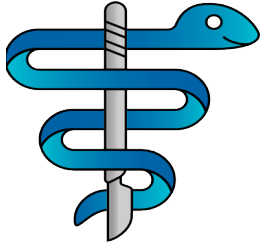

scikit-surgerybk Documentation

Thomas Dowrick

Feb 24, 2022

Contents

1	Developing	3
2	Installing	5
3	Licensing and copyright	7
4	Acknowledgements	9
	Python Module Index	15
	Index	17



Author: Thomas Dowrick

scikit-surgerybk is a Python library for interfacing with BK ultrasound machines, and is part of the [SciKit-Surgery](#) software project, developed at the [Wellcome EPSRC Centre for Interventional and Surgical Sciences](#), part of [University College London \(UCL\)](#).

scikit-surgerybk supports Python 2.7 and Python 3.6/3.7/3.8.

1.1 Cloning

You can clone the repository using the following command:

```
git clone https://github.com/SciKit-Surgery/scikit-surgerybk
```

1.2 Running tests

Pytest is used for running unit tests:

```
pip install pytest  
python -m pytest
```

1.3 Linting

This code conforms to the PEP8 standard. Pylint can be used to analyse the code:

```
pip install pylint  
pylint --rcfile=tests/pylintrc sksurgerybk
```


You can pip install directly from the repository as follows:

```
pip install git+https://github.com/SciKit-Surgery/scikit-surgerybk
```

2.1 Contributing

Please see the [contributing guidelines](#).

2.2 Useful links

- [Source code repository](#)
- [Documentation](#)

CHAPTER 3

Licensing and copyright

Copyright 2019 University College London. scikit-surgerybk is released under the BSD-3 license. Please see the [license file](#) for details.

Acknowledgements

Supported by [Wellcome](#) and [EPSRC](#).

4.1 Requirements for scikit-surgerybk

This is the software requirements file for scikit-surgerybk, part of the SNAPPY project. The requirements listed below should define what scikit-surgerybk does. Each requirement can be matched to a unit test that checks whether the requirement is met.

4.1.1 Requirements

ID	Description	Test
0000	Module has a help page	pylint, see tests/pylint.rc and tox.ini
0001	Functions are documented	pylint, see tests/pylint.rc and tox.ini
0002	Package has a version number	No test yet, handled by git.

4.2 latest

4.2.1 sksurgerybk package

Subpackages

sksurgerybk.interface package

Submodules

sksurgerybk.interface.bk5000 module

This module sets the connection to the BK scanner

class sksurgerybk.interface.bk5000.**BK5000** (*timeout, frames_per_second*)
Bases: object

This class sets the TCP connection with the BK scanner

clear_bytes_in_buffer (*start, end*)
Clear a set of bytes in bytearray buffer

Parameters

- **start** (*integer*) – Start index
- **end** (*integer*) – End integer

connect_to_host (*address, port*)
Connects the client to the host/serverself.

Implements a try/except block to catch potential errors.

Parameters: address(string): the IP address port(integer): the port

decode_image ()
Process the stream of data received from the BK5000 and convert it into a numpy array which represents the ultrasound image.

Control bytes are 1, 4 and 27. Flipped control bytes (1s complement of control bytes) are 254, 251, 228. Any time a flipped control bytes occurs after a 27, the value should be flipped and the preceding 27 deleted. See page 9 of 142 in BK doc PS12640-44 for further details.

disconnect_from_host ()
Disconnects the client from the host.

If the socket is already closed, a recv() call will throw an error. If it doesn't, we can close the socket.

enable_rgb_output ()
The 'QUERY:GRAB_FRAME "ON"' gets the BK to stream greyscale data (e.g. 640 x 480 x 1), some applications might want this in RGB format (640 x 480 x 3)

find_first_a_not_preceded_by_b (*start_pos, a, b*)
Find the first instance of 'a' in an array that isn't preceded by 'b'

Parameters

- **start_pos** (*integer*) – Index in array to begin search at
- **a** (*integer*) – Value to find
- **b** (*integer*) – Value not to precede a

Returns Index of first a not preceded by b, -1 if none found

Return type integer

generate_command_message (*message*)
Append 0x01 and 0x04 to start/end of message before sending

Parameters: message(string): the message to be sent

get_frame ()
Get the next frame from the BK5000.

parse_scanarea_message ()

Separate the scanarea response message into the separate components.

