
Timeflux Plux

Release 0.3.dev2+gb918e91

Pierre Clisson

Feb 16, 2022

CONTENTS

1	Installation	3
2	Documentation	5
2.1	API Reference	5
	Python Module Index	9
	Index	11

This plugin provides a driver to connect to [Plux](#) devices.

When connecting to the device for the first time, use the following pairing code: *123*.

INSTALLATION

First, make sure that [Timeflux](#) is installed.

You can then install this plugin in the *timeflux* environment:

```
$ conda activate timeflux  
$ pip install timeflux_plux
```


DOCUMENTATION

The API documentation and examples can be found [here](#).

2.1 API Reference

This page contains auto-generated API reference documentation.

timeflux_plux

2.1.1 timeflux_plux

timeflux_plux.helpers

helpers

timeflux_plux.helpers.transfer

Transfer functions

transfer

`timeflux_plux.helpers.transfer.RESOLUTION = 16`

`timeflux_plux.helpers.transfer.VCC = 3`

`timeflux_plux.helpers.transfer.ECG(signal)`
ECG value in millivolt ()

`timeflux_plux.helpers.transfer.BVP(signal)`
BVP value in r.i. units

`timeflux_plux.helpers.transfer.EDA(signal)`
EDA value in microsiemens ()

`timeflux_plux.helpers.transfer.EMG(signal)`
EMG value in millivolt ()

`timeflux_plux.helpers.transfer.PZT(signal)`
Displacement value in percentage (%) of full scale

`timeflux_plux.helpers.transfer.EEG(signal)`
EEG value in microvolt ()

`timeflux_plux.helpers.transfer.LUX(signal)`
LUX normalized value (0-1)

`timeflux_plux.nodes`

nodes

`timeflux_plux.nodes.driver`

driver

class `timeflux_plux.nodes.driver.Plux(address=None, rate=None)`

Bases: `timeflux.core.node.Node`

This node connects to a BiosignalsPlux device and streams data at a provided rate.

Two output streams are provided. The default output is the data read from the analog and digital channels, converted to meaningful units according to the sensor types. The `o_raw` output provides the data directly returned from the device.

Parameters

- **port** (*string/None*) – Path to the Plux device. e.g. `xx:xx:xx:xx:xx:xx` (Bluetooth Mac Address), `COMx` (serial port on Windows), `/dev/cu.biosignalsplux-Bluetooth` (serial port on macOS). If not specified, the node will connect to the first detected device.
- **rate** (*int/None*) – The device rate in Hz. Maximum value for one channel: `8000`. Maximum value for eight channels: `2000`. If not specified, the rate will be set to the maximum value allowed for the number of detected sensors.

Variables

- **i** (*Port*) – Default input, expects DataFrame.
- **o** (*Port*) – Signal converted to meaningful units, provides DataFrame.
- **o_raw** (*Port*) – Raw signal, provides DataFrame.

Example

```
graphs:
- id: acquisition
  nodes:
  - id: plux
    module: timeflux_plux.nodes.driver
    class: Plux
```

(continues on next page)

(continued from previous page)

```

    params:
      address: /dev/cu.biosignalsplux-Bluetooth
      rate: 1000
- id: pub_raw
  module: timeflux.nodes.zmq
  class: Pub
  params:
    topic: raw
- id: pub_converted
  module: timeflux.nodes.zmq
  class: Pub
  params:
    topic: converted
edges:
- source: plux:raw
  target: pub_raw
- source: plux
  target: pub_converted
rate: 1

- id: display
  nodes:
    - id: subscribe
      module: timeflux.nodes.zmq
      class: Sub
      params:
        topics: [ converted ]
    - id: debug
      module: timeflux.nodes.debug
      class: Display
  edges:
    - source: subscribe:converted
      target: debug
  rate: 1

- id: broker
  nodes:
    - id: broker
      module: timeflux.nodes.zmq
      class: Broker

```

Attention:

- On macOS, device autodetection and MAC addresses seem to work, but data is not actually streamed. Use the serial port instead.
- Multiple sensors of the same type are currently not supported.
- For sensors that return multiple channels (accelerator for example), only the first channel is available.

See also:

- Official (outdated) API documentation
- Official libraries
- Discussion about sensor detection mapping
- Helpful examples on how to write transfer functions

Instantiate the node.

update(*self*)

Update outputs

terminate(*self*)

Cleanup

info(*self*)

Get some info about the connected device

convert(*self*, *samples*)

Convert signal to meaningful units

PYTHON MODULE INDEX

t

- `timeflux_plux`, 5
- `timeflux_plux.helpers`, 5
- `timeflux_plux.helpers.transfer`, 5
- `timeflux_plux.nodes`, 6
- `timeflux_plux.nodes.driver`, 6

INDEX

B

BVP() (in module *timeflux_plux.helpers.transfer*), 5

C

convert() (*timeflux_plux.nodes.driver.Plux* method), 8

E

ECG() (in module *timeflux_plux.helpers.transfer*), 5

EDA() (in module *timeflux_plux.helpers.transfer*), 5

EEG() (in module *timeflux_plux.helpers.transfer*), 6

EMG() (in module *timeflux_plux.helpers.transfer*), 5

I

info() (*timeflux_plux.nodes.driver.Plux* method), 8

L

LUX() (in module *timeflux_plux.helpers.transfer*), 6

M

module

timeflux_plux, 5

timeflux_plux.helpers, 5

timeflux_plux.helpers.transfer, 5

timeflux_plux.nodes, 6

timeflux_plux.nodes.driver, 6

P

Plux (class in *timeflux_plux.nodes.driver*), 6

PZT() (in module *timeflux_plux.helpers.transfer*), 5

R

RESOLUTION (in module *timeflux_plux.helpers.transfer*),
5

T

terminate() (*timeflux_plux.nodes.driver.Plux* method),
8

timeflux_plux

 module, 5

timeflux_plux.helpers

 module, 5

timeflux_plux.helpers.transfer

 module, 5

timeflux_plux.nodes

 module, 6

timeflux_plux.nodes.driver

 module, 6

U

update() (*timeflux_plux.nodes.driver.Plux* method), 8

V

VCC (in module *timeflux_plux.helpers.transfer*), 5