BIOSPACE User Manual

Hampus Månefjord

Department of Physics, Lund University

March 20, 2022

This document contains instructions for the biophotonics platform BIOSPACE. Contact us at:

hampus.manefjord@forbrf.lth.se

mikkel.brydegaard@forbrf.lth.se

Contents

1. Instrumentation overview

The instrument consists of a multispectral light source; optical elements for imaging a biological target; a mechanical structure with four degrees of freedom for rotation; a hardware control interface; and an application controlling the data acquiring process. An overview of the instrument is presented in Fig.1.

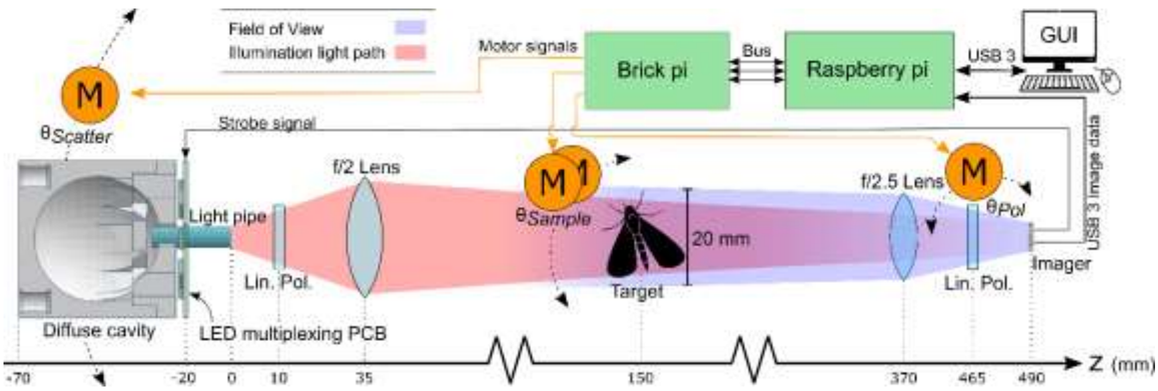


Fig. 1. Block diagram of hardware control interfaces and schematic drawing of optical components.

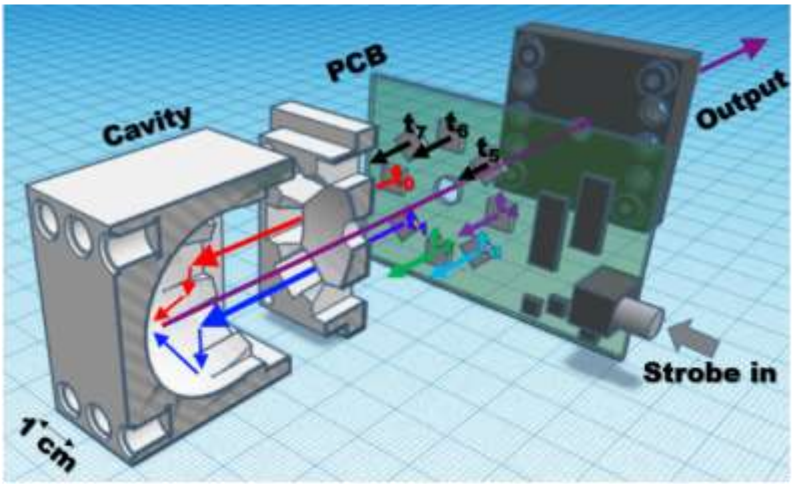


Fig. 2. A schematic drawing based on a CAD of the LED multiplexer with a white diffusive cavity displaying its functionality

The light source is a LED multiplexor with a white, diffusive cavity seen in Fig. 2. The LED multiplexor is made of some basic electronic components including LEDs with 8 different wavelengths (365 – 940 nm) that are strobed one at a time in different time slots. A camera synchronized with the LEDs captures each image which together form a multispectral image.

The instrument’s mechanical parts are based on Lego and 3D-printed parts. The sample can be rotated in two degrees of freedom: the “yaw” and “aspect” angles. Further a polarization filter can be rotated and the angle of light incidence on the sample can be controlled with the “scatter angle”. These angles are seen in Fig. 3.

