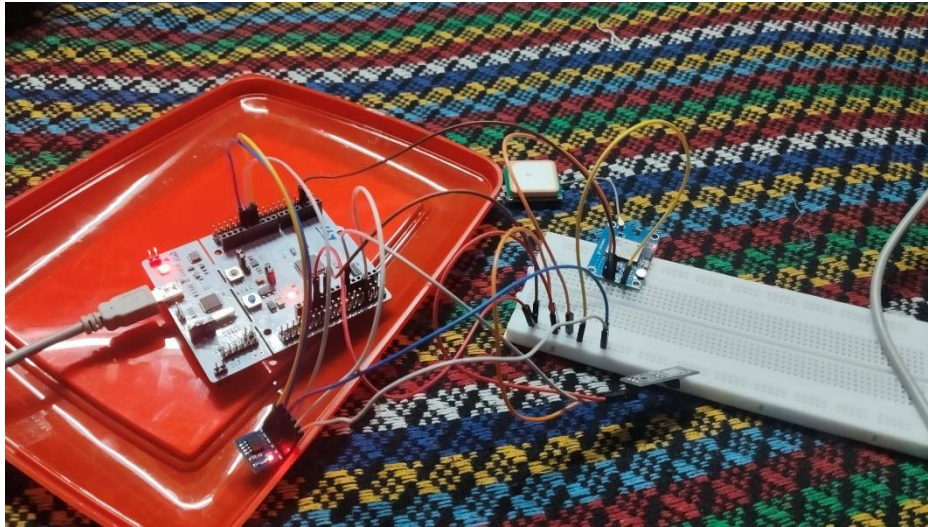


Data Logger using STM32



Introduction:

I designed and developed a bare-metal STM32-based real-time Data Logger system that acquires GPS data and analog sensor inputs, processes them efficiently using interrupt-driven architecture, and stores structured logs into external W25Q64 SPI flash memory.

The focus of this project was performance, reliability, and low-level peripheral control — without using any high-level libraries or RTOS.”

System Overview:

◆ Core Components

- 🧠 STM32 Microcontroller (Bare-metal Embedded C)
- 📍 GPS Module (USART1 – Interrupt Mode)
- 🖨️ USART2 Terminal (Real-time Monitoring)
- 📊 Analog Sensors (ADC-based acquisition)
- 💾 W25Q64 SPI Flash Memory (External Data Storage)

Key Engineering Strengths Demonstrated:

1. Bare-Metal Programming

No HAL shortcuts.

Direct register-level configuration:

- RCC
- GPIO
- USART
- SPI
- ADC
- NVIC

This proves strong hardware-level understanding.

2. Interrupt-Driven System Design

Used interrupts for:

- USART1 (GPS)
- USART2 (Terminal)
- ADC conversion completion

This ensures:

- Non-blocking execution
- Efficient CPU usage
- Real-time responsiveness

OUTPUT:

