# Introduction

This document is a small user manual for SICSURFIS AgMFPI-H019 Vaasa University Spectral Imager camera.

# Version history

2020.12.16 Daniel Tisza  
2020.12.17 Daniel Tisza  
2020.12.18 Daniel Tisza  
2021.01.05 Daniel Tisza  
2021.02.14 Daniel Tisza  
2021.05.23 Daniel Tisza  
2021.08.31 Daniel Tisza  
2021.09.05 Daniel Tisza

Contents

[Introduction 1](#_Toc81729590)

[Version history 1](#_Toc81729591)

[Terminology 2](#_Toc81729592)

[Camera 2](#_Toc81729593)

[Overview 2](#_Toc81729594)

[Camera properties 3](#_Toc81729595)

[Camera wavelength bands 4](#_Toc81729596)

[LED settings for wavelength bands 4](#_Toc81729597)

[Single wavelength bands 5](#_Toc81729598)

[LED settings for single wavelength bands 5](#_Toc81729599)

[FPI USB IDs 6](#_Toc81729600)

[LED USB Ids 6](#_Toc81729601)

[Tarvittavat ohjelmistot 7](#_Toc81729602)

[Windows 7](#_Toc81729603)

[Kameran ajurit 7](#_Toc81729604)

[Python 3.7.6 8](#_Toc81729605)

[Camazing 8](#_Toc81729606)

[Matplotlib 8](#_Toc81729607)

[Git 9](#_Toc81729608)

[FPIPY 9](#_Toc81729609)

[Leddriver 9](#_Toc81729610)

[Spectracular 9](#_Toc81729611)

[Numpy sanity check error 9](#_Toc81729612)

[LED-moduuli COM-porttinumero 9](#_Toc81729613)

[LED-moduuli fixed file kopiointi 10](#_Toc81729614)

[FPIPY fixed file kopiointi 10](#_Toc81729615)

[Linux 11](#_Toc81729616)

[Tietokoneen tiedot 11](#_Toc81729617)

[Kameran ajurien asennus 11](#_Toc81729618)

[Ubuntu 18.04 Python3.7 virtuaaliympäristön asennus 13](#_Toc81729619)

[Python3.7 ohjelmistopakettien asennus 14](#_Toc81729620)

[Käynnistä tietokone uudelleen 14](#_Toc81729621)

[Site-packages tiedostojen korjaus 15](#_Toc81729622)

[Kopioi korjattu FPIPY fixed file ”fpi\_driver.py” 15](#_Toc81729623)

[Korjaa LED USB ID:t “led\_driver.py” 15](#_Toc81729624)

[Ajetaan moduulien testiskriptejä 16](#_Toc81729625)

[LED-testaus 16](#_Toc81729626)

[MFPI-testaus 16](#_Toc81729627)

[Kameramoduulin testaus 16](#_Toc81729628)

[Spektrikuvien ottaminen 17](#_Toc81729629)

[Kameraliitännän testaus 17](#_Toc81729630)

[test\_camera\_module.py 18](#_Toc81729631)

[test\_fpi\_module.py 18](#_Toc81729632)

[test\_led\_module.py 18](#_Toc81729633)

[Kameran käyttö 20](#_Toc81729634)

[Kalibraatiotiedosto 20](#_Toc81729635)

[Ledien valinta 20](#_Toc81729636)

[Python-ohjelma 21](#_Toc81729637)

# Terminology

FWHM Full width at half maximum (at half maximum amplitude)

# Camera

## Overview

SICSURFIS AgMFPI-H019 camera is intended for taking images at different wavelengths. The camera itself consists of three different components: Basler camera module, AgMFPI MEMS filter and 27 LEDs control unit. Each of these components has a USB connection for controlling them for taking images.

Following overview diagram shows camera components and connections of the SICSURFIS AgMFPI-H019 camera.

Basler Camera

Module

AgMFPI

MEMS

Filter

27 LEDs

Control

USB3

USB2

USB2

Picture 1. SICSURFIS AgMFPI-H019 camera overview diagram showing camera components and connections.

## Camera properties

|  |  |  |
| --- | --- | --- |
| Parameter | Value |  |
| MFPI spectral range | 500 – 950 nm |  |
| Spectral resolution | 15…32 nm @ FWHM |  |
| Spectral step | < 1 nm |  |
| Frame rate for data cube | < 5.0 s |  |

(Heikki Saari, 2020).

## Camera wavelength bands

Total 306 wavelength band images can be captured.

