

Objective

Create a cheap, modular FPGA data acquisition and control system using commercial off-the-shelf parts.

Value Proposition

Traditional data acquisition systems are very expensive due to their degree of specialization. An affordable and flexible solution is desirable for companies like ThorLabs that are looking for a data acquisition and control solution.

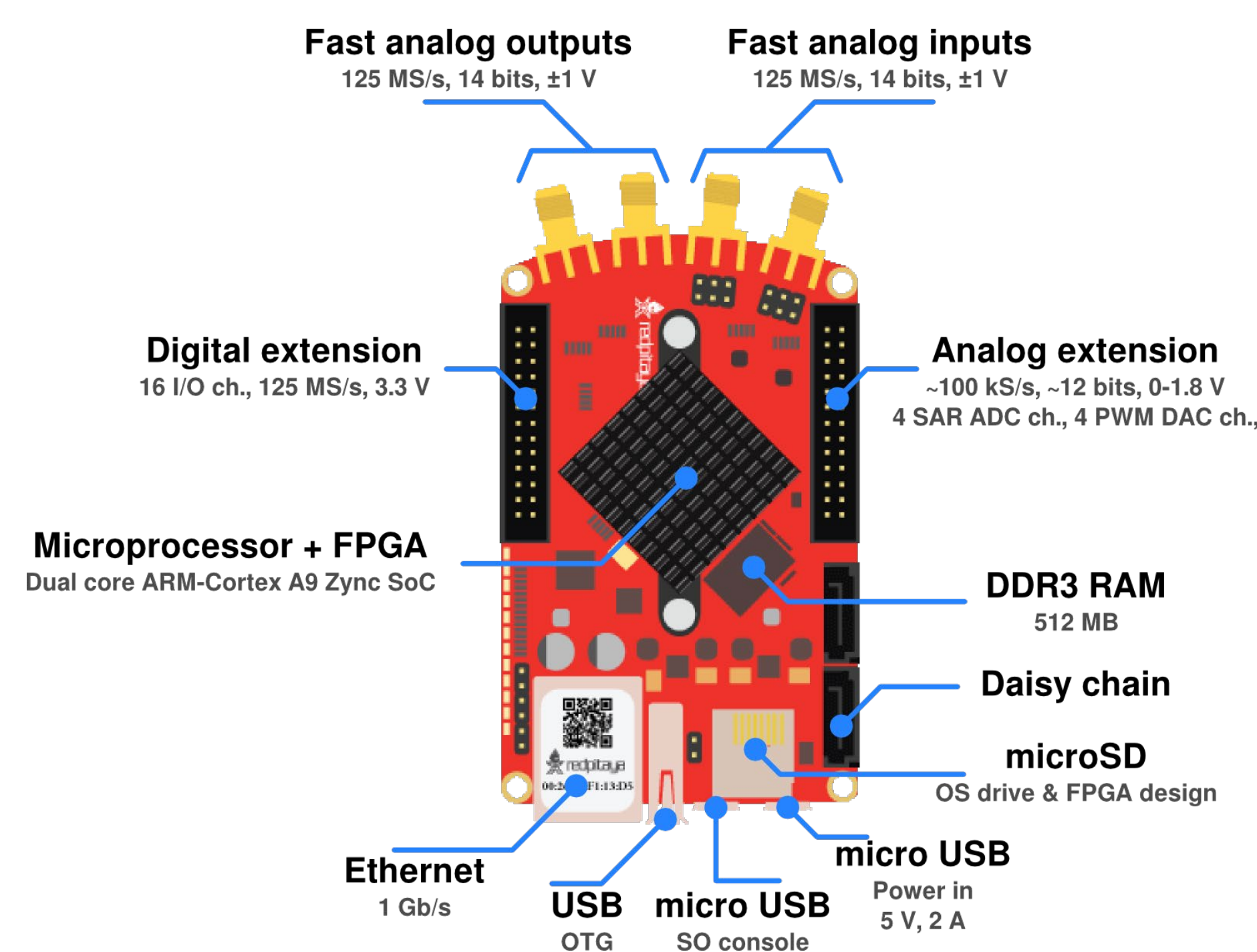
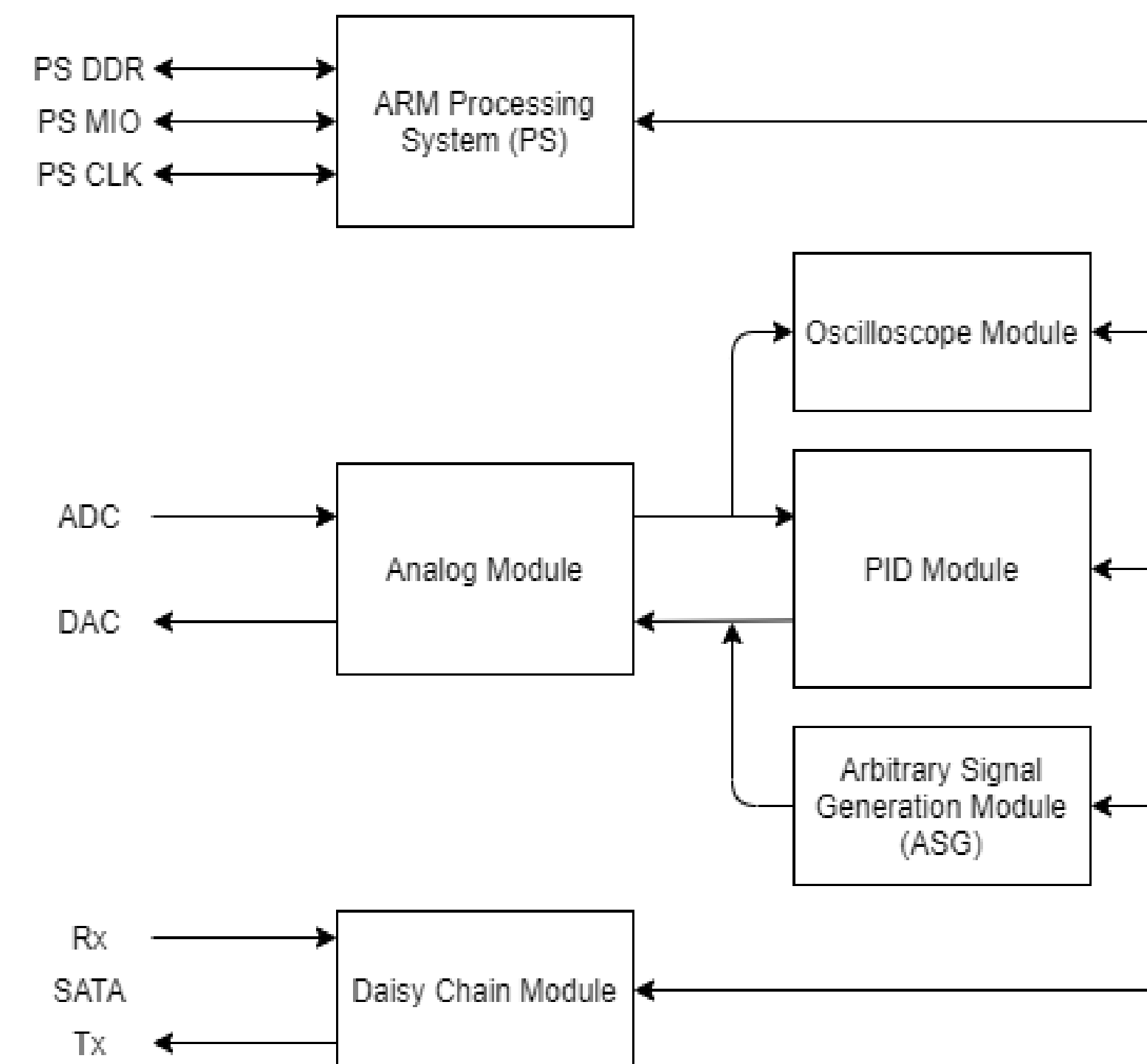


Diagram of Red Pitaya development board.

