# Plasmetry

## A Plasma Diagnostics and Data Acquisition Solution

Advisor: Dr. Joanne Brenes Catinchi
Electrical & Computer Engineering and Computer Sciences Department
Carlos J. Figueroa (90894) | Alberto J. Cases (123268) | Jahn L. Fortis (116916) | 2024

## Introduction

The Plasmetry Software is a data acquisition solution designed to support and enhance the research efforts of the PUPR Plasma Engineering Laboratory staff, by providing an efficient means of obtaining, processing, and storing measurements from a body of plasma through a highly configurable and modular application.

The system offers a graphical user interface that allows configuration of the software for flexible control of hardware components (Fig. 4). It is designed to interface with six different electrostatic probe implementations (Fig. 3). Our software uses these probes to calculate various plasma parameters that provide insight on the plasma's characteristics and displays these parameters to the user in real-time (Fig. 5). The processed data is programmatically formatted and saved in local storage and in an external cloud storage service (Fig 6).

## Problem

Research at the PUPR Plasma Engineering Laboratory needs the info and accuracy of new probe implementations. The software they depend on is not compatible, and lacks the modularity to adapt. Thus, the lab cannot obtain and store measurements with new probes.

Plasma is hazardous. Researchers need continuous, real-time info; recording data for later analysis is not enough to ensure their safety.

Additionally, the lab notes that operating probes is tedious, and how the circuits are controlled changes based on experiment constraints.

## Solution

Dr. González commissioned us to design software that must interface with existing or designed hardware available to the lab. The software will be deployed to a multi-purpose plasma meter, which is a data acquisition system consisting of an RPIplate CM4 microcomputer and a DAQC2plate for ADC/DAC functionality. The DAQC2plate interfaces the circuits that operate the probes, such as high voltage amplifiers and voltage or current sensors.

This device shall support different probes, including Langmuir Probes and Energy Analyzers.

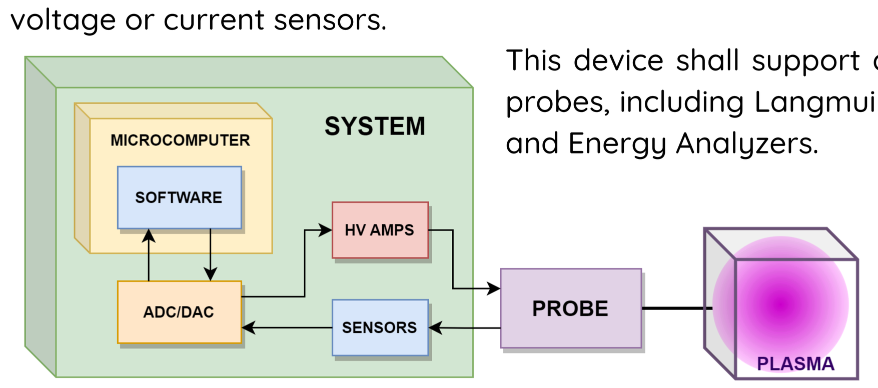


Figure 1: System of Interest

## Background

Plasma, often referred to as the fourth state of matter, is an ionized fluid-like substance. It is defined as a gaseous assembly of electrons, ions, and neutral molecules residing in electric and magnetic fields, where some atoms have had electrons removed from their nuclei. Irving Langmuir coined the term "plasma" in the 20th century to describe this unique state of matter. Plasma research benefits many fields, such as manufacturing, energy production, climate science, and space exploration.

A key phenomenon relevant to our project is the flow of charged particles in plasma. This flow is measured as electric current. Electrostatic probes are inserted into the plasma, and current is measured as a function of applied voltage. This data provides insight about the plasma, such as electron temperature and electron density.

## System Architecture

The Plasmetry Software Architecture consists of four layers (Fig. 2). The fifth layer depicted at the bottom represents the physical hardware components the system must interface with. This architecture pattern is a widely used concept; its major benefit is the modularity it provides by decoupling each layer from the other's implementation.

