

# Control and Monitoring GUI for the ITk-ATLAS MOPS-Hub System in an Educational Context

Alexis F. Espinoza Quinteros

Universidad de Talca

2026 SAPHIR Internships

## Abstract

The ATLAS Inner Tracker (ITk) upgrade requires reliable and scalable control systems for power distribution and detector monitoring. **The Monitoring of Pixel System (MOPS) is a radiation-hardened ASIC for monitoring front-end electronics used in the Inner Tracker.** The MOPS-Hub is a dedicated aggregation and control platform that interfaces multiple MOPS devices, concentrating power monitoring and telemetry data and providing a communication bridge to higher-level control systems. This work presents the design and implementation of a Control and Monitoring Interface for the MOPS-Hub, **combining embedded firmware and a graphical user interface (GUI) to enable monitoring hardware and real-time interactions.** The system integrates Low-level firmware on an Arduino-based controller; a custom binary framed UART protocol; a Python-based GUI for user level interaction, and a mode that facilitates the validation of CAN communication protocol, all of these things allowing on demand manipulation of LDO regulators and ADC monitoring, streamlining the hardware testing process compared to traditional command-line methods.

## Motivation

The MOPS-Hub is responsible for aggregating monitoring data from the pixel sensors to the Detector Control System (DCS)

- The Challenge: **Interacting with the MOPS-Hub in academic context.**
- The Solution: **A dedicated Control and Monitoring Interface was developed to abstract low-level commands, providing a user-friendly environment for hardware verification and debugging.**

During development and integration phases, engineers require:

- Direct access to low-level hardware functions
- Fast diagnostics of power and sensor data
- Flexible control over individual detector channels

**Existing tools are often command-line-based or fragmented, making rapid debugging and system validation difficult.**

**This project addresses this gap by providing a unified control and monitoring interface designed specifically for MOPS-Hub development and testing.**

## System Architecture Overview

### 1. Embedded Firmware (Arduino-based)

Including 16 CIC channels, LDO enable and value configuration, multi-channel ADC acquisition, 3.3 V rail current monitoring (INA232), and CAN communication support. A **command-based CLI** and a **binary framed UART protocol** are implemented for robust interaction.

### 2. Communication Layer

A custom framed UART protocol provides **robust data transfer** and **binary data transmission for ADC and status frames.** A dedicated CAN service mode enables downstream communication testing and validation.

### 3. Graphical User Interface (Python / Tkinter)

Provides real-time visualization and control, **abstracting low-level commands into user-friendly actions.** The interface displays live ADC values, current channel selection, and power status, and supports interactive hardware configuration.



Figure 1: UART frame structure used for sending data.