Three different kinds of images can be produced:

* RAW image
* Radiance image
* Reflectance image

Camera wavelength band is selected by adjusting the AgMFPI MEMS filter drive voltage and turning on the correct LEDs using the 27 LEDs control unit.

Depending on AgMFPI MEMS filter setting there are 1-3 wavelength bands that are transmitted to the image sensor.

Following table shows the LED set and peak WL, first and last wavelengths and wavelength count and LED control value (reserved bits).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| LED set and peak WL | First wavelength [nm] | Last wavelength [nm] | Wavelength count | LED control (reversed) |
| C WL1 | 504.5181438 | 612.2236991 | 75 | 000000001000000001000000001 |
| B WL1 | 533.8275323 | 568.2446615 | 31 | 000000011000000011000000011 |
| A WL1 | 542.8327583 | 552.8525817 | 12 | 000000111000000111000000111 |
| C WL2 | 587.2723752 | 658.6319132 | 26 | 000000001000000001000000001 |
| D | 695.9738452 | 738.7581953 | 34 | 000011110000011110000011110 |
| B WL2 | 660.5024267 | 700.3681421 | 31 | 000000011000000011000000011 |
| A WL2 | 701.3626464 | 710.1310492 | 12 | 000000111000000111000000111 |
| E | 740.2281587 | 778.9766281 | 19 | 000111100000111100000111100 |
| F | 775.054537 | 818.8148698 | 22 | 001111000001111000001111000 |
| G | 810.0078184 | 848.0322309 | 17 | 011110000011110000011110000 |
| H | 851.0365439 | 939.6902201 | 27 | 111100000111100000111100000 |

### LED settings for wavelength bands

## Single wavelength bands

Depending on AgMFPI MEMS filter setting there are 1-3 wavelength bands that are transmitted to the image sensor. I have selected 221 settings where only one wavelength band is transmitted to the image sensor.

Picture 2. Wavelength bands where only one wavelength band is transmitted to the image sensor.

Following table shows the selected index, wavelength band center wavelength and wavelength band.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Index | Center [nm] | FWHM [nm] |  |  |
| 92 | 554,047301218381 | 20,3669241303736 |  |  |
| … |  |  |  |  |
| 1 | 584,577534977943 | 15,8482711179199 |  |  |
| … |  |  |  |  |
| 119 | 904,48089817979 | 23,8573250412726 |  |  |
| … |  |  |  |  |
| 222 | 939,6902201137 | 23,169132461026 |  |  |

### LED settings for single wavelength bands

## FPI USB IDs

FPI\_IDS = [  
 # ( VID, PID) (and the same in decimal)  
 ('1FC9', '0083'), (8137, 131),  
 ]

"""Known VID:PID pairs of FPI devices."""  
FPI\_HWIDS = [  
 # Strings returned by read\_hardware\_id  
 'd02b012 af380065 5b5bbeab f50019c1'  
]

## LED USB Ids

LED-ohjaus toimii kamerassa USB:n virtuaalisen sarjaportin kautta ja ajuri tarvitsee LED-ohjauksen tunnistamiseen oikeat VID, PID ja HWID-tiedot.

LED\_IDS = [  
 # ( VID, PID) (and the same in decimal)  
 ('1FC9', '0083'), (8137, 131),  
 ]

"""Known VID:PID pairs of LED devices."""  
LED\_HWIDS = [  
 # Strings returned by read\_hardware\_id  
 '1000e016 aefba123 580267dd f5001982',  
 '10025018 af28a028 5a66a511 f5001983'  
]

# Tarvittavat ohjelmistot

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Kameran ajurit |  |  |
| GenICam GenTL Produce tiedosto |  |  |
| Python 3.7.6 |  |  |
| Camazing |  |  |
| Fpipy |  |  |
| Leddriver |  |  |
| Spectracular |  |  |

# Windows

## Kameran ajurit

<http://static.matrix-vision.com/mvIMPACT_Acquire/2.39.0/>

Asennetaan Administratorina:

mvGenTL\_Acquire-x86\_64-2.39.0.exe

C:\Program Files\MATRIX VISION\mvIMPACT Acquire

Asennetaan Administratorina:

mvBlueFOX-x86\_64-2.39.0.msi

mvDeviceConfigure-ohjelmassa tuli näkyviin tällaisia tietoja:

Family: USB3VisionDevice  
 Product: daA2500-14uc  
 Serial: 22458797  
 State: Present

mvDeviceConfigure

Tuo ohjelma näyttää, että kamera löytyy USB:stä.