Background

- An FPGA (Field Programmable Gate Array) is a piece of hardware with configurable digital logic
- Red Pitaya utilizes the Zynq 7000 SoC which combines an FPGA and an ARM processor that is Linux capable
- Board also equipped with ADCs and DACs, allowing for analog input and output capabilities



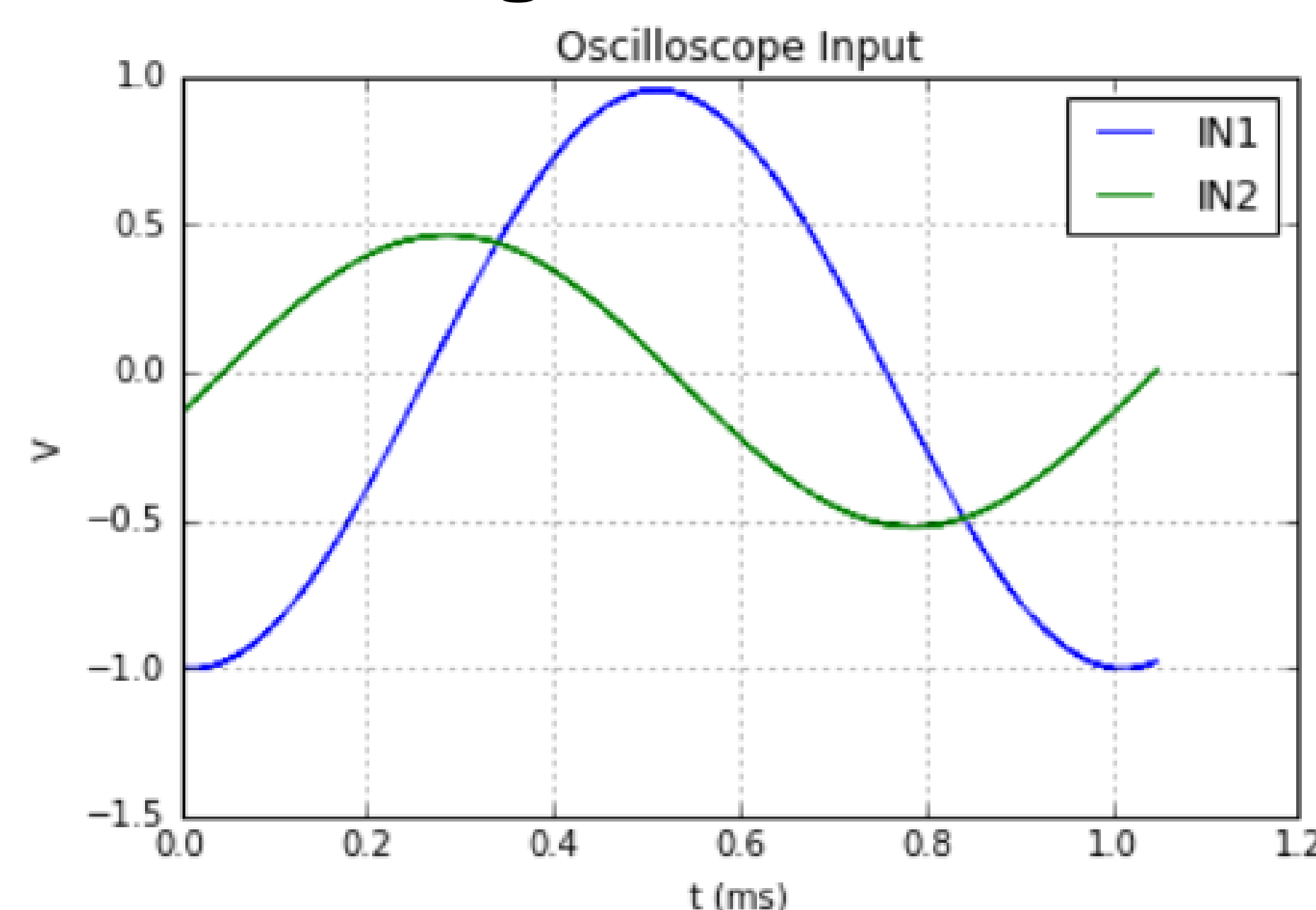
A block diagram of the Red Pitaya FPGA system.

