**STM32F103C8T6 Communication Challenges for CubeSat Applications**

**1. Processing Power Limitations**

**Insufficient Data Rate Handling**

The STM32F103C8T6 microcontroller, based on a single-core 72 MHz architecture, lacks the computational capacity to handle high-data-rate communication protocols efficiently. Its limited processing bandwidth creates difficulty when simultaneously managing communication tasks and onboard data processing. These constraints are especially critical in real-time scenarios, where the microcontroller cannot perform complex modulation or demodulation while executing other time-sensitive operations.

**Mathematical Processing Limitations:** The absence of a hardware floating-point unit (FPU), forcing the system to rely on slow software-based floating-point computations. This inadequacy severely impacts digital signal processing (DSP) capabilities, making it impractical to implement advanced modulation schemes. Complex algorithms, such as forward error correction and encryption, place heavy demands on the CPU, consuming excessive cycles and introducing latency that further hampers communication responsiveness.

**2. Memory Constraints**

**Severe RAM Limitations:** With only 20 KB of SRAM, the STM32F103C8T6 cannot allocate sufficient memory for essential communication buffers or full protocol stacks. Modern communication protocols typically require significantly more memory for queuing and handling packet data, making them incompatible with the microcontroller’s limited resources.

**Flash Memory Restrictions**

The 64 KB flash memory imposes additional limitations by restricting the size and complexity of the firmware that can be stored. Communication libraries with comprehensive protocol support often exceed this capacity, forcing developers to make difficult trade-offs between features. Furthermore, the limited flash space complicates over-the-air firmware updates, which are critical for long-duration missions requiring post-launch software enhancements.

**3. Modern Communication Interface Gaps**

**Missing Critical Interfaces**

The STM32F103C8T6 does not feature native support for modern wireless standards such as Wi-Fi or Bluetooth, necessitating external modules that increase system complexity and integration overhead. Additionally, the microcontroller lacks high-speed peripheral interfaces, such as full USB 2.0 and Ethernet, which are often required for advanced communication modules. Compared to alternatives like the Raspberry Pi Pico, which offers more capable USB implementations, the STM32 falls short in interface support.

**Advanced Protocol Support Issues**

Implementing modern communication standards and technologies such as software-defined radio (SDR) is infeasible due to the microcontroller’s limited processing power. It cannot accommodate adaptive protocols that require real-time switching or dynamic resource allocation. Additionally, its limited capacity for secure computation restricts the implementation of modern encryption and authentication protocols, critical for mission data security.

**4. Advanced Modulation and Signal Processing**

**Digital Signal Processing Inadequacy**

The STM32F103C8T6 lacks the resources to execute complex modulation schemes such as QAM or OFDM and cannot implement sophisticated error correction algorithms. Real-time signal quality analysis is also infeasible, which undermines the ability to dynamically adjust system parameters based on link conditions. Adaptive communication systems, which are increasingly essential in space applications, are beyond the microcontroller’s capabilities.

**Software-Defined Radio Limitations**

The microcontroller is unsuitable for software-defined radio applications due to its inadequate computational performance and memory. It cannot support frequency agility, generate complex waveforms, or operate on multiple frequency bands simultaneously. These limitations prevent the implementation of advanced, flexible communication systems tailored to dynamic mission requirements.

Summary on critical limitations:

The STM32F103C8T6 is fundamentally constrained in several key areas:

1. It lacks the computational capability to support modern, high-data-rate communication.
2. It has insufficient memory to implement full-featured communication protocol stacks.
3. It cannot perform real-time digital signal processing or error correction.
4. It is incapable of supporting software-defined radio functionality.
5. It does not natively interface with modern high-speed communication hardware

To overcome these challenges, a hybrid architecture is recommended. The STM32F103C8T6 should be relegated to support functions such as radio power management, basic UART or SPI communication, and watchdog monitoring. Meanwhile, a more capable microcontroller like the Raspberry Pi Pico can serve as the primary communication controller.

This approach leverages the strengths of both systems:

* The STM32 handles low-level interfacing, power control, and auxiliary communication tasks.
* The Pi Pico is responsible for computationally intensive communication processing, including protocol handling, modulation/demodulation, and error correction.
* The combined system balances reliability with performance, providing a robust solution for CubeSat communications in low-Earth orbit and beyond.