solar}}=2,832\text{ W}$.

14 — Why the calculation works

Multiplication converts a rate or specific quantity into a total over the chosen interval or population.

15 — Independent check

Dividing the result by any non-zero factor must recover the other factor.

16 — Mental estimate

Estimating each factor separately predicts the order of magnitude of the product.

17 — Interpretation

Dust, temperature, occultations and ageing must be added to the mission model.

18 — What the result does not prove

The calculation proves neither that the inputs are correct nor that the model remains valid outside the declared scenario; those questions require separate evidence.

19 — Sensitivity

With independent factors, a relative change in one factor transfers directly to the product.

20 — Guided and autonomous exercises

Guided exercise. $G=600$, $A=10$, $\eta=0.28$, $\cos_{\text{theta}}=1$.

Detailed guided correction — open after trying

G=600, A=10, eta=0.28, cos_theta=1. P_solar=1,680 W.

Autonomous exercise. G=500, A=15, eta=0.25, cos_theta=0.6.

Autonomous correction — open after trying

G=500, A=15, eta=0.25, cos_theta=0.6. P_solar=1,125 W.

21 — Mission decision

Keep qualified minimum generation above critical load with margin.

Current demanded from a bus

$I_{\text{bus}} = P_{\text{load}} / V_{\text{bus}}$

1 — Concrete question

What does " $I_{\text{bus}} = P_{\text{load}} / V_{\text{bus}}$ " compute in "Current demanded from a bus"?

2 — Intuition without symbols

For a given power, higher bus voltage reduces required current and some resistive losses.

3 — Quantities

I_{bus} : bus current; P_{load} : load power; V_{bus} : bus voltage

4 — Formula

$I_{\text{bus}} = P_{\text{load}} / V_{\text{bus}}$

5 — Read aloud

Read " $I_{\text{bus}} = P_{\text{load}} / V_{\text{bus}}$ " by naming every operation, subscript and grouping explicitly.

6 — Symbols and meaning

I_{bus} : bus current; P_{load} : load power; V_{bus} : bus voltage

7 — Pronunciation

Say subscripts, powers, Greek letters and parentheses explicitly so the relation can be reconstructed without seeing the page.

8 — Units

I_{bus} in A; P_{load} in W; V_{bus} in V

9 — Convention

Keep the same frame, time basis, system boundary and sign convention throughout the substitution.

10 — Why this operation

Division compares one quantity with a homogeneous reference and produces a proportion indicator.

11 — Assumptions

Both quantities must belong to the same scope and time period.

12 — Unit check

I_{bus} in A; P_{load} in W; V_{bus} in V Verify that dimensional reduction reaches the unit of the requested output.

13 — Numerical case

With $P_{\text{load}}=2,800$ W and $V_{\text{bus}}=28$ V, $I_{\text{bus}}=100$ A.

14 — Why the calculation works

Division compares one quantity with a homogeneous reference and produces a proportion indicator.

15 — Independent check

Multiplying the result by the denominator must recover the numerator.

16 — Mental estimate

Before calculating, compare numerator and denominator mentally to predict whether the result should be below, near or above one.

17 — Interpretation

The ideal relation does not replace analysis of losses, transients and converter limits.

18 — What the result does not prove

The calculation proves neither that the inputs are correct nor that the model remains valid outside the declared scenario; those questions require separate evidence.

19 — Sensitivity

The result rises with the numerator and falls with the denominator, all else equal.

20 — Guided and autonomous exercises

Guided exercise. $P_{\text{load}}=1,200$ W, $V_{\text{bus}}=24$ V.

Detailed guided correction — open after trying

$P_{\text{load}}=1,200$ W, $V_{\text{bus}}=24$ V. $I_{\text{bus}}=50$ A.

Autonomous exercise. $P_{\text{load}}=4,800 \text{ W}$, $V_{\text{bus}}=48 \text{ V}$.

Autonomous correction — open after trying

$P_{\text{load}}=4,800 \text{ W}$, $V_{\text{bus}}=48 \text{ V}$. $I_{\text{bus}}=100 \text{ A}$.

21 — Mission decision

Verify protections, wiring and converters tolerate the maximum credible current.