These software interfaces have been abstracted as four main layers that are responsible of different operations within our system. We've defined the layer interfaces as well as concrete implementations using Python 3.11 and deployed to the RPIplate CM4. These layers are listed as follows:

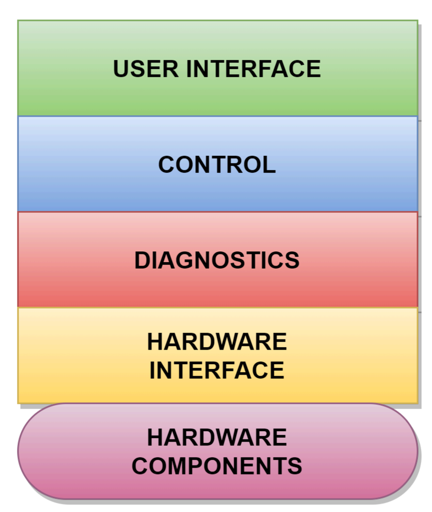


Figure 2: Software Layers

- **Hardware Interface Layer:** Provides software artifacts that model and control the physical circuit elements used to interface with the electrostatic probes.
- **Diagnostics Layer:** Employs instantiated Hardware Interface Layer artifacts to perform plasma diagnostics, including controlling the voltages applied on the probes and acquiring data samples to yield plasma parameters.
- **Control Layer:** This interface is responsible for coordinating all major functionalities and operational aspects of the software. The Control Layer interface provides configuration file and file upload features, Diagnostics Layer initialization and interaction, and method definitions that are invoked by the User Interface Layer.
- **User Interface Layer:** Interface that instantiates all components required to offer the user the means to interact with the system, by mapping to various functions provided by the Control Layer. Primarily, this layer provides added functionality pertaining specifically to user interactions, such as graphic user interface display and overall logic needed to interpret user input.

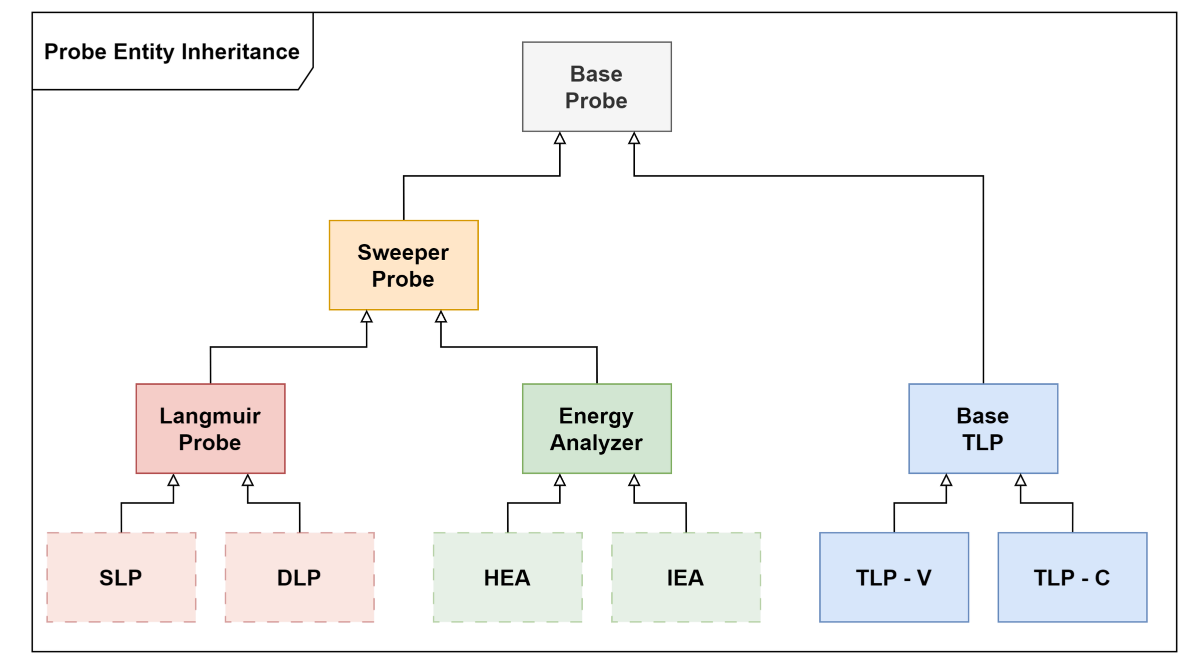


Figure 3: Probe Classes

There are five types of probes Plasmetry supports: Single Langmuir Probes (SLP), Double Langmuir Probes (DLP), Hyperbolic Energy Analyzers (HEA), Ion Energy Analyzers (IEA), and Triple Langmuir Probes (TLP). We treat the two TLP operating modes as distinct entities, arriving at six physical probe implementations.

