A picture containing text, crossword puzzle

Description automatically generated

|  |
| --- |
| **Overview** |

NASA has been developing an open source flight software project with ground control for several years on GitHub with the intention to produce satellites more quickly and cost-effectively. They have successfully implemented their work on the raspberry pi for use on small satellites and cube satellites. In this project, we will attempt to improve on the current program by testing it on the Xilinx Pynq FPGA using Vivado. The process is split into two main parts. The first part involves working with the program itself on GitHub while the later stage will involve programming, testing, and receiving data from the FPGA. In this project, we hope to be able to improve the F-Prime usability to a level which will popularize the work so that a hobbyist or student may be able to implement the program in their own work. Our goal for this project is to help NASA/JPL to be able to run the F-Prime flight software on a wide variety of hardware and create a guide for others who want to run F-Prime on the Pynq Z1 FPGA board.

|  |
| --- |
| **About F-Prime and F-Prime Components** |

F-Prime is comprised of:

* An architecture that decomposes flight software into discrete components with well-defined interfaces
* A C++ framework that provides core capabilities such as message queues and threads
* Modeling tools for specifying components and connections and automatically generating code
* A growing collection of ready-to-use components
* Testing tools for testing flight software at the unit and integration levels.

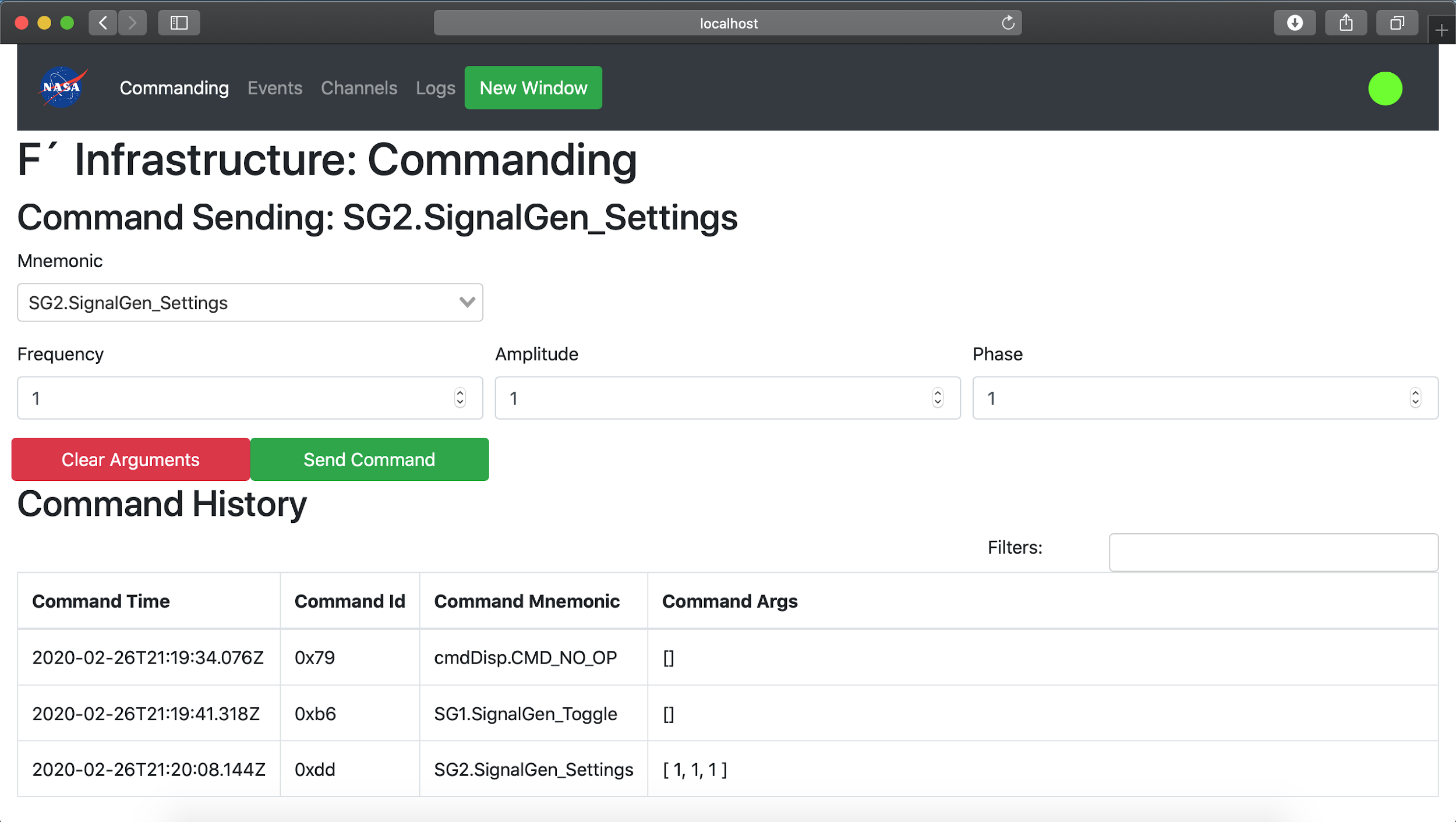
 NASA F-Prime

A circuit board

Description automatically generated

|  |
| --- |
| **Hardware/Software Specifications** |

For the implementation of the project, we are using a Pynq Z1 Field Programmable Gate Array with a ARM Cortex A9 processor. We booted the board with Pynq Linux, which is based on Ubuntu 18.04, using the Pynq image provided by Xilinx. We are also using a computer running Linux or MacOS to run the ground control system and code and to access the terminal on the Pynq Z1 via UART. The ground control system GUI is in HTML, so we are able to send commands to the board using a browser. When originally starting this project, it was necessary to upgrade everything to Python 3 due to Python 2’s end of life this year.



|  |
| --- |
| **Intended Goals** |

The overarching goal of this project is to demonstrate the feasibility of implementing F-Prime onto the Pynq Z1 FPGA board. We will be documenting our work as we go along so that any new users will be able to replicate our process, as well as verify our work. After we have set up F-Prime successfully, we hope to benchmark its

performance on the Pynq Z1 FPGA board against other hardware platforms such as the Raspberry Pi.

|  |
| --- |
| Current State of Affairs |

We have installed the F-Prime software on the Pynq Z1 FPGA board running Pynq Linux and a ground control system on a MacOS computer. We are currently working towards understanding F-Prime flight software better through the tutorials provided on NASA F-Prime Github. The next major milestone in our project is to verify if F-Prime is working as intended. Also, F-Prime comes with many different modules. JPL currently has their GPS module set up and working and our next step will be to test out this

|  |
| --- |
| **Conclusion** |

This project will demonstrate the capabilities of FPGAs through the implementation of F-Prime flight software on it. FPGAs are a proven technology which has shown to be much cheaper to use compared to other hardware options such as application-specific integrated circuits (ASICs). This is due to the fact that FPGAs are reprogrammable so if there is a new update to the F-Prime software, it can be quickly updated with any new changes. Compared to an ASIC chip which will be more difficult to upgrade. Another advantage that FPGAs offer over traditional processors is that it is capable of parallel processing compared to sequential processing, making it a lot faster than traditional processors. Because FPGAs offer a faster solution to certain problems, if we moved some of the processing on the ARM onto the FPGA using hardware description language, we can also reintegrate that specialized process into the ARM using the Zynq processing system and Xilinx AXI interconnect IP. In conclusion, we have demonstrated that it is possible to have flight software on an FPGA board and we hope to demonstrate its feasibility.