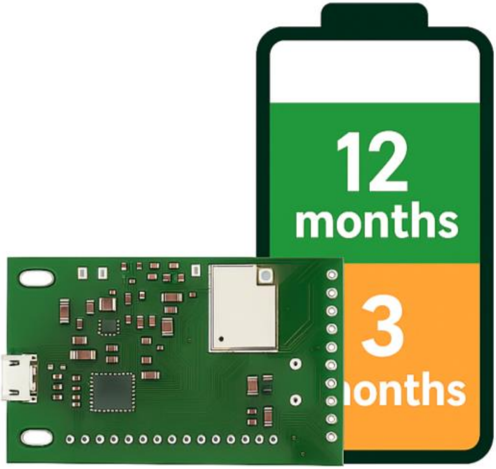
How I Extended Battery Life from 3 to 12 Months Using Low-Power Firmware Strategies



AZMAN BAKHTIAR
Embedded Developer

One of my clients had a promising IoT tracking device — compact, wireless, equipped with GPS and BLE, and fully battery-powered.

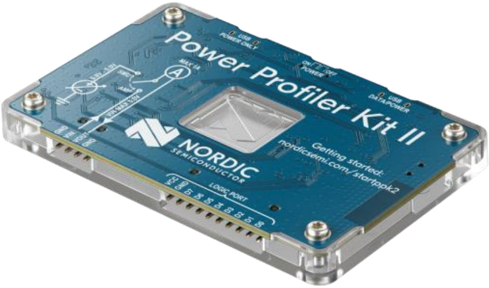
But there was a serious issue:

- The device’s battery only lasted 3 months per charge.
- The client needed at least 1 year of uptime — and they couldn’t afford a larger battery or a hardware redesign.

Analyzed Power Consumption Sources

I began by mapping out current consumption over time using a power analyzer:
What I found:

- GPS was active too often
- BLE was advertising too frequently
- MCU was rarely in deep sleep
- Peripherals (UART, I2C) were left on unnecessarily



AZMAN BAKHTIAR
Embedded Developer

Optimizing GPS – Cut the Biggest Power Consumer First

GPS was the largest energy hog — drawing ~30mA when active.

I implemented:

- Burst mode instead of continuous tracking
- Wakeups every 30–60 minutes for a quick fix
- Stored last known position and almanac data to reduce fix time
- Shut down GPS completely between updates

AZMAN BAKHTIAR
Embedded Developer

BLE Optimization – Smarter, Scheduled Communication

BLE was consuming too much power due to frequent advertising and immediate uploads.

I optimized it by:

- Increasing the advertising interval
- Sending data every 10 minutes instead of instantly
- Buffering readings, then transmitting as a single compact packet
- Disabling BLE between transmissions

AZMAN BAKHTIAR
Embedded Developer

MCU Sleep – Unlocking Deep Power Savings

The microcontroller was rarely sleeping, even when idle.

I solved this by:

- Using STOP mode (deep sleep) between cycles
- Setting the RTC alarm to wake the MCU every 10 minutes
- Ensuring wake-up time was under 1 ms for a smooth resume
- Disabling all unused peripherals before sleep (UART, ADC, I2C)
- Setting unused GPIOs to analog input or pull-down to avoid leakage

AZMAN BAKHTIAR
Embedded Developer

GPS & Data Sync – Every 10 Minutes, Not Every Second

Rather than activating GPS and sending data immediately:

- I used the RTC to wake the MCU every 10 minutes
- Activated GPS briefly, captured position, and powered it down again
- Buffered readings and sent them via BLE in a single session
- Compressed payloads to reduce transmission time

After the complete optimization:

- MCU slept 95% of the time
- BLE and GPS were active only a few seconds every 10 minutes
- Battery life extended from 3 months to over 12 months
- No hardware changes required

AZMAN BAKHTIAR
Embedded Developer

AZMAN BAKHTIAR
Embedded Developer

Reach Out for Free IoT & Embedded Consultation

Reshare if You Found it Helpful