pyadps v0.2.0 (Amol Prakash, 2025)

pyadps is a Python package that can be used to process moored ADCP data. It provides data reading, quality control, NetCDF file creation and visualization functionalities in a graphical interface (Streamlit), as well as direct Python package access. The software is designed for Teledyne RDI workhorse ADCPs, and other company’s ADCP files are not compatible. It was last updated on 17/06/2025 and is actively maintained.

* [Homepage & Documentation](https://pyadps.readthedocs.io/en/latest/index.html)
* [GitHub Repository](https://github.com/p-amol/pyadps)

Usage:

* Launch an anaconda prompt terminal and activate the environment for pyadps: conda activate pyadps\_env (has to be python 3.12)
* Install pyadps in the pyadps\_env using pip install pyadps
* Run the interactive interface from the terminal: run-pyadps
* Click the ReadFile button in the left-side panel to upload your file. Follow the steps outlined in the left panel menu sequentially to process the data.
* For processing a new dataset, restart the browser using the run-pyadps command. This ensures no residual data from the previous file remains in the web cache.
* When done, deactivate the environment in the terminal: conda deactivate

Notes:

I am working on the GUI. Uploading a raw data file (.000) gives the error: IndexError: list index out of range. The error is coming from the line in readrdi.py where data\_id\_array = self.dataid[ens]. readrdi.py is trying to access an element of self.dataid at index ens, but that index does not exist. The issue is that the instrument is burst averaging. The instrument likely recorded short burst of raw velocity profiles and then averaged them into one ensemble and then stored those as averaged ensembles.

I am waiting to see if I can get access to WinADCP from Teledyne Marine. If yes, then I might be able to convert the averaged file back to a PD0 so that I can try and process it with pyadps. If not, then I move on to the next software, but I am likely to encounter the same problem with. If this is the case, I need to find a way to write a custom parser. I might also be able to modify the source code of readrdi.py to accept my data file format.