From the software's perspective, probes are abstracted as inputs and outputs for voltage signals. Three abstract classes group common elements: Base Probe, Sweeper Probe, and Base TLP. Four concrete subclasses implement data acquisition: Langmuir Probe, Energy Analyzer, TLP - V, and TLP - C.

Since SLPs and DLPs are operated the same way, (likewise HEA and IEA), one class can model both. The difference is the equations to calculate plasma parameters. Software factories assemble probe objects with the needed components and equations, using the four concrete subclasses.

## Software Interfaces

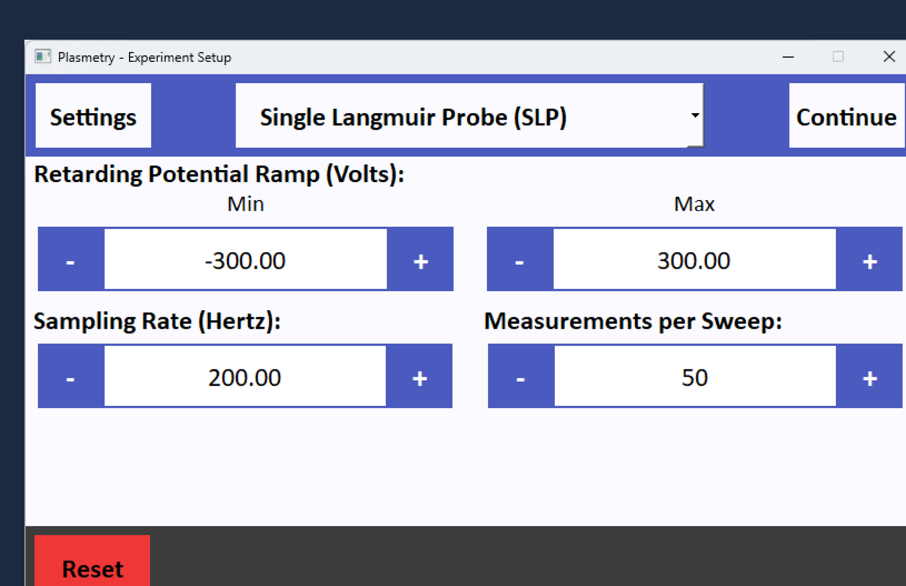


Figure 4: Application's Experiment Setup Window

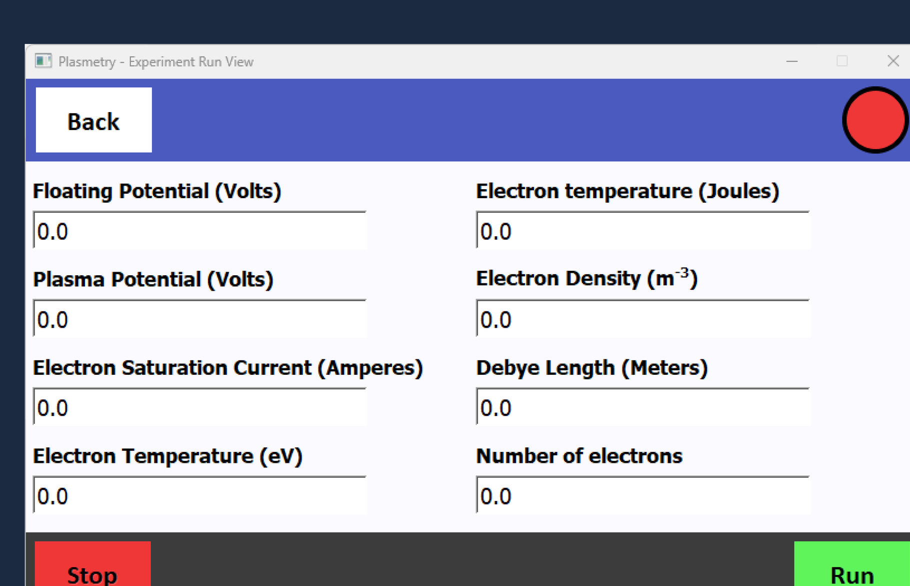


Figure 5: Application's Experiment Run Window

The Plasmetry system's software interfaces, written in Python, provide users control of the probe hardware, parameter calculations & display, data acquisition, and data management. It also features JSON file interfacing that allows the contents of a configuration file to be easily modified by the user (Fig. 4), data formatting of acquired parameters or probe data, and file storage, both local and through the Google Drive API (Fig. 6).