CPU utilisation of periodic tasks

$$U_{\text{cpu}} = C1/T1 + C2/T2 + C3/T3$$

1 — Concrete question

What does " $U_{\text{cpu}} = C1/T1 + C2/T2 + C3/T3$ " compute in "CPU utilisation of periodic tasks"?

2 — Intuition without symbols

Each periodic task consumes a fraction of the processor; these fractions add to estimate timing load.

3 — Quantities

U_{cpu} : demanded CPU fraction; $C1$: execution time of task one; $T1$: period of task one; $C2$: execution time of task two; $T2$: period of task two; $C3$: execution time of task three; $T3$: period of task three

4 — Formula

$$U_{\text{cpu}} = C1/T1 + C2/T2 + C3/T3$$

5 — Read aloud

Read " $U_{\text{cpu}} = C1/T1 + C2/T2 + C3/T3$ " by naming every operation, subscript and grouping explicitly.

6 — Symbols and meaning

U_{cpu} : demanded CPU fraction; $C1$: execution time of task one; $T1$: period of task one; $C2$: execution time of task two; $T2$: period of task two; $C3$: execution time of task three; $T3$: period of task three

7 — Pronunciation

Say subscripts, powers, Greek letters and parentheses explicitly so the relation can be reconstructed without seeing the page.

8 — Units

each C/T ratio dimensionless; U_{cpu} dimensionless

9 — Convention

Keep the same frame, time basis, system boundary and sign convention throughout the substitution.

10 — Why this operation

Addition combines contributions expressed on the same basis into one coherent total.

11 — Assumptions

All contributions must use a common unit and scope.

12 — Unit check

each C/T ratio dimensionless; U_{cpu} dimensionless Verify that dimensional reduction reaches the unit of the requested output.

13 — Numerical case

With $C1/T1=2/10$, $C2/T2=1/5$ and $C3/T3=1/20$, $U_{\text{cpu}}=0.20+0.20+0.05=0.45$.

14 — Why the calculation works

Addition combines contributions expressed on the same basis into one coherent total.

15 — Independent check

Removing one contribution from the total must recover the sum of the others.

16 — Mental estimate

Adding the dominant terms first quickly gives the order of magnitude.

17 — Interpretation

Utilisation below one is not sufficient to prove all deadlines schedulable.

18 — What the result does not prove

The calculation proves neither that the inputs are correct nor that the model remains valid outside the declared scenario; those questions require separate evidence.

19 — Sensitivity

The total changes linearly with each contribution when the others stay fixed.

20 — Guided and autonomous exercises

Guided exercise. $C1/T1=1/4$, $C2/T2=1/10$, $C3/T3=1/20$.

Detailed guided correction — open after trying

$$C1/T1=1/4, C2/T2=1/10, C3/T3=1/20. U_{\text{cpu}}=0.25+0.10+0.05=0.40.$$

Autonomous exercise. $C1/T1=3/10$, $C2/T2=2/10$, $C3/T3=1/10$.

1. Power budget and mission profile

The budget separates average power, peak power and energy over time. A 2 kW load for ten minutes does not consume the same energy as a 500 W load for ten hours. For each mission phase, engineers add loads, losses and margin, then compare them with available generation and storage.

Engineering habit. For “power budget and mission profile”, write the inputs, outputs, units and validity range first. Then build one nominal and one degraded case. This two-case approach prevents a correct equation from being mistaken for an operationally robust Mars architecture.

2. Generation: solar, alternatives and pointing

Solar generation depends on incident flux, area, efficiency, temperature and orientation. On Mars, dust and seasons further change availability. Other architectures may use nuclear or hybrid sources. The systems question remains: what power can be guaranteed in the worst credible condition, not merely at local noon in the nominal case?

Engineering habit. For “generation: solar, alternatives and pointing”, write the inputs, outputs, units and validity range first. Then build one nominal and one degraded case. This two-case approach prevents a correct equation from being mistaken for an operationally robust Mars architecture.

3. Batteries: energy, power, depth of discharge and ageing

A battery has energy capacity but also current, temperature and state-of-charge limits. Deep cycling and adverse temperatures accelerate ageing. Sizing must preserve margin after years of use, not only on day one. The battery management system protects cells but can itself become a critical function.

Engineering habit. For “batteries: energy, power, depth of discharge and ageing”, write the inputs, outputs, units and validity range first. Then build one nominal and one degraded case. This two-case approach prevents a correct equation from being mistaken for an operationally robust Mars architecture.