Message has format: <StartLineX>,<StartLineY>,<StartLineAngle>,<StartDepth>, <StopLineX>,<StopLineY>,<StopLineAngle>,<StopDepth>

Example message: `DATA:B_GEOMETRY_SCANAREA:A` 0.0017218,-0.000171398, 1.37236,0,-0.00174855,-0.000176821,1.77236,0.0203479;

parse_win_size_message (message)

Extract the size of the US window from the response message

Message has format “`DATA:US_WIN_SIZE` 640,480;”

Parameters: message(string): the received message

query_scanarea ()

Query the BK5000 for the scan area. (width/height/scanning depth etc.)

query_win_size ()

Query the BK5000 for the window/image size

receive_image ()

Scan the incoming data stream to find the start and end of the image data.

See BK doc PS12640-44 for further details.

receive_response_message (expected_size=1024)

Receive a message

Stores it under the data class member

Parameters: expected_size(int): the receive message size in bytes

request_stop ()

Set the appropriate class member

send_command_message (message)

Send a message through the socket.

Implements a couple of checks to verify the message has been sent correctly.

Parameters: message(string): the message to be sent

start_streaming ()

Send a message to start the streaming

stop_streaming ()

Send a message to stop the streaming. send_command_message and receive_response_message will throw errors if there is a problem with the socket connection.

```
class sksurgerybk.interface.bk5000.BKOpenCV (TCP_IP='128.16.0.3', TCP_PORT=7915,
                                             TIMEOUT=5, FPS=25)
```

Bases: object

Display BK data using OpenCV.

start ()

Start acquisition/streaming.

```
class sksurgerybk.interface.bk5000.BKpyIGTLink (TCP_IP='128.16.0.3',
                                                TCP_PORT=7915,      TIMEOUT=5,
                                                FPS=8)
```

Bases: object

Send BK data over OpenIGTLink.

start ()
Start acquisition/streaming.

stop ()
Stop acquisition/streaming.

`sk surgerybk.interface.bk5000.main ()`
Entry point for OpenCV/pyIGTLink.

Module contents

sk surgerybk.pyigtlink package

Submodules

sk surgerybk.pyigtlink.pyIGTLink module

Created on Tue Nov 3 19:17:05 2015

@author: Daniel Hoyer Iversen

class `sk surgerybk.pyigtlink.pyIGTLink.ImageMessage` (*image, spacing=[1, 1, 1], timestamp=None, device_name=""*)

Bases: `sk surgerybk.pyigtlink.pyIGTLink.MessageBase`

pack_body ()

class `sk surgerybk.pyigtlink.pyIGTLink.ImageMessageMatlab` (*image, dim, spacing=[1, 1, 1], timestamp=None*)

Bases: `sk surgerybk.pyigtlink.pyIGTLink.ImageMessage`

class `sk surgerybk.pyigtlink.pyIGTLink.MessageBase`

Bases: `object`

`message`

get_binary_body ()

get_binary_message ()

get_body_pack_size ()

is_valid ()

pack ()

pack_body ()

class `sk surgerybk.pyigtlink.pyIGTLink.PyIGTLink` (*port=18905, localServer=True, iface='eth0'*)

Bases: `socketserver.TCPServer`

For streaming data over TCP with IGTLink

add_message_to_send_queue (*message, wait=False*)

Returns True if successful

close_server ()

Will close connection and shutdown server

get_ip_adress ()

get_port_no ()


```
    is_connected()
    update_connected_status(val)
class sksurgerybk.pyigtlink.pyIGTLink.TCPRequestHandler(request,    client_address,
                                                         server)
    Bases: socketserver.BaseRequestHandler
    Help class for PyIGTLink
    handle()
class sksurgerybk.pyigtlink.pyIGTLink.TransformMessage(tform, timestamp=None, de-
                                                         vice_name="")
    Bases: sksurgerybk.pyigtlink.pyIGTLink.MessageBase
    pack_body()
```

Module contents

Module contents

scikit-surgerybk

4.2.2 work_in_progress module

- [modindex](#)
- [genindex](#)
- [search](#)

i

`skssurgerybk.interface`, [12](#)

`skssurgerybk.interface.bk5000`, [10](#)

p

`skssurgerybk.pyigtlink`, [13](#)

`skssurgerybk.pyigtlink.pyIGTLink`, [12](#)

s

`skssurgerybk`, [13](#)

A

`add_message_to_send_queue()`
*(sksurgerybk.pyigtlink.pyIGTLink.PyIGTLink
method), 12*

B

`BK5000` (class in *sksurgerybk.interface.bk5000*), 10
`BKOpenCV` (class in *sksurgerybk.interface.bk5000*), 11
`BKpyIGTLink` (class in *sksurgerybk.interface.bk5000*),
11

