

---

# hcam-drivers Documentation

*Release 1.2.2*

**Stuart Littlefair**

**Mar 09, 2022**



---

# Contents

---

<b>1 HiperCAM Python Driver</b>	<b>3</b>
1.1 Installation . . . . .	3
<b>2 Detailed Installation</b>	<b>5</b>
2.1 Stable release . . . . .	5
2.2 From sources . . . . .	5
<b>3 Usage</b>	<b>7</b>
<b>4 Contributing</b>	<b>9</b>
4.1 Types of Contributions . . . . .	9
4.2 Get Started! . . . . .	10
4.3 Pull Request Guidelines . . . . .	11
4.4 Tips . . . . .	11
<b>5 Indices and tables</b>	<b>13</b>



Contents:



---

## HiperCAM Python Driver

---

`hcam_drivers` provides Python tools for interfacing with the HiperCAM high-speed camera. `hcam_drivers` is written in Python and is based on TKinter. It should be compatible with Python2 and Python3.

### 1.1 Installation

The software is written as much as possible to make use of core Python components. The third-party requirements are:

- My own `hcam_widgets` package;
- `astropy`, a package for astronomical calculations;
- `twisted`, a package for asynchronous programming;
- `autobahn`, a package to use the Web Application Messaging Protocol (WAMP);
- `tornado`, used for file server which allows remote access to observing data;
- `pyaml` and `configobj` for loading config files and
- `pymodbus` (Python 2) or `pymodbus3` (Python 3) for talking to the flow-rate monitor.

Usually, installing with `pip` will handle these dependencies for you, so installation is a simple matter of typing:

```
pip install hcam_drivers
```

or if you don't have root access:

```
pip install --prefix=my_own_installation_directory hcam_drivers
```

For more information, see:

- [The documentation](#)
- [This packages' Github code repository](#)





## 2.1 Stable release

To install `hcam-drivers`, run this command in your terminal:

```
$ pip install hcam-drivers
```

This is the preferred method to install `hcam-drivers`, as it will always install the most recent stable release.

If you don't have `pip` installed, this [Python installation guide](#) can guide you through the process.

## 2.2 From sources

The sources for `hcam-drivers` can be downloaded from the [Github repo](#).

You can either clone the public repository:

```
$ git clone git://github.com/HiPERCAM/hcam-drivers
```

Or download the [tarball](#):

```
$ curl -OL https://github.com/HiPERCAM/hcam-drivers/tarball/master
```

Once you have a copy of the source, you can install it with:

```
$ python setup.py install
```



## CHAPTER 3

---

### Usage

---

To use hcam-drivers in a project:

```
import hcam_drivers
```



Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given. You can contribute in many ways:

## 4.1 Types of Contributions

### 4.1.1 Report Bugs

Report bugs at <https://github.com/HiPERCAM/hcam-drivers/issues>.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

### 4.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with “bug” and “help wanted” is open to whoever wants to implement it.

### 4.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with “enhancement” and “help wanted” is open to whoever wants to implement it.

## 4.1.4 Write Documentation

hcam-drivers could always use more documentation, whether as part of the official hcam-drivers docs, in docstrings, or even on the web in blog posts, articles, and such.

## 4.1.5 Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/HiPERCAM/hcam-drivers/issues>.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

## 4.2 Get Started!

Ready to contribute? Here's how to set up *hcam-drivers* for local development.

1. Fork the *hcam-drivers* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/hcam-drivers.git
```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

```
$ mkvirtualenv hcam-drivers
$ cd hcam-drivers/
$ python setup.py develop
```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

```
$ flake8 hcam-drivers tests
$ python setup.py test or py.test
$ tox
```

To get flake8 and tox, just pip install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

## 4.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 2.6, 2.7, 3.3, 3.4 and 3.5, and for PyPy. Check [https://travis-ci.org/HiPERCAM/hcam-drivers/pull\\_requests](https://travis-ci.org/HiPERCAM/hcam-drivers/pull_requests) and make sure that the tests pass for all supported Python versions.

## 4.4 Tips

To run a subset of tests:

```
$ py.test tests.test_hcam-drivers
```





## CHAPTER 5

---

### Indices and tables

---

- genindex
- modindex
- search