Key Requirements

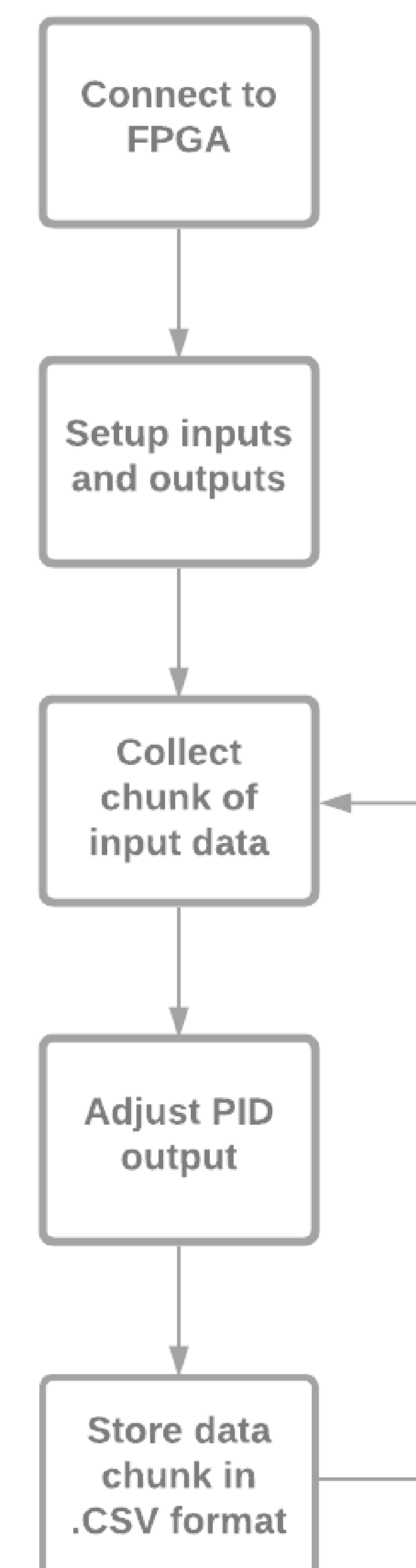
- Tier 1 – Dual-channel data acquisition, decimation, filtering, and data logging
- Tier 2 – PID Control loop
- Tier 3 – Quad-channel data acquisition

Design

- Implemented through a Python script using Jupyter Notebook
- Red Pitaya Python SDK interacts with Analog Module in FPGA