4. Conversion and distribution

Solar arrays and batteries do not necessarily provide the voltage every load needs. Converters, buses, protection and switches distribute energy. The architecture must isolate a fault without blacking out the entire vehicle. Short-circuit currents, startup surges and conversion losses belong in the budget, as do return paths and electromagnetic compatibility.

Engineering habit. For “conversion and distribution”, write the inputs, outputs, units and validity range first. Then build one nominal and one degraded case. This two-case approach prevents a correct equation from being mistaken for an operationally robust Mars architecture.

Approfondissement — Electrical architecture: a bus is more than a nominal voltage

Saying a habitat uses a 120-V bus tells almost nothing about resilience. Engineers need peak power, fault current, protection, converters, grounding, island modes and load priorities. In direct current, electrical power P is $P = U \times I$, where P is power in watts, U voltage in volts and I current in amperes. A 6-kW load on 120 V therefore draws $6,000 \div 120 = 50$ A. Ten identical loads starting together would require 500 A.

Startup matters because motors, compressors and converters can draw transient current. Protection has to keep one failed load from removing all healthy loads. Selective protection attempts to trip the device nearest the fault before upstream protection opens. On Mars, this is not merely fire protection; life support, thermal control and communications may have to remain powered while an industrial workshop is isolated.

5. Sensors, data acquisition and onboard computing

Avionics converts the physical world into data: voltages, temperatures, accelerations, images and pressure. Computers timestamp, filter and interpret those measurements. A number is useful only if its range, precision, sample rate and status are known. A sensor frozen at a plausible value can be more dangerous than one that is obviously dead.

Engineering habit. For “sensors, data acquisition and onboard computing”, write the inputs, outputs, units and validity range first. Then build one nominal and one degraded case. This two-case approach prevents a correct equation from being mistaken for an operationally robust Mars architecture.

6. Data buses and interfaces

Subsystems exchange commands and telemetry through buses and networks. Interfaces specify formats, units, cadence, latency, error behaviour and authority. Two pieces of equipment can work perfectly alone and fail after integration if one expects degrees while the other sends radians, or if a command is interpreted in the wrong state.

Engineering habit. For “data buses and interfaces”, write the inputs, outputs, units and validity range first. Then build one nominal and one degraded case. This two-case approach prevents a correct equation from being mistaken for an operationally robust Mars architecture.

7. Flight software: states, tasks and determinism

Flight software executes periodic tasks, manages mission states and applies safety rules. In a critical system, engineers must explain which condition triggers a transition and which component has authority. Determinism means a critical function meets timing and ordering constraints; being ‘fast on average’ is not enough.

Engineering habit. For “flight software: states, tasks and determinism”, write the inputs, outputs, units and validity range first. Then build one nominal and one degraded case. This two-case approach prevents a correct equation from being mistaken for an operationally robust Mars architecture.

Approfondissement — Real-time software: a correct answer delivered too late may be wrong

Flight software has to produce the right result inside a guaranteed time window. Functional correctness and timing correctness are therefore separate requirements. If a guidance loop must execute every 20 ms, where ms means millisecond, a mathematically perfect result that arrives in 35 ms can destabilize the vehicle. Worst-case execution time, or WCET, has to remain below the allocated budget with margin.

Task concurrency creates another risk: a low-value function may block a resource needed by a critical one. Real-time architectures therefore define priorities, memory access, message queues and watchdogs. Mars communications delay makes this discipline even more important. A habitat cannot wait for a distant operator to reboot software that is controlling cabin pressure or battery temperature.

8. FDIR and safe mode

Fault Detection, Isolation and Recovery covers detecting, locating and responding to faults. Automation can save a vehicle but can also amplify a bad measurement. Safe mode aims for a sustainable state where power, thermal control and communications are stabilized while diagnosis continues. It must be tested as a real configuration, not merely described in a document.

Engineering habit. For “fdir and safe mode”, write the inputs, outputs, units and validity range first. Then build one nominal and one degraded case. This two-case approach prevents a correct equation from being mistaken for an operationally robust Mars architecture.

Approfondissement — FDIR and degraded modes: keep living after the first fault

FDIR means Fault Detection, Isolation and Recovery. Detection recognizes that observations are inconsistent with expected behavior. Isolation identifies the likely failing element without unnecessarily disabling healthy functions. Recovery chooses a fallback state. A safe mode may be uncomfortable: nonessential loads can be shed, production reduced, valves closed and the system held while more evidence is collected.

Good FDIR also distinguishes a real equipment failure from a lying sensor. If two temperature sensors disagree, independent clues may help: heater current, a nearby wall temperature, pressure or a thermal model. Mars recovery procedures also need authority rules: when may automation restart by itself and when must it wait for a human? Useful autonomy is not maximum software freedom; it is the minimum authority required to prevent a local fault from becoming loss of the habitat.

9. Critical software: requirements, testing and configuration

Software evolves until launch and often beyond. Code, parameters, interfaces and test evidence must therefore be version controlled. Every critical requirement must be verifiable. Unit tests alone are insufficient: integration, simulation, hardware-in-the-loop, fault scenarios and regression testing are needed to understand complete-system behaviour.

Engineering habit. For “critical software: requirements, testing and configuration”, write the inputs, outputs, units and validity range first. Then build one nominal and one degraded case. This two-case approach prevents a correct equation from being mistaken for an operationally robust Mars architecture.

Worked example step by step

The work method is always the same: state what every symbol represents, convert all units into a coherent system, perform the operation, then translate the result into a sentence. Finally perform an order-of-magnitude check. If the answer changes by a factor of one thousand because millimetres were treated as metres, the conversion must be visible in the calculation.

Progressive exercise

1. Choose a simple case and list every input with units.
2. Compute the nominal result without margin.
3. Vary the most uncertain parameter by $\pm 20\%$ and compare.
4. Inject one credible failure and explain which indicator detects it.
5. Decide whether the system continues, degrades or stops.

Reasoned solution

A good solution is not only the final number. It shows conversions, why the equation applies, sensitivity and the resulting decision. If different plausible assumptions lead to the same operational decision, the design is relatively robust to that uncertainty. If a small variation reverses the decision, the parameter becomes a priority for measurement or margin.

Validation mini-project

Build a two-to-four-page engineering note applying this course to one Mars subsystem. Include need, assumptions, functional sketch, hand calculation, second calculation or simulation, uncertainties, injected failure, decision criteria and three primary references. The goal is a chain of evidence that another reader can reproduce.

Common errors to detect

- mixing units or frames without explicit conversion;
- presenting calculated values as measured data;
- ignoring a model's validity range;
- confusing numerical precision with physical accuracy;
- sizing only the nominal case with no margin or degraded mode.

Mission reasoning laboratory — connect the calculation to a real decision

The purpose of this laboratory is not to memorise isolated definitions. It links the module to a Mars mission decision, identifies evidence, tests boundary cases and preserves an explicit recovery path.

Build the power budget by mission state

A single total-watt number hides when loads operate together. Create states such as cruise, science, communication pass, EVA support, safe mode and battery recharge. For each state, list essential and discretionary loads, conversion losses and expected duration. Energy storage is sized from the timeline, while generation and distribution must support instantaneous power. A system can have enough daily energy yet still fail because a short peak exceeds the bus, converter or battery current limit.

Protect batteries from optimistic capacity

Nameplate ampere-hours are measured under specific test conditions and do not equal guaranteed mission-usable energy. Depth-of-discharge limits, temperature, ageing, discharge rate and cell imbalance reduce usable capacity. Reserve policy should also protect

survival modes and uncertain recovery time. Track both charge and energy: Ah multiplied by representative voltage can estimate Wh, but detailed models should use the actual voltage-versus-state-of-charge behaviour and include conversion efficiency.

Design distribution for fault containment

Power distribution defines which loads share a source, switch, converter, breaker and return path. A faulted branch should be isolated before it drags down essential services. Cross-strapping can increase flexibility but also create new paths for fault propagation or operator error. Map each critical load to its power sources and protection devices, then ask which single failure still removes multiple supposedly redundant functions. Electrical architecture is part of system safety, not only wiring.

Follow information through avionics

A sensor measurement is useful only if it survives conditioning, digitisation, time tagging, transmission, computation and command output. Each interface introduces failure modes: stale data, wrong units, bus congestion, timestamp error, corrupted packets or mismatched configuration. Trace one critical variable from physical sensor to software decision and back to actuator response. This end-to-end view reveals dependencies that component-level tests may miss.

Make flight software timing explicit

Flight software often manages many tasks with different rates and priorities. Deterministic timing matters because a correct algorithm delivered too late can still be unsafe. Define execution periods, deadlines, watchdogs, timeout behaviour and resource limits. Test not only normal logic but overloaded and degraded states. A processor reboot, bus fault or runaway task should lead to a bounded recovery path rather than an undefined combination of partially completed commands.

Use safe mode to preserve energy and commandability

A good safe mode reduces nonessential loads, stabilises attitude or thermal state and keeps a communication or autonomous recovery path alive. The safe configuration must be power-positive for long enough to diagnose the fault. Entering safe mode can itself be hazardous if it disables needed heaters, pumps or navigation. Analyse each transition and test that the spacecraft or habitat can remain within battery, thermal and life-support limits while waiting for recovery.

Verify software and hardware together

Software requirements are tested against simulated and real interfaces, but system behaviour also depends on electrical timing, sensor ranges, actuator dynamics and network failure modes. Hardware-in-the-loop testing helps reveal interactions that unit tests cannot. Configuration management is essential: the tested software build, parameter set and interface definition must match the deployed system. A perfectly tested old build does not verify a later field modification.

Progressive exercises — solve first, then open the correction

Synthesis exercise — power and energy

A rover has a 28 V battery with 90 Ah nameplate capacity. Operations limit usable capacity to 70% of nameplate, and average discharge voltage over that window is estimated at 26 V. Estimate usable energy in kWh. Then determine how long a constant 420 W safe-mode load could run if only 80% of that usable energy may be allocated to safe mode.

Detailed correction — Synthesis exercise — power and energy

Usable charge. $90 \text{ Ah} \times 0.70 = 63 \text{ Ah}$.

Approximate usable energy. $63 \text{ Ah} \times 26 \text{ V} = 1,638 \text{ Wh} = 1.638 \text{ kWh}$.

Safe-mode allocation. $0.80 \times 1.638 = 1.310 \text{ kWh}$. Runtime $\approx 1.310 \text{ kWh} / 0.420 \text{ kW} \approx 3.12 \text{ h}$. Real sizing would include converter losses, temperature and battery ageing.

Beginner vocabulary checkpoint

Focus or tap each highlighted term for its definition. The goal is to remove hidden prerequisites before the learner reaches a safety-significant calculation or procedure.

- voltage — Electrical potential difference that drives charge through a circuit.
- current — Rate of electric charge flow, measured in amperes.
- battery capacity — Stored charge or energy capability of a battery under specified conditions.
- depth of discharge — Fraction of battery capacity removed relative to a defined full state.
- state of charge — Estimate of remaining battery charge relative to its usable capacity.
- power budget — Accounting of electrical generation, conversion, storage and loads over mission states.
- energy budget — Accounting of accumulated electrical energy produced, stored and consumed over time.

- peak load — Highest short-duration power demand that a source or bus must support.
- duty cycle — Fraction of time a device operates in a given state or power level.
- solar array — Photovoltaic surface that converts incident sunlight into electrical power.
- maximum power point — Voltage-current operating condition at which a photovoltaic array produces maximum electrical power for the current environment.
- DC-DC converter — Power-electronic device that changes one direct-current voltage level into another.
- inverter — Power-electronic device that converts direct current to alternating current when AC is required.
- breaker — Protective switching device intended to interrupt abnormal electrical current.
- load shedding — Intentional disconnection of lower-priority loads to preserve essential power capability.
- brownout — Undervoltage condition that can cause malfunction even when power is not fully lost.
- sensor — Device that converts a physical quantity into information for monitoring or control.
- data acquisition — Sampling, conditioning and digitisation of sensor signals for onboard processing.
- flight computer — Onboard processor executing critical flight or vehicle-management software.
- data bus — Shared communication pathway used by onboard computers, sensors and actuators.
- determinism — Property that timing and behaviour remain bounded and predictable under specified conditions.
- watchdog — Independent monitor that detects software or processor failure and can trigger recovery.
- safe mode — Predefined low-risk spacecraft state designed to preserve essential power, thermal and communications capability.
- software requirement — Testable statement of behaviour or constraint that flight software must satisfy.
- verification — Evidence that an implementation meets its specified requirement.
- validation — Evidence that the implemented system fulfils the intended operational need in the relevant environment.
- configuration management — Control of approved software versions, parameters, interfaces and associated records.