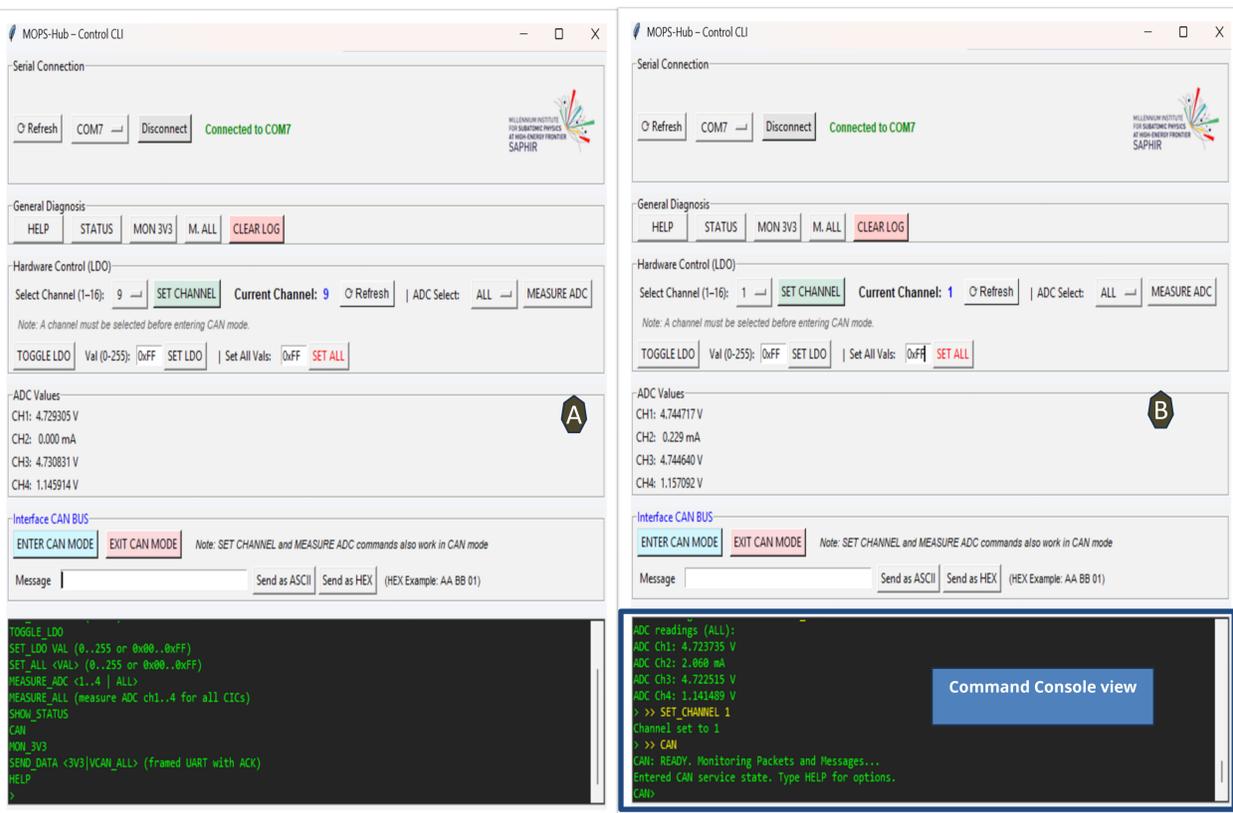


Figure 2: MOPS-Hub Control CLI (GUI) developed. A) Available actions . B) Console and indicators showing ADC measurements and setting the CAN Mode.

## GUI Implementation

A Python-based GUI was developed using Tkinter to provide an intuitive control environment for the MOPS-Hub system. The GUI includes:

- Serial port management and connection monitoring
- **Channel selection and on-demand status display**
- **Live visualization of ADC voltage and current measurements**
- **LDO control and global configuration tools**
- **Integrated CAN bus message interface (ASCII and HEX modes)**
- A threaded design with command queuing prevents race conditions between user actions and hardware responses.

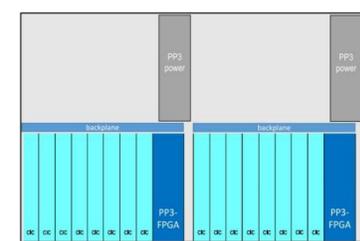


Figure 3: MOPS-Hub crate, CIC power and monitoring channels controlled by the developed interface.

## Results

The system was validated on a MOPS-Hub test setup, demonstrating:

- Reliable control of 16 independent power channels
- **Stable ADC acquisition and On-demand visualization**
- **Robust framed communication without data corruption**
- Seamless transition between normal operation and CAN service mode
- **The interface significantly reduced debugging time during hardware tests and enabled faster identification of power and communication issues.**

## Acknowledgments

The author thanks Matías Henríquez for his assistance and the UNAB-CTEPP team for their support with electronics and mechanics. We also thank Marco Ayala and Sergey Kuleshov, for their constant guidance. We are grateful to SAPHIR for its administrative assistance. This work was supported by ANID FONDECYT grants No. 3230806, No. 1240066, 1240216 and ANID - Millennium Science Initiative Program - ICN2019\_044.