



Flomobility Pvt Ltd

No. 531/144, Ground Floor, 100 Feet Ring Rd
HSR Layout, Bengaluru, Karnataka 560102

Website: <https://flomobility.com>

Job Description – Firmware Engineer

Job type: On-site, Full-Time

Experience: 1 - 2 Years

Education Background

- Bachelor's degree (B.E./B.Tech.) in Electronics & Communication Engineering (ECE), Electrical & Electronics Engineering (EEE), Instrumentation Engineering, or a related field.
- Relevant certifications in Firmware Design (ESP32, STM32, Arduino), Embedded Systems

Company Overview

The world is moving towards an autonomous future where sooner or later everything that is driven, pushed or pulled today, will be moving by itself.

At Flo Mobility, we believe that the true adaptation of autonomy lies in the seamless coexistence of humans and bots. By leveraging the power of sensors, computer vision, edge computing, and actuators, we optimize this relationship to deliver the best speed, agility, comfort and control. Our hero product, FLO Hauler, is an Autonomous Electric Material mover, helping customers reduce time and cost of moving material across Construction, Agriculture, Mining and Warehousing industries.

Flo Hauler is being used by industry leaders like L&T, PSP, Total Environment to gain 50% savings on cost and time for the last mile of material movement. Our solutions are

offered on Raas (Robot-as-a-service) model. The no capex approach makes it easy for our clients to adopt our MMRs (Material Movement Robots).

Role Overview

We are looking for a Firmware Engineer with 1 to 2 years of hands-on experience to join our robotics team and build the next generation of autonomous systems. You will design, develop, and debug embedded firmware that controls and communicates with our MMRs and autonomous platforms, working closely with electronics, mechanical, software, and systems engineers.

This role offers real ownership of production-grade firmware on robots deployed in the field, along with the chance to deepen your expertise in STM32, ESP32, CAN, and robotics communication stacks.

You will collaborate with firmware and cross-functional engineering teams to troubleshoot hardware-software issues, optimize system performance, and develop robust firmware for production-grade robotic platforms.

Responsibilities

- **Embedded Firmware Development:** Design, develop, and maintain firmware in C/C++ for robotics platforms, primarily on STM32 and ESP32 microcontrollers.
- **STM32 Development:** Use STM32CubeIDE, STM32CubeMX, HAL, and CMSIS for peripheral configuration, driver development, hardware bring-up, and debugging.
- **ESP32 Development:** Build embedded applications using PlatformIO, Arduino framework, or ESP-IDF, including sensor integration, wireless connectivity, and system-level firmware.
- **Peripheral Interfacing:** Develop and debug drivers for GPIO, UART, SPI, I2C, ADC, DAC, PWM, timers, interrupts, DMA, and watchdogs.
- **CAN Communication:** Implement and debug CAN-based communication between STM32 controllers, motor controllers, sensors, and other embedded nodes, including frame design, IDs, baud rates, filtering, termination, and error handling.

- **Protocol & Sensor Integration:** Integrate sensors, motor drivers, displays, and other devices over UART, I2C, SPI, CAN, RS485, and similar interfaces. Exposure to higher-level protocols such as Modbus or CANopen is a plus.
- **Debugging & Diagnostics:** Debug firmware and hardware issues using ST-LINK/SWD, JTAG, oscilloscopes, logic analyzers, multimeters, CAN analyzers, and serial terminals. Analyze logs, reproduce field issues, and drive problems to root cause.
- **Python Tooling & Automation:** Write Python scripts for firmware testing, log analysis, CAN/serial data parsing, test automation, and diagnostic tools.
- **Testing & Validation:** Create test procedures, perform functional and integration testing, and validate firmware against system requirements.
- **Firmware Architecture:** Design modular, non-blocking, interrupt-driven firmware with state machines, fault handling, and clean driver abstractions.
- **Cross-functional Collaboration:** Work with electronics, mechanical, and software teams to resolve hardware-software integration issues and improve system performance.
- **Version Control & Releases:** Use Git for branching, merging, code reviews, and maintaining versioned firmware releases. Contribute to documentation of firmware design and release notes.

