

F Prime Automation Notes

- F Prime Features
 - Decomposes flight software into discrete components with well-defined interfaces
 - C++ framework provides core capabilities like message queues and threads
 - Autocoding tools that generate code from component/connection specs
 - Has a library of ready-to-use components
 - Contains a built-in testing framework
- Limitations
 - [CubeSat Flight Software: Insights and a Case Study | Journal of Spacecraft and Rockets](#)
 - "Additionally, neither cFS nor F Prime provides a built-in script engine for automating satellite operations, or a flexible bootloader for in-mission software updates. These features are essential for mission adaptability and efficiency."
 - A bootloader:
 - Runs before FSW, checking and executing it
 - Allows for modification and updating of FSW post-launch
- Levels of Automation: CmdSequencer < FpySequencer < Script Engine
 - CmdSequencer: [https://fprime.jpl.nasa.gov/latest/Svc/CmdSequencer/docs/sdd/Sequencing In F - F Prime](https://fprime.jpl.nasa.gov/latest/Svc/CmdSequencer/docs/sdd/Sequencing%20In%20F%20-%20F%20Prime)
 - Features
 - Loads and runs a sequence of commands
 - Can be triggered at absolute times or at times relative to previous commands
 - Flaws
 - Sequence will abort if any given command in the sequence returns a failed status
 - No conditional branching
 - No telemetry awareness
 - No jumps or loops
 - No mid-sequence debugging (by default)
 - FpySequencer: <https://fprime.jpl.nasa.gov/latest/Svc/FpySequencer/docs/sdd/>
 - Features
 - State machine controlling loading, validation, starting, and stopping of a sequence
 - Execution happens in a sectioned-off runtime
 - Support stack-based operations and flags -> conditional-like control
 - Debug mode
 - Flaws
 - Still in development (but might not be by the time we get to Spring 2028 LEONIRD I launch)
 - No structured control flow other than GOTO and IF
 - No functions or subroutines
 - Data models are limited to stack, registers, and basic get/set
 - Full Script Engine
 - Not available in F Prime
 - We would need to make this ourselves to do stuff like dynamically re-plan orientation based on live sensor readings rather than a pre-computed schedule
 - Options:
 - Embed a general purpose interpreter like CircuitPython or Lua
 - Takes a lot of storage space
 - Needs to be integrated with satellite parameter system
 - Program a minimal custom instruction set
 - Would probably include control flow, basic arithmetic, and get/set memory operations
 - Requires a lot of work
 - Example: <https://arxiv.org/pdf/2506.03152>
- Previous Uses
 - ASTERIA (Arcsecond Space Telescope Enabling Research in Astrophysics) (2017)
 - Collab by JPL & MIT
 - 6U CubeSat (like LEONIRD Phase II)
 - Goal: achieve arcsecond-level line-of-sight pointing error and highly stable focal plane temperature control, necessary for precision photometry
 - Led by Sara Seager (MIT Prof)
 - Used by Georgia Tech in their GT1 CubeSat
- References
 - <https://spaceq.ca/open-source-flight-software-called-f-prime-used-on-nasas-ingenuity-mars-helicopter/>
 - <https://ai.jpl.nasa.gov/public/projects/asteria/>
 - [Extended Mission Technology Demonstrations Using the ASTERIA Spacecraft](#)