**1. IMU Data Polling / Filtering**

* **Frequency**: 200–500 Hz → Standard for real-time motion tracking (see Bosch/TDK datasheets like BNO085 or MPU9250)
* **Execution time**: < 100 µs
  + Reading via SPI/I2C takes ~30–50 µs per sensor
  + Simple digital filtering (low-pass, smoothing) is minimal
* **Reference**: MPU9250 Register Map, SlimeVR firmware timing benchmarks.

**2. Sensor Fusion (Kalman/VQF)**

* **Frequency**: 100–200 Hz
  + Common rate for attitude estimation
  + VQF typically runs at 100 Hz
* **Execution time**: ~0.5–1 ms
  + Kalman filters (e.g., Mahony, Madgwick, or VQF) optimized for M4F cores complete within 1 ms
  + Verified in projects like:
    - [SlimeVR's VQF implementation](https://github.com/SlimeVR)
    - [PX4-ECL](https://github.com/PX4/ecl)
    - ST's own X-CUBE-MEMS1

**3. OSC/UDP Transmission**

* **Frequency**: 20–100 Hz
  + Matches standard motion capture / VR sync rates
* **Execution time**: ~0.1–0.3 ms
  + Lightweight OSC packet construction and UDP send using libraries like [CNMAT/OSC](https://github.com/CNMAT/OSC)
  + Based on STM32 + lwIP stack

**4. IR LED Toggling**

* **Frequency**: 100–500 Hz
  + For blinking/tracking modulation visible to IR cameras
* **Execution time**: < 100 µs
  + Controlled via GPIO or PWM
  + Negligible CPU use if done in interrupt or DMA

**5. Debug UART Logging**

* **Frequency**: On demand
  + Done via printf over UART
* **Execution time**: Asynchronous (background)
  + Buffered or DMA-based
* **Ref**: ST HAL UART with HAL\_UART\_Transmit\_IT() or DMA mode

**6. External Sync**

* **Trigger**: On GPIO edge interrupt
* **Execution time**: Instantaneous
  + GPIO interrupt with flag handling
* Similar usage in camera synchronization or timecode triggers.

**CPU Load Estimate (~40–50%)**

* Based on:
  + STM32F4 at 84 MHz ≈ 84 MIPS
  + Main tasks are spaced out and most require <1 ms per iteration
  + Even at 500 Hz = 2 ms cycle time → only ~0.8–1 ms total load per loop
* This gives a **CPU usage of ~40–50%**, leaving margin for future features (like calibration, logging, or display).

**Disclaimer**

This table is a **design-level estimate**, not a benchmark. It’s a **conservative, experience-based** planning tool engineers use to choose the microcontroller and understand the headroom.