Skills

MANDATORY: MUST HAVE SKILLS AND HANDS-ON EXPERIENCE

- 1 to 2 years of hands-on experience in embedded firmware development
- Strong fundamentals in Embedded C/C++ (pointers, structures, enums, bit manipulation, memory management, interrupts)
- Hands-on experience with STM32 microcontrollers using STM32CubeIDE / STM32CubeMX, HAL, and CMSIS
- Hands-on experience with ESP32 (PlatformIO, Arduino framework, or ESP-IDF)
- Solid understanding of UART, SPI, I2C, GPIO, ADC, PWM, timers, interrupts, and DMA

- Practical experience with CAN protocol: frames, standard/extended IDs, arbitration, baud rates, acceptance filters, and error handling
- Proficiency in Python for scripting, tooling, and test automation
- Strong debugging skills using ST-LINK/SWD, JTAG, breakpoints, watchpoints, register inspection, and serial logs
- Comfortable with lab equipment: oscilloscopes, logic analyzers, multimeters, bench power supplies, and CAN/serial analyzers
- Proficiency with Git (repositories, branches, merges, tags, releases)

SKILLS REQUIRED FOR THE JOB ROLE

- **Embedded C/C++:** Good understanding of C programming, pointers, structures, enums, bit manipulation, memory management, interrupts, and low-level embedded programming concepts.
- **STM32 Ecosystem:** Hands-on experience with **STM32 microcontrollers, STM32CubeIDE, STM32CubeMX, HAL, and CMSIS**. Understanding of peripheral configuration, clock configuration, GPIO alternate functions, NVIC, timers, and interrupt handling.
- **ESP32 Development:** Experience developing firmware for **ESP32** using **PlatformIO**, Arduino framework, ESP-IDF, or equivalent embedded development environments.
- **Microcontroller Peripherals:** Practical knowledge of **GPIO, UART, SPI, I2C, ADC, DAC, PWM, timers, input capture, output compare, watchdogs, interrupts, and DMA**.
- **CAN Protocol:** Understanding of **CAN bus fundamentals, CAN frames, standard/extended IDs, arbitration, baud rates, acceptance filters, error handling, and CAN debugging**. Experience interfacing STM32 with external CAN transceivers and motor controllers is highly desirable.
- **Embedded Debugging:** Ability to debug firmware using **ST-LINK/SWD, JTAG, breakpoints, watchpoints, register inspection, serial logs, and debugging tools available in STM32CubeIDE**.
- **Lab Equipment:** Comfortable using **oscilloscopes, logic analyzers, multimeters, bench power supplies, and CAN/serial analyzers** to diagnose embedded system issues.

- **Communication Protocol Debugging:** Ability to analyze and troubleshoot **UART, SPI, I2C, and CAN signals** using oscilloscopes and logic analyzers.
- **Firmware Architecture:** Understanding of **state machines, interrupt-driven firmware, non-blocking programming, timing requirements, fault handling, and modular driver development.**
- **RTOS Fundamentals:** Basic understanding of **RTOS concepts**, including tasks/threads, scheduling, queues, semaphores, mutexes, and synchronization.
- **Build & Development Tools:** Experience with **STM32CubeIDE, PlatformIO, Git, serial terminals, logic analyzers, and embedded debugging tools.**
- **Version Control:** Proficiency with **Git**, including repositories, branches, commits, merges, tags, and maintaining stable firmware releases.
- **Problem Solving:** Strong debugging mindset with the ability to analyze logs, reproduce failures, identify root causes, and work collaboratively with hardware and systems engineers.

What We Look For

- A strong foundation in embedded C/C++, microcontrollers, communication protocols, and basic electronics
- A debugging mindset: the ability to analyze logs, reproduce failures, find root causes, and fix issues on real hardware
- Candidates who have shipped STM32/ESP32 projects, CAN-based systems, sensor or motor-controller interfaces, or other embedded products will be preferred
- Hands-on experience and the ability to solve real hardware/firmware problems are valued more than purely theoretical knowledge
- Ownership, curiosity, and the ability to work collaboratively in a fast-moving robotics startup