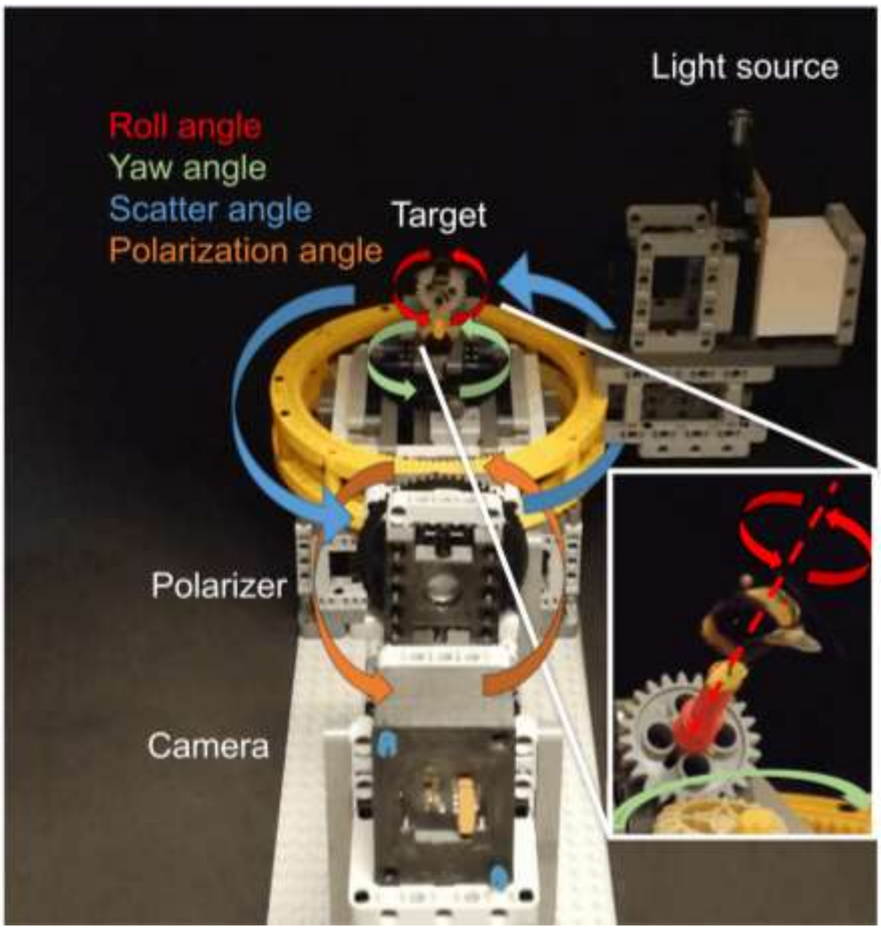


Fig. 3. Motorized degrees of freedom of the movements of the platform includes scatter angle, polarization angle, aspect angle and yaw angle.

2. Hardware

2.1. Electronics components

List of hardware includes:

- Raspberry pi 4b
- Brickpi board 3
- Camera: Basler Ace acA1920-155um
- 3 units of EV3 lego servo motors
- USB3 Hard drive
- Display/Monitor
- Keyboard
- Mouse
- Custom Multiplexing PCB

Cables:

- 3 units of 50cm EV3 cables
- Basler GP-I/O Cable, HRS 6p/open
- MicroHDMI cable
- Coaxial BNC cable
- 9V 4.44A power supply

- 18V 1A power supply

Connect as follows:

- Assemble the BrickPi3 case according to https://www.dexterindustries.com/BrickPi/brickpi3-getting-started-step-1-assembly/
- The keyboard and mouse are connected to the USB2 ports of the Raspberry pi
- The hard drive and the camera are connected to the USB3 ports of the Raspberry pi
- The display/monitor are connected to the Raspberry pi through the microHDMI port
- The Brickpi 3 is placed on the 40-pin GPIO header of the Raspberry pi
- The 3 lego motors are connected to the Brickpi through lego EV3 cables
- The Brickpi is supplied with 9V
- The PCB is connected to the Strobe-out of the camera with a Basler GP-I/O Cable, HRS 6p/open to coaxial BNC
- The PCB is supplied with 18V

See the bill_of_materials.pdf for full list of components.

2.2. 3D-printed components

The 3D printed components are compatible with Lego Technic and includes:

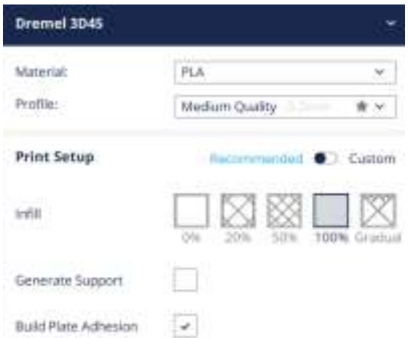
- Cavity for diffusing the LED light
  - Print with 100% infill with white ABS/PLA filament
  - for optimal UV performance apply some $BaSO_4$-coating to the inside of the cavity
- Lens adaptor
- Camera adaptor
- PCB adaptor

Print all adaptors with black filament (100% infill is not necessary)

If you are using the Dremel 3D45 for 3D-printing, use the software Dremel Digilab 3D slicer

Select medium (0.2mm) or higher profile quality, chose infill (eg 100% or 20% depending on the piece) have Generate support unticked, but build plate adhesion ticked.

Generate a file and load in onto the printer with a USB memory stick.



2.3. PCB

The PCB is driving the LEDs one at the time, synchronized with the camera. There are attached electrical schema, PCB-layout file, and a ready-to-run (setting may vary depending on router) gcode file with the routing pattern for milling the PCB.

