

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

Prompt #1

I am preparing a request for Firmware Development Services by the EnCata team and want to make sure my input contains enough information to start the work without delays or unnecessary rework.

Here is my current input:

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

At EnCata, a firmware development request should clarify the following areas:

1. Product overview and current stage

- What type of device or physical product is being developed
- What the device is expected to do
- What stage the project is currently at: concept, proof of concept, functional prototype, integrated prototype, or pilot-ready device
- Whether the electronics and mechanical design already exist

2. Required device behavior

- How the device should behave during normal operation
- What should happen during startup and shutdown
- What operating modes are required
- How the device should respond to errors, power loss, communication failures, or invalid sensor data
- Which actions must happen automatically and which are controlled by the user

3. Existing hardware

- Which microcontroller, processor, PLC, FPGA, or other control platform is used
- Whether a custom PCB or development board is available
- Which sensors, actuators, displays, motors, valves, pumps, buttons, or other peripherals are connected
- Whether schematics, PCB files, and component datasheets are available

4. Firmware scope

- Whether firmware must be developed from scratch, extended, debugged, refactored, or optimized
- Whether low-level drivers, control algorithms, state machines, signal processing, or data logging are required
- Whether there is existing source code that must be reused

5. Firmware architecture and operating system

- Whether the device should use Bare Metal, a simple loop-based architecture, RTOS, Embedded Linux, ROS, or another platform
- Whether the architecture has already been selected
- Whether multiple tasks must run at the same time
- Whether strict timing or real-time behavior is required

6. Interfaces and communication

- Which wired interfaces are used: USB, SPI, I2C, UART, RS-232, RS-485, CAN, Ethernet, or others
- Which wireless technologies are required: Wi-Fi, BLE, Bluetooth, LoRaWAN, NB-IoT, NFC, GPS, cellular, UWB, or others
- Whether custom communication protocols are required
- What external systems the device must communicate with

7. Cloud, mobile app, and IoT integration

- Whether the device connects to a cloud platform, mobile app, web interface, gateway, or another device
- What data should be sent or received
- Whether MQTT or another data exchange method is required
- Whether the device must continue operating if the connection is lost

8. Sensors, actuators, and control logic

- What the firmware must measure, calculate, or control
- Required accuracy, response time, sampling frequency, or update rate, if known
- Whether calibration is required
- Whether motors, pumps, valves, heaters, or other actuators need closed-loop control

9. Power and energy behavior

- How the device is powered
- Whether battery operation, low-power modes, sleep modes, or power-saving logic are required
- Expected runtime or energy consumption targets
- Whether the firmware must control charging, power rails, or startup sequence

10. Fault handling and reliability

- Which failure scenarios must be handled
- Whether watchdogs, retries, safe modes, recovery logic, or event logging are required
- What the device should do if a sensor, actuator, communication module, or power supply fails
- Whether the product has safety-critical behavior

11. Mechanical and physical constraints

- Whether enclosure geometry, heat, vibration, motor inertia, airflow, button behavior, or other physical factors affect firmware logic
- Whether the firmware must be tested in the assembled device
- Whether there are mechanical limits, end stops, delays, or movement restrictions

12. Testing and debugging

- What should be tested on real hardware
- Whether unit testing, integration testing, hardware bring-up, protocol testing, or field testing is required
- What test results, logs, or reports are expected
- Whether special test modes or production diagnostics are needed

13. Deployment and updates

- How firmware should be uploaded to the device
- Whether a bootloader is required
- Whether firmware updates should be performed locally, during production, or remotely
- Whether rollback or recovery after a failed update is required

14. Production and service requirements

- Whether firmware must support production testing, calibration, programming, or end-of-line checks
- Whether service technicians need diagnostic tools or service modes
- Whether several device versions or hardware revisions must be supported

15. Documentation and handover

- What documentation is required
- Whether architecture notes, interface descriptions, test procedures, deployment instructions, or service documentation are needed
- Whether another team will maintain the firmware after delivery

16. Existing project inputs

- Available requirements, schematics, PCB files, source code, component lists, protocol descriptions, test results, videos, or hardware samples

- Existing technical decisions that should not be changed
- Known firmware or hardware issues

17. Timeline and project outcome

- Desired project timeline
- Important milestones or demonstration dates
- Whether the expected result is a working proof of concept, functional prototype, integrated device, tested firmware build, or pilot-ready product

18. Risks and open questions

- What is still unclear or undecided
- Known concerns related to hardware stability, timing, communication, power consumption, sensor accuracy, control behavior, maintainability, or integration

Your task:

- Compare my input with the requirements above
- Identify missing, unclear, or weakly defined areas
- Structure your answer by the categories above
- Phrase each missing item as a short, practical question that I can answer
- Briefly explain why the requested information matters for firmware development

Important instructions for your response:

- Use clear, everyday language that a non-engineer can understand
- Avoid unexplained abbreviations and specialist terminology
- If a technical term is necessary, explain it briefly in simple language
- Keep every clarification request concise
- Make it clear exactly what information I should provide
- If an exact technical value is unavailable, ask for an approximate range or a practical example
- Do not rewrite my request
- Do not propose a final firmware architecture
- Focus only on identifying what information is still needed

Prompt #2

I am continuing preparation of my Firmware Development Services 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>>>"

At EnCata, a firmware development request requires clarity in the following areas:

1. Product overview and current stage
2. Required device behavior
3. Existing hardware
4. Firmware scope
5. Firmware architecture and operating system
6. Interfaces and communication
7. Cloud, mobile app, and IoT integration
8. Sensors, actuators, and control logic
9. Power and energy behavior
10. Fault handling and reliability
11. Mechanical and physical constraints
12. Testing and debugging

13. Deployment and updates
14. Production and service requirements
15. Documentation and handover
16. Existing project inputs
17. Timeline and project outcome
18. Risks and open questions

Your task:

1. Re-evaluate the combined input: initial input + additional information
2. Check whether the available information is sufficient to prepare a useful Firmware Development Services request
3. Based on the available information, compile a structured request specifically for the EnCata team

If some information is missing:

- Make reasonable high-level assumptions only where necessary
- Clearly mark every assumption
- Clearly mark any information that still needs to be confirmed
- Do not invent exact hardware specifications, timing requirements, protocols, performance values, safety requirements, or test limits

Structure the final request as follows:

1. Project overview
2. Current product and hardware stage
3. Required device behavior and operating modes
4. Existing electronics and hardware platform
5. Firmware scope and current code status
6. Firmware architecture and operating system
7. Sensors, actuators, interfaces, and communication
8. Cloud, mobile app, and external system integration
9. Timing, performance, and power requirements
10. Fault handling, recovery, and reliability requirements
11. Mechanical and physical constraints affecting firmware
12. Testing, debugging, and validation scope
13. Deployment, bootloader, and update requirements
14. Production, calibration, and service requirements
15. Documentation and handover expectations
16. Existing files, code, and hardware inputs
17. Timeline and milestones
18. Risks, assumptions, and open questions
19. Expected development outcome and deliverables

Output requirements:

- Produce a structured, ready-to-send request for EnCata
- Use clear and concise language
- Preserve all relevant details provided by the user
- Clearly mark assumptions and missing information
- Do not add explanations outside of the final request