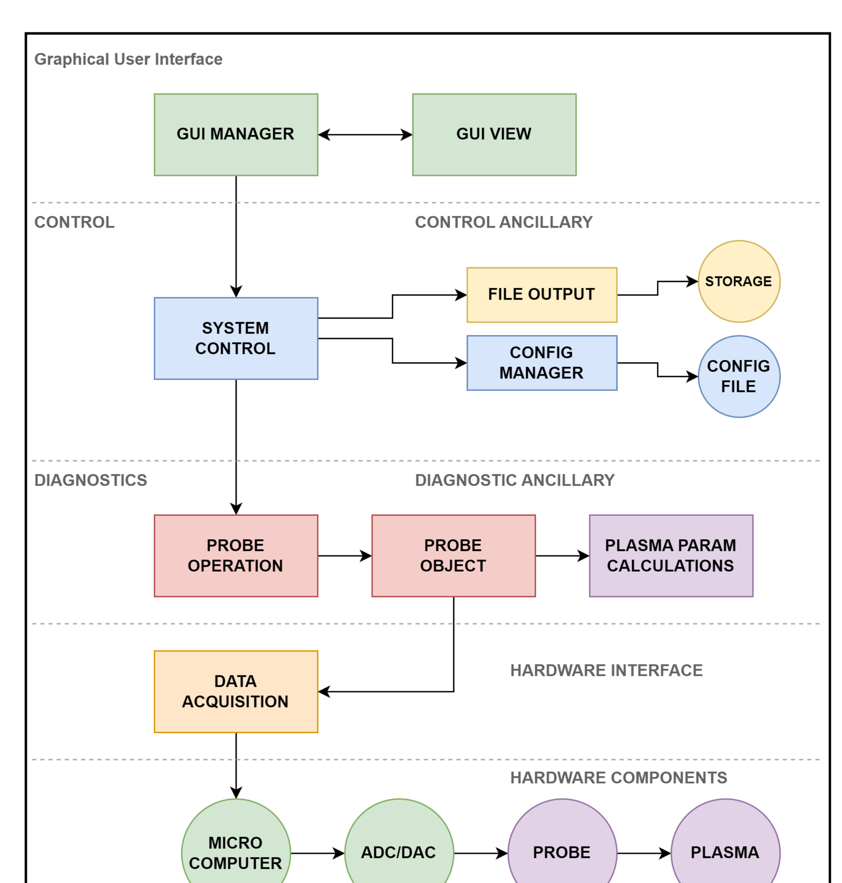


Figure 7: Major Components

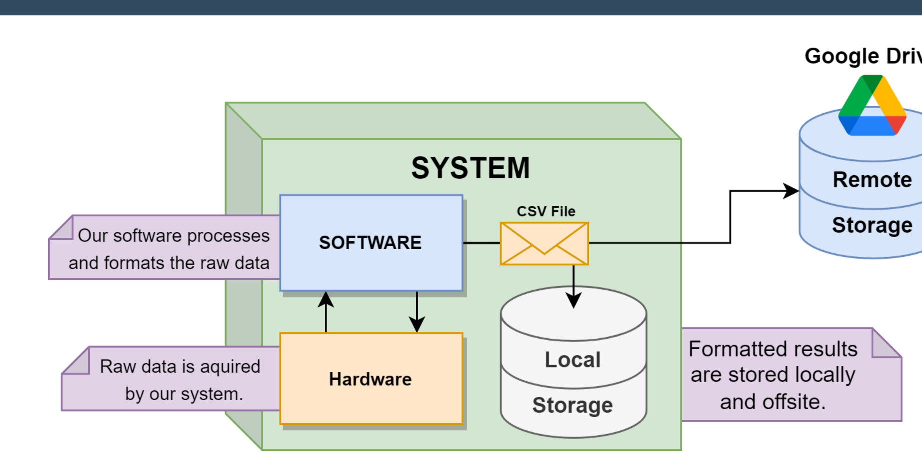


Figure 6: Data Storage

The diagram on the left (Fig. 7) is an abstract conceptualization of the major components in our software architecture, and how they relate to each other.

