

# Epicure Robotics

## Robotic Software Engineer - Task Document

Version: 1.0

### 1. Objective

Develop a Python-based communication system that interfaces between a computer (Python), an ESP microcontroller (via MQTT), and an STM microcontroller (via UART). The STM will be responsible for controlling a stepper motor and an LED connected to a PCB. (Hardware implementation is not completely necessary; just the code base is sufficient, we will test with our hardware)

### 2. Functional Overview

The system architecture is as follows:

Python Script → MQTT → ESP[ESP32 wroom32d] → UART → STM[STM32f407VET6]  
→ Stepper Motor[Nema17 4.4kgcm] + LED[5v led]

- The Python program will take input commands from the user.
- It will publish these commands via MQTT to the ESP.
- The ESP will receive MQTT messages and send them via UART to the STM.
- The STM will interpret the received command and control the connected stepper motor and LED accordingly.

The intern's responsibility is only to write the Python, ESP, and STM codes for communication and command handling.

### 3. Requirements

Python Program:

1. Continuously read user input (e.g., motor/LED commands).
2. Publish the input to a specific MQTT topic (e.g., "epicure/commands").
3. Maintain connection reliability and handle basic error cases (e.g., reconnection on broker failure).

ESP:

1. Subscribe to the MQTT topic.
2. On receiving a command, send it to STM via UART.
3. Maintain proper UART communication protocol and handle message integrity.

STM:

- Controls a stepper motor and an LED based on UART commands received from the ESP.

## 4. Example Flow

User input in Python console:

motor:100:1

led:on

Flow:

1. Python publishes → motor:100:1 to epicure/commands
2. ESP receives MQTT message → forwards 'motor:100:1' via UART
3. STM receives UART data → moves stepper motor 100 steps forward
4. For 'led: on' → STM turns LED on

## 5. Deliverables

1. Python Script:
  - Handles MQTT connection, publishes user commands.
  - Includes proper exception handling and clean exit.
2. ESP Firmware:
  - MQTT subscriber + UART forwarder implementation.
  - Basic serial protocol handling.
3. STM Firmware:
  - UART Listener.
  - Stepper and LED control based on command.
4. Documentation:
  - Short readme explaining setup, dependencies, and run instructions.
  - Clear explanation of message format and flow.

## 6. Next Steps

After evaluating the submitted task:

1. Shortlisted candidates will be invited for an interview call to discuss their code and logic.
2. The interview will be scheduled within 3 days of task submission.
3. Within 3 days after the interview, the offer letter will be issued to the selected candidates.

Duration & Conversion:

- For Freshers: 3-month internship period.
- For Experienced Candidates (1–2 years): 1-month probation period.
- Based on performance, a Pre-Placement Offer (PPO) may be extended for a full-time role at Epicure Robotics.