
Timeflux BITalino

Release 0.2.0

Pierre Clisson

May 18, 2020

CONTENTS

| | |
|-----------------------------|----------|
| 1 Installation | 3 |
| 1.1 API Reference | 3 |
| Python Module Index | 7 |
| Index | 9 |

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

When connecting a BITalino device for the first time, the default pairing code is: *1234*.

INSTALLATION

First, make sure that `Timeflux` is installed.

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

```
$ conda activate timeflux
$ pip install timeflux_bitalino
```

1.1 API Reference

This page contains auto-generated API reference documentation.

`timeflux_bitalino`

1.1.1 `timeflux_bitalino`

`timeflux_bitalino.nodes`

nodes

`timeflux_bitalino.nodes.driver`

driver

class `timeflux_bitalino.nodes.driver.BitAlino` (*port*, *rate=1000*, *channels='A1', 'A2', 'A3', 'A4', 'A5', 'A6'*)

Bases: `timeflux.core.node.Node`

BITalino driver.

This node connects to a BITalino device and streams data at a provided rate. It is based on the original BITalino Python library, with some performance improvements and careful timestamping.

Two output streams are provided. The default output is the data read from the analog and digital channels. The `o_offsets` output provides continuous offsets between the local time and the estimated device time. This enables drift correction to be performed during post-processing, although no significant drift has been observed during testing.

Variables

- `o` (*Port*) – BITalino data, provides DataFrame.
- `o_offsets` (*Port*) – Time offsets, provide DataFrame.

Parameters

- **port** (*string*) – The serial port. e.g. COM3 on Windows; /dev/tty.bitalino-DevB on MacOS; /dev/ttyUSB0 on GNU/Linux.
- **rate** (*int*) – The device rate in Hz. Possible values: 1, 10, 100, 1000. Default: 1000.
- **channels** (*tupple*) – The analog channels to read from. Default: ('A1', 'A2', 'A3', 'A4', 'A5', 'A6').

Example

```
graphs:

- id: acquisition
  nodes:
  - id: bitalino
    module: timeflux_bitalino.nodes.driver
    class: Bitalino
    params:
      port: /dev/tty.BITalino-02-44-DevB
      #port: /dev/tty.BITalino-03-87-DevB
      rate: 1000
  - id: pub_bitalino
    module: timeflux.nodes.zmq
    class: Pub
    params:
      topic: bitalino
  - id: pub_offsets
    module: timeflux.nodes.zmq
    class: Pub
    params:
      topic: offsets
  edges:
  - source: bitalino
    target: pub_bitalino
  - source: bitalino:offsets
    target: pub_offsets
  rate: 30

- id: display
  nodes:
  - id: subscribe
    module: timeflux.nodes.zmq
    class: Sub
    params:
      topics: [ bitalino, offsets ]
  - id: debug
    module: timeflux.nodes.debug
    class: Display
  edges:
  - source: subscribe:bitalino
    target: debug
```

(continues on next page)

(continued from previous page)

```
rate: 1

# - id: record
#   nodes:
#     - id: sub
#       module: timeflux.nodes.zmq
#       class: Sub
#       params:
#         topics: [ bitalino, offsets ]
#     - id: save
#       module: timeflux.nodes.hdf5
#       class: Save
#   edges:
#     - source: sub:bitalino
#       target: save:bitalino
#     - source: sub:offsets
#       target: save:offsets
#   rate: 1

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

Notes:

Attention: Make sure to set your graph rate to an high-enough value, otherwise the device internal buffer may saturate, and data may be lost. A 30Hz graph rate is recommended for a 1000Hz device rate.

Create instance and initialize the logger.

update (*self*)

terminate (*self*)

PYTHON MODULE INDEX

t

timeflux_bitalino, 3
timeflux_bitalino.nodes, 3
timeflux_bitalino.nodes.driver, 3

INDEX

B

`Bitalino` (*class in timeflux_bitalino.nodes.driver*), 3

M

module

`timeflux_bitalino`, 3

`timeflux_bitalino.nodes`, 3

`timeflux_bitalino.nodes.driver`, 3

T

`terminate()` (*timeflux_bitalino.nodes.driver.Bitalino*
 method), 5

`timeflux_bitalino`

 module, 3

`timeflux_bitalino.nodes`

 module, 3

`timeflux_bitalino.nodes.driver`

 module, 3

U

`update()` (*timeflux_bitalino.nodes.driver.Bitalino*
 method), 5