- **GUI View:** Defines visual elements of the graphical user interface using PyQt5.
- **GUI Manager:** Logic object to interface with System Control.
- **System Control:** Coordinates high level processes.
- **File Output:** Formats, stores, and uploads results.
- **Config Manager:** Reads and writes system and user settings to a JSON configuration file.
- **Probe Operation:** Manages Probe Objects; calculates plasma parameters to update, display, and aggregate them.
- **Probe Object:** Performs data acquisition and stores probe specific equations.
- **Parameter Calculations:** Reusable equations and processing functions.
- **Data Acquisition:** Interfaces software with physical probe hardware.

## Conclusion

The Plasmetry Software meets its objectives by providing a robust, adaptable, and highly configurable tool for plasma diagnostics in the PUPR Plasma Engineering Laboratory. The system offers features such as a graphical user interface, hardware control, continuous data acquisition, parameter calculations, and proper data management.

The developed product meets the specified needs and requirements of the PUPR Plasma Engineering Laboratory while focusing on modularity and support for multiple probe types and configurations. This flexibility aids researchers to easily configure experiments, control hardware components, and visualize plasma parameters in real-time (Fig. 5).

The solution presented not only fulfills the specified requirements, but also lays the foundations for future expansions. The software has been designed so that it may easily adapt for evolving needs, enhancing the research capabilities Plasmetry provides.

## Future Work

Our work developing this project was extremely satisfying. Though its requirements are met, there is work to be done. We hope to have the opportunity to continue improving the Plasmetry Software and collaborate with future developers after we graduate.

Error handling must be improved. For example, some errors are not shown in the GUI, only in the console or logs.

Remote operation is desirable, but never as a requirement due to our deadlines. A client/server layer may be inserted between existing ones.

Plasmetry's sampling rate is limited by the DAQC2plate's SPI protocol. A second plate in the stack could operate as a waveform generator, and the other as an oscilloscope, surpassing the current 500 Hz limit.

Ultimately, support for new probes and circuit elements will always be needed, thus, Plasmetry's development should be a continuous effort.

## Bibliography


- I. H. Hutchinson, Principles of plasma diagnostics. Cambridge University Press, 2002.
- F. F. Chen, Introduction to plasma physics and controlled fusion. Cham: Springer International Publishing, 2016.
- K. A. Polzin, E. Blumhagen, A. C. Sherrod, and T. Moeller, "Behavior of Triple Langmuir Probes in Non-equilibrium Plasmas," presented at the AIAA Propulsion and Energy 2019 Forum, Reston, Virginia, Aug. 2019
- E. Leal-Quiros and M. A. Prelas, "A hyperbolic energy analyzer," Rev. Sci. Instrum., vol. 61, no. 6, pp. 1708–1712, Jun. 1990, doi: 10.1063/1.1141136.
- E. Leal-Quirós and Á. González, "Basic Plasma Diagnostics: Probes and Analyzers," May 2008. http://hdl.handle.net/20.500.12475/1606
- WallyWare Inc., "DAQC2plate Users Guide - Pi-Plates," Pi-Plates. https://pi-plates.com/daqc2-users-guide/ (accessed Sept. 29, 2024).
- Google LLC, "Google Drive API overview | Google for Developers", Google Workspace. https://developers.google.com/drive/api/reference/rest/v3
- Riverbank Computing Ltd, "PyQt5 Reference Guide — PyQt Documentation v5.15.7", PyQt5. https://www.riverbankcomputing.com/static/Docs/PyQt5/


## Acknowledgements

We'd like to thank Dr. González for providing us the opportunity and challenge of working with the PUPR Plasma Engineering Laboratory. Our thanks also extend to our advisor, Dr. Brenes for her support throughout our endeavor. And ultimately, we thank our loved ones, who've given us nothing but patience and support during this challenging undertaking.