C

`clear_bytes_in_buffer()`
*(sksurgerybk.interface.bk5000.BK5000
method), 10*
`close_server()` (*sksurgerybk.pyigtlink.pyIGTLink.PyIGTLink*
method), 12
`connect_to_host()`
*(sksurgerybk.interface.bk5000.BK5000
method), 10*

D

`decode_image()` (*sksurgerybk.interface.bk5000.BK5000*
method), 10
`disconnect_from_host()`
*(sksurgerybk.interface.bk5000.BK5000
method), 10*

E

`enable_rgb_output()`
*(sksurgerybk.interface.bk5000.BK5000
method), 10*

F

`find_first_a_not_preceded_by_b()`
*(sksurgerybk.interface.bk5000.BK5000
method), 10*

G

`generate_command_message()`
*(sksurgerybk.interface.bk5000.BK5000
method), 10*
`get_binary_body()`
*(sksurgerybk.pyigtlink.pyIGTLink.MessageBase
method), 12*
`get_binary_message()`
*(sksurgerybk.pyigtlink.pyIGTLink.MessageBase
method), 12*
`get_body_pack_size()`
*(sksurgerybk.pyigtlink.pyIGTLink.MessageBase
method), 12*
`get_frame()` (*sksurgerybk.interface.bk5000.BK5000*
method), 10
`get_ip_address()` (*sksurgerybk.pyigtlink.pyIGTLink.PyIGTLink*
method), 12
`get_port_no()` (*sksurgerybk.pyigtlink.pyIGTLink.PyIGTLink*
method), 12

H

`handle()` (*sksurgerybk.pyigtlink.pyIGTLink.TCPRequestHandler*
method), 13

I

`ImageMessage` (class in
sksurgerybk.pyigtlink.pyIGTLink), 12
`ImageMessageMatlab` (class in
sksurgerybk.pyigtlink.pyIGTLink), 12
`is_connected()` (*sksurgerybk.pyigtlink.pyIGTLink.PyIGTLink*
method), 13
`is_valid()` (*sksurgerybk.pyigtlink.pyIGTLink.MessageBase*
method), 12

M

`main()` (in module *sksurgerybk.interface.bk5000*), 12
`MessageBase` (class in
sksurgerybk.pyigtlink.pyIGTLink), 12

P

`pack()` (*sk surgerybk.pyigtlink.pyIGTLink.MessageBase* *method*), 12

`pack_body()` (*sk surgerybk.pyigtlink.pyIGTLink.ImageMessage* *method*), 12

`pack_body()` (*sk surgerybk.pyigtlink.pyIGTLink.MessageBase* *method*), 12

`pack_body()` (*sk surgerybk.pyigtlink.pyIGTLink.TransformMessage* *method*), 13

`parse_scanarea_message()` (*sk surgerybk.interface.bk5000.BK5000* *method*), 10

`parse_win_size_message()` (*sk surgerybk.interface.bk5000.BK5000* *method*), 11

`PyIGTLink` (*class in sk surgerybk.pyigtlink.pyIGTLink*), 12

Q

`query_scanarea()` (*sk surgerybk.interface.bk5000.BK5000* *method*), 11

`query_win_size()` (*sk surgerybk.interface.bk5000.BK5000* *method*), 11

R

`receive_image()` (*sk surgerybk.interface.bk5000.BK5000* *method*), 11

`receive_response_message()` (*sk surgerybk.interface.bk5000.BK5000* *method*), 11

`request_stop()` (*sk surgerybk.interface.bk5000.BK5000* *method*), 11

S

`send_command_message()` (*sk surgerybk.interface.bk5000.BK5000* *method*), 11

`sk surgerybk` (*module*), 13

`sk surgerybk.interface` (*module*), 12

`sk surgerybk.interface.bk5000` (*module*), 10

`sk surgerybk.pyigtlink` (*module*), 13

`sk surgerybk.pyigtlink.pyIGTLink` (*module*), 12

`start()` (*sk surgerybk.interface.bk5000.BKOpenCV* *method*), 11

`start()` (*sk surgerybk.interface.bk5000.BKpyIGTLink* *method*), 11

`start_streaming()` (*sk surgerybk.interface.bk5000.BK5000* *method*), 11

`stop()` (*sk surgerybk.interface.bk5000.BKpyIGTLink* *method*), 12

`stop_streaming()` (*sk surgerybk.interface.bk5000.BK5000* *method*), 11

T

`TCPRequestHandler` (*class in sk surgerybk.pyigtlink.pyIGTLink*), 13

`TransformMessage` (*class in sk surgerybk.pyigtlink.pyIGTLink*), 13

`update_connected_status()` (*sk surgerybk.pyigtlink.pyIGTLink.PyIGTLink* *method*), 13

U