UCI cBP demo Documentation

Release 0.4.0

UCI cBP demo

Apr 16, 2020

Contents:

CHAPTER 1

UCI cBP demo

GUI to demo continuous blood pressure sensing works on Linux, Windows, and MacOS X. This GUI requires custom firmware installed on MbientLab Metamotion R device, and a capacitor to digital converter from Analog Devices, AD7746.

Pull requests welcome! Please fork repository to begin with.

- Free software: MIT license
- Documentation: https://uci-bp-demo.readthedocs.io/en/latest/index.html.

1.1 Quick Start

Following command assumes a Linux environment. For Windows and MacOSX setup, you may need to tweak the commands a little bit, according to your system setup.

```
# setup virtual environment
python -m venv venv

# enter virtual environment
source venv/bin/activate

# install the latest code from PyPI
pip install uci-cbp-demo

# power up the hardware

# start GUI with parameters a=1 b=0
uci_cbp_demo gui -a 1 -b 0
```

To list available CLI options, use

```
uci_cbp_demo gui --help
# Usage: uci_cbp_demo gui [OPTIONS]
#
# Options:
# -a INTEGER   Scaling coefficient
# -b INTEGER   Shifting in Y
# --help       Show this message and exit.
```

1.2 Troubleshooting

1. If you run into issues saying

   > Could not fetch URL https://pypi.python.org/ ... There was a problem confirming the ssl certificate: [SSL: TLSV1_ALERT_PROTOCOL_VERSION] tlsv1 alert protocol version (_ssl.c:645) - skipping

   Try follow steps described in pypa repository

```
curl https://bootstrap.pypa.io/get-pip.py | python
```

1.3 MAC OS Notes

1. OS X/macOS support via Core Bluetooth API, from at least version 10.11

2. The macOS backend of Bleak is written with pyobjc directives for interfacing with Foundation and CoreBluetooth APIs. There are some values that pyobjc is not able to overwrite and thuse the corebleak framework was written to circumvent these issues. The most noticible difference between the other backends of bleak and this backend, is that CoreBluetooth doesn't scan for other devices via MAC address. Instead, UUIDs are utilized that are often unique between the device that is scanning the the device that is being scanned.

Installation

2.1 Stable release

To install UCI cBP demo, run this command in your terminal:

```
$ pip install uci_cbp_demo
```

This is the preferred method to install UCI cBP demo, 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 UCI cBP demo can be downloaded from the Github repo.

You can either clone the public repository:

```
$ git clone git://github.com/taoyilee/uci_cbp_demo
```

Or download the tarball:

```
$ curl -OJL https://github.com/taoyilee/uci_cbp_demo/tarball/master
```

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

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

Usage

To use UCI cBP demo in a project:

```python
import uci_cbp_demo
```

CHAPTER 4

Contributing

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/taoyilee/uci_cbp_demo/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

UCI cBP demo could always use more documentation, whether as part of the official UCI cBP demo 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/taoyilee/uci_cbp_demo/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 *uci_cbp_demo* for local development.

1. Fork the *uci_cbp_demo* repo on GitHub.

2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/uci_cbp_demo.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 uci_cbp_demo
$ cd uci_cbp_demo/
$ 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 uci_cbp_demo tests
$ python setup.py test or pytest
$ 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 3.5, 3.6, 3.7 and 3.8, and for PyPy. Check https://travis-ci.com/taoyilee/ uci_cbp_demo/pull_requests and make sure that the tests pass for all supported Python versions.

4.4 Tips

To run a subset of tests:

```
$ pytest tests.test_uci_cbp_demo
```

4.5 Deploying

A reminder for the maintainers on how to deploy. Make sure all your changes are committed (including an entry in HISTORY.rst). Then run:

```
$ bump2version patch # possible: major / minor / patch
$ git push
$ git push --tags
```

Travis will then deploy to PyPI if tests pass.

Credits

5.1 Development Lead

- UCI cBP demo <taoyil@uci.edu>

5.2 Contributors

None yet. Why not be the first?

History

6.1 0.1.0 (2020-02-17)

- First release on PyPI.

6.2 0.1.2 (2020-02-17)

- Fix typo and restrict support to Python 3.8

6.3 0.1.3 (2020-02-17)

- Add device scanner and MacOSX support

6.4 0.2 (2020-02-17)

- Support 2 channel operation and new board

Indices and tables

- genindex
- modindex
- search