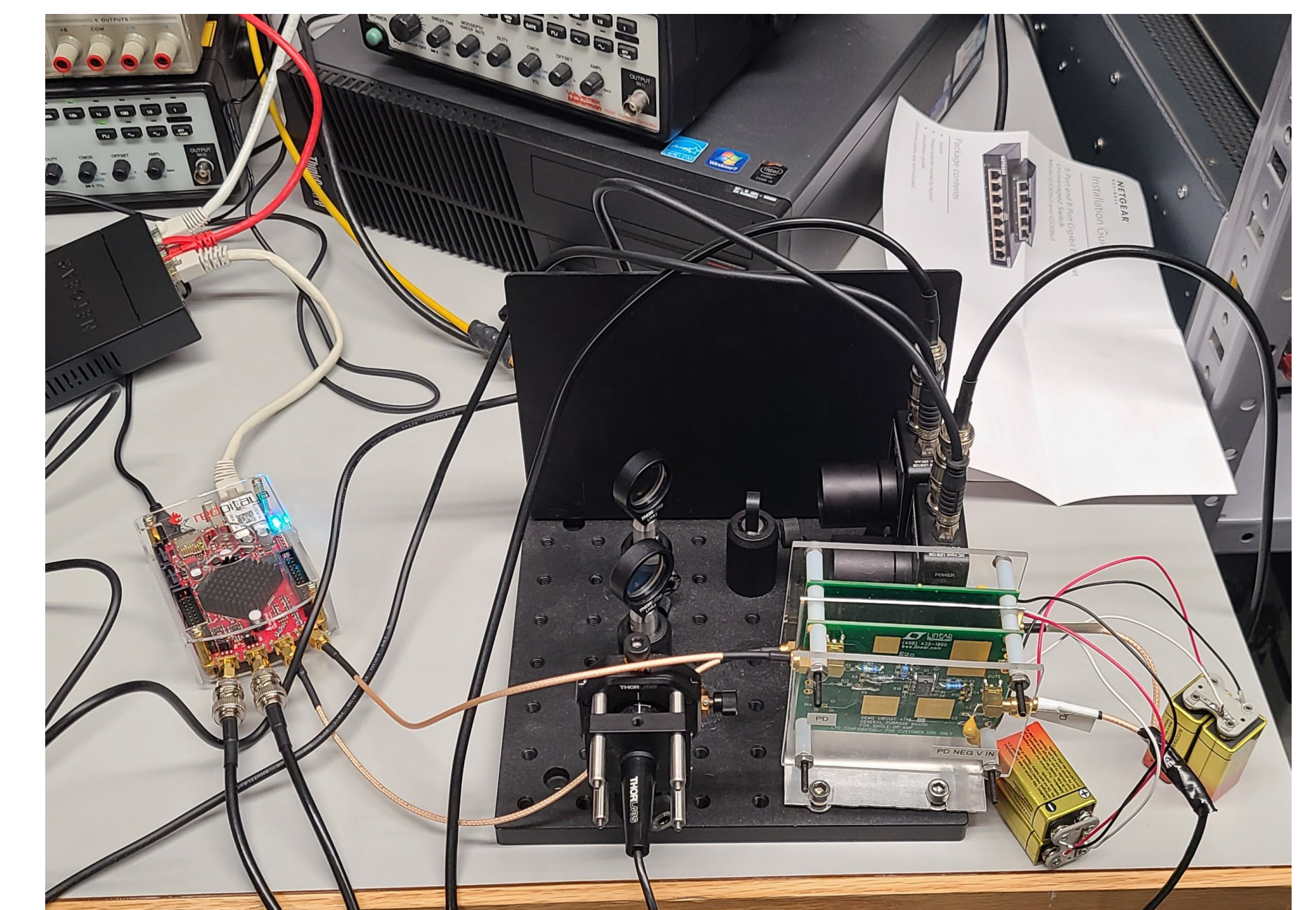


Data recorded by the ADCs during testing (left) and a diagram of system functionality (right).



Validation

- Data acquisition validated by capturing known input data through the oscilloscope
- Due to issues with laser diode from Thorlabs, validated PID loop being using a "Hardware-in-the-loop" simulation, substituting the laser setup with a second Red Pitaya



Laser system provided by ThorLabs for testing.

Conclusion & Recommendations

- Our work satisfies two tiers of our requirements
- Red Pitaya's Python has limited capabilities in interacting with the FPGA system
- The Red Pitaya C SDK is more robust and feature rich, so future work may benefit by switching to C instead of using Python

Acknowledgements

Sponsor: Dr. Chris Manning
Mentors: Dr. James Frenzel
Dr. Feng Li