If you are making the PCB yourself with a CNC-mill see the document **PCB_making_workflow.pdf**

2.4. Lego Assembly

Lego prices needed are found in the

- Bucket-wheel-excavator kit
- 4 EV3 motors
- 4 EV3 cables

Instructions for assembly are found in the file **Lego_assembly_Instructions_for_BIOSPACE.pdf**

3. Installing software

Operating system

Flash SDcard of the raspberry pi by following the guide at

https://www.dexterindustries.com/howto/install-raspbian-for-robots-image-on-an-sd-card/

with biospace_install\2020-02-13-raspbian-buster-full.img

Move the SDcard to the raspberry pi and boot

You can set the time with terminal input

```
sudo date -s "Jul 5 08:10"

sudo dpkg-reconfigure tzdata
```

Brickpi

```
curl -kL dexterindustries.com/update_brickpi3 | bashkpy

sudo reboot
```

Testfiles are located at /home/pi/Dexter/BrickPi3/Software/Python

More info at https://www.dexterindustries.com/BrickPi/brickpi3-getting-started-step-4-program-brickpi-robot/brickpi3-getting-started-program-python/

Camera software

Install by

```
sudo dpkg -i pylon_6.X.XX.XXXXX-deb0_amd64.deb
```

Run the software

```
/opt/pylon/bin/pylonviewer
```

More info at https://github.com/strawlab/flydra/wiki/Basler-Pylon-for-Linux-backend

Camera – python interface

find correct version of whl file:

```
uname -a
    ➢ armv7l
python3 -V
    ➢ cp37
```

Your responses might differ. Find the corresponding .whl file at https://github.com/Basler/pypylon/releases

and download

install pypylon:

```
pip3 install pypylon-1.5.1-cp37-cp37m-linux_armv7l.whl
```

or equivalent.

4. Operation

Place your sample on a pin, with some adhesive/blu tack or on a 3D-printed sample holder. Some modification to the setup might be necessary depending on the size of the sample

Test motors

After connecting motors according to section 2, run the python programs

move_roll_angle.py

move_yaw_angle.py

move_scatter_angle.py

and make sure the specified motor rotates. The number of degrees to be moved by each script can be modified by editing the code in each file.

Test camera

For testing, start the pylonviewer software to get a stream from the camera. You do that by opening a terminal and writing:

```
/opt/pylon/bin/pylonviewerApp
```

Here you can adjust camera setting such as exposure time until you get an image of your sample.

To test the multiplexing PCB connect its BNC coax to the GPIO of the camera. The use the setting

Output

Line 3

Exposure active

And connect the multiplexing PCB to a 18V power supply.

Adjust the camera distance from the lens until it is in focus.

If you will use multiple aspect/yaw angles of your sample, you can rotate it and make sure is still in the field of view of the camera for all relevant rotations.

Now you need to define your measurement protocol which is a series of angles and exposure times. One row of the protocol spreadsheet is executed at a time

Roll	**Yaw**	**Scatter**
0	0	0
0	90	0
0	180	0
0	270	0
0	0	30
0	90	30
0	180	30
0	270	30
90	0	0
90	90	0
90	180	0
90	270	0
90	0	30
90	90	30
90	180	30
90	270	30

In this example protocol file there will be 16 measurements, first the yaw angle of the sample will be rotated a complete revolution, then the roll will be rotated, and a new revolution of the aspect angle will be captured. Next the same procedure will be done again but with a new scatter angle and exposure time.

You then find the acquisition code in biospace_code/ biospace_main.py

The program can be run as any python script, e.g., with Thonny.

To make a measurement:

1. Have a defined protocol of relevant positions
2. Move the instrument into a starting position.
3. Turn off (or down) the room lighting.
4. Start the measurement GUI.
5. power on the instrument light source.
6. Press start camera in the GUI. The light source should now cycle through the different colored LEDs, and it performs a calibration to find the timeslot with all LEDs turned off.
7. After a few cycles - press “start measurement”. A text “MEASURING” will be in the GUI while the measurement is ongoing.

After you have done your measurement make sure to load and look at the files to make sure you have captured what you need, and not, e.g., underexposed or overexposed all images.

There will be 8 images for each row in the protocol file so possibly a lot of images to look at. So there is also a spreadsheet output to get an overview of how many overexposed pixels there are in each image.

5. Reading and visualizing data

The program captures images, saves them in a folder named after the instrument angles, and names the tiff files according to the illumination wavelength. E.g. the path for an image could be "..\scatter_60_yaw_10_roll_0\365_nm.tiff".

Images can be analyzed using any program that can read tiff files, e.g. Matlab.

Load the image in Matlab the with

```
img_630nm=imread('\scatter_60_yaw_10_roll_0\630_nm.tiff');
```

A matrix of intensity values between 0 and 1 is loaded, in double precision. The acquisition program has before saving the images subtracted the background signal and converted the images from 12 bit unsigned integer to double.

The values need to be corrected for the emission intensity of the LEDs. That is done by dividing the intensity values with the values from an image of a white calibration target illuminated with the same emission wavelength.

```
img_630nm = img_630nm/mean(mean(calibration_img_630nm));
```

A color image is displayed by concatenating three images representing red, green and blue such as

```
color_img= cat(3,img_red, img_green, img_blue);
imagesc(color img);
```

There are some Matlab scripts available for reading the images and doing some simple visualization.

Start with the file biospace_main.m

Good luck!