Decision closeout — evidence before acceptance

Separate power-positive from energy-positive operation

A mission mode can have enough total energy over a day and still be impossible at a particular moment. Solar generation may exceed daily consumption while a short communications pass plus heaters and a pump overload the converter or battery current limit. Conversely, a high peak may be acceptable if storage can supply it briefly and recharge later. Check both instantaneous power and accumulated energy for every critical state. This distinction is essential for safe mode: survival requires not only low average energy use but also a bus architecture that can start pumps, heaters or transmitters when needed.

Model battery ageing as loss of operational choice

As a battery ages, usable capacity, peak power capability and cold-temperature performance can degrade. The consequence is not just a smaller number in the energy budget; some mission modes may disappear entirely because their peak current or reserve duration is no longer supportable. Track capacity and internal-resistance trends through life, then update operational constraints. A colony should know when a rover can still perform local transport but no longer has enough protected reserve for a distant traverse. Ageing data should therefore feed procedures and scheduling, not remain only in maintenance records.

Treat data timing as a physical interface

Avionics data carries time as well as value. A perfectly accurate attitude sample delivered too late can be dangerous to a control loop. Sensors, buses and processors should therefore preserve timestamps and bounded latency. When messages are queued, retried or routed through redundant networks, software must know whether a value is fresh enough for the decision. Tests should inject delays, missing packets and out-of-order messages, because these conditions appear in real systems even when individual hardware components remain healthy.

Design watchdog recovery for partial failures

A watchdog that simply resets a computer can create a reboot loop if the underlying fault remains. Recovery logic should distinguish transient software upset, persistent hardware fault and bad external input where possible. After restart, the system needs a known configuration and a rule for restoring functions gradually. Critical outputs may remain inhibited until sensors and state estimates are credible again. This prevents a recovering computer from issuing commands based on stale memory or incomplete initialization.

Make software changes operationally auditable

Mars crews may eventually need parameter updates or software patches without immediate hands-on support from Earth. Safe change requires version identity, checksums, rollback capability, test evidence and a record of why the change was authorised. A patch

should not silently alter interfaces or power behaviour outside its declared scope. Before activation, verify that the previous known-good build remains recoverable. After activation, compare key telemetry and timing against expected signatures so that a subtle regression is detected before the new version becomes the only available configuration.

Cross-check energy telemetry against physics

Electrical telemetry can be wrong because of sensor drift, scaling error or software interpretation. Independent physical checks help: compare battery energy change with integrated bus power, compare solar-array output with illumination and pointing, and compare an equipment current increase with its thermal dissipation. These relationships will not match perfectly because of losses and measurement uncertainty, but large inconsistencies reveal bad data or an unmodelled load. Operators should know which coarse checks can be performed quickly during an anomaly.

Close the chain from fault to mission consequence

A bus undervoltage can corrupt sensor data; corrupted data can trigger software protection; protection can shed loads; load shedding can reduce thermal circulation or communications. This chain shows why power, avionics and software cannot be certified independently and then assumed safe together. For each critical fault, trace the sequence through electrical distribution, data validity, software state and physical actuators. Identify where the chain is detected, where it is contained and what evidence confirms recovery. The result is a system-level fault-protection case rather than four separate subsystem checklists.

Primary sources and pathways

- [NASA — Small Spacecraft Avionics](#)
- [NASA-STD-8739.8 — Software Assurance and Software Safety](#)
- [www.nasa.gov — institutional documentation](#)
- [standards.nasa.gov — institutional documentation](#)
- [Space Academy — full pathway](#)

FR ↔ EN parity matrix

EN	FR
Electrical power on a DC bus	Puissance électrique sur un bus continu
Energy consumed over time	Énergie consommée sur une durée
Usable battery energy	Énergie utilisable d'une batterie
Captured solar power	Puissance solaire captée
Current demanded from a bus	Courant demandé au bus
CPU utilisation of periodic tasks	Utilisation processeur de tâches périodiques