

This document includes two AI prompts that help you prepare a complete and structured request.

Prompt #1

I am preparing a request for Hardware Prototype Development by the EnCata team and want to make sure my input contains enough information to define the right prototype scope, integrate the key systems, and understand what should be tested before moving to the next development stage.

Here is my current input:

"<<<INSERT YOUR INPUT HERE>>>"

At EnCata, a Hardware Prototype Development request should clarify the following areas:

1. Prototype goal

- What main question the prototype should answer
- What functionality, integration, or technical behavior must be validated
- Why the prototype is needed now
- What decision should become possible after testing
- Whether the prototype is intended mainly for engineering validation, an investor demo, user feedback, or another purpose

2. Current development stage

- Whether you already have a PoC, mockup, CAD, electronics, firmware, or an existing prototype
- What has already been tested successfully
- What is still untested
- Which parts of the current solution are considered reliable
- Which parts may still need redesign

3. Core prototype functions

- What the prototype absolutely must do
- Which functions are required to answer the main prototype question
- Which functions are useful but not essential for this iteration
- Which final-product features can be excluded from the prototype

4. System architecture

- What mechanical, electronic, firmware, power, sensor, actuator, connectivity, and enclosure subsystems are involved
- How these subsystems are expected to interact
- Which parts already exist
- Which parts still need to be developed or integrated

5. Electronics and components

- Existing development boards, modules, PCBs, sensors, actuators, or other electronics
- Components already selected or preferred
- Whether off-the-shelf modules are acceptable
- Whether custom electronics may be required
- Known power, signal, size, or integration constraints

6. Mechanical system and enclosure

- Existing mechanisms, CAD, enclosure concepts, or physical layouts
- Required moving parts, fastening, mounting, or assembly logic
- Approximate size and form-factor constraints
- Whether the enclosure needs to be representative of the future product
- What mechanical behavior must be tested

7. Firmware and control logic

- What the prototype needs to control, measure, communicate, or record
- Existing firmware, if any
- Required sensor reading or actuator behavior
- Connectivity or communication needs
- Whether data logging or debug modes are required for testing

8. Power system

- How the prototype should be powered
- Battery or external power requirements
- Expected runtime, if relevant
- Peak power or load conditions, if known
- Whether power behavior itself needs to be validated

9. Operating conditions

- Where and how the prototype will be tested
- Expected load, movement, heat, vibration, moisture, dust, handling, or other environmental conditions
- Whether bench testing is sufficient
- Which real-use conditions should be reproduced during validation

10. Prototype fidelity

- Which parts need to be close to the future product
- Which parts can remain simplified
- Whether appearance matters
- Whether ergonomics or user interaction need to be tested
- Whether the enclosure, materials, or manufacturing methods need to represent the next development stage

11. Testing and success criteria

- What should be measured, observed, or demonstrated
- What result would be considered successful
- What failure conditions should be checked
- Whether test data, logs, measurements, photos, or demonstrations are required
- Whether several prototype iterations are expected

12. Production and component considerations

- Target product cost or BOM expectations, if known
- Components that may be difficult to source
- Whether component availability or lifecycle matters
- Whether the prototype should avoid solutions that are impractical for later development
- Whether future manufacturing constraints should already influence prototype decisions

13. Existing materials

- Product description
- PoC results
- CAD files
- Schematics or PCB files
- Firmware
- BOM
- Component datasheets

14. Next development stage

- What should happen after the prototype
- Whether the expected next step is MVP, DFM, pilot batch, another prototype iteration, or another stage
- What must be proven before moving forward

- Which prototype findings should guide the next engineering decisions

Your task:

- Identify missing, unclear, or weakly defined areas in my input
- Structure your answer by the categories above
- For each gap, briefly explain why it matters for Hardware Prototype Development
- Separate critical gaps from information that can reasonably be clarified during development
- Identify functions or subsystems that may create integration risks
- Point out where it is unclear whether an off-the-shelf solution is sufficient or custom engineering may be required
- Flag requirements that are too vague to define a meaningful prototype test
- Do not turn the request into an MVP specification
- Do not repeat PoC validation if the core technical principle has already been proven
- Do not invent components, performance targets, test criteria, materials, costs, or technical solutions that I have not provided
- Do not rewrite my request. Focus only on identifying gaps and questions

Important instructions for your response:

- Use clear and practical language
- Keep clarification questions short and easy to answer
- Avoid unexplained engineering terminology
- If technical wording is necessary, briefly explain it in simple terms
- Focus on helping a non-engineering person understand what information is needed to define a focused prototype scope

Prompt #2

I am continuing preparation of my Hardware Prototype Development request based on your previous analysis.

Here is my initial input:

"<<<INSERT INITIAL INPUT HERE>>>"

Here is the additional information I have provided to fill the gaps:

"<<<INSERT ADDITIONAL INFORMATION HERE>>>"

Use the same Hardware Prototype Development categories from the previous analysis:

1. Prototype goal
2. Current development stage
3. Core prototype functions
4. System architecture
5. Electronics and components
6. Mechanical system and enclosure
7. Firmware and control logic
8. Power system
9. Operating conditions
10. Prototype fidelity
11. Testing and success criteria
12. Production and component considerations
13. Existing materials
14. Next development stage

Your task:

- Re-evaluate the combined input
- Check whether the previously identified critical gaps have been resolved
- Identify anything that is still missing, unclear, or contradictory

- Keep assumptions separate from confirmed requirements
- Check whether the prototype has a clear goal and meaningful success criteria
- Identify unresolved integration risks between subsystems
- Do not invent missing technical information

If critical information is still missing:

- List only the remaining critical gaps
- Briefly explain why each one matters
- Ask concise clarification questions
- Do not generate the final request yet

If the information is sufficient:

- Compile a structured Hardware Prototype Development request for EnCata
- Use the available information only
- Mark unresolved points as "To be clarified during prototype development"
- Mark unvalidated technical ideas as "Assumption"
- Keep the prototype scope focused on what actually needs to be tested

Structure the final request as follows:

1. Project overview
2. Prototype goal
3. Current development stage
4. Core prototype functions
5. System architecture
6. Electronics and components
7. Mechanical system and enclosure
8. Firmware and control logic
9. Power system
10. Operating conditions
11. Prototype fidelity
12. Testing and success criteria
13. Production and component considerations
14. Existing materials
15. Main risks and open questions
16. Next development stage

For the final request:

- Clearly distinguish what already exists from what still needs to be developed or integrated
- Separate must-have prototype functions from non-essential functions
- Indicate where off-the-shelf solutions may be sufficient and where custom engineering may need to be evaluated
- Preserve any known test criteria and identify criteria that still need to be defined
- Keep future-product requirements separate from requirements that are necessary for this prototype iteration

Important instructions:

- Do not turn the request into an MVP specification
- Do not repeat PoC validation if the core technical principle has already been proven
- Do not assume custom electronics, mechanics, or firmware are required unless the input supports it
- Do not invent components, performance targets, test criteria, materials, costs, or technical solutions
- Keep the response clear, concise, and practical
- Do not include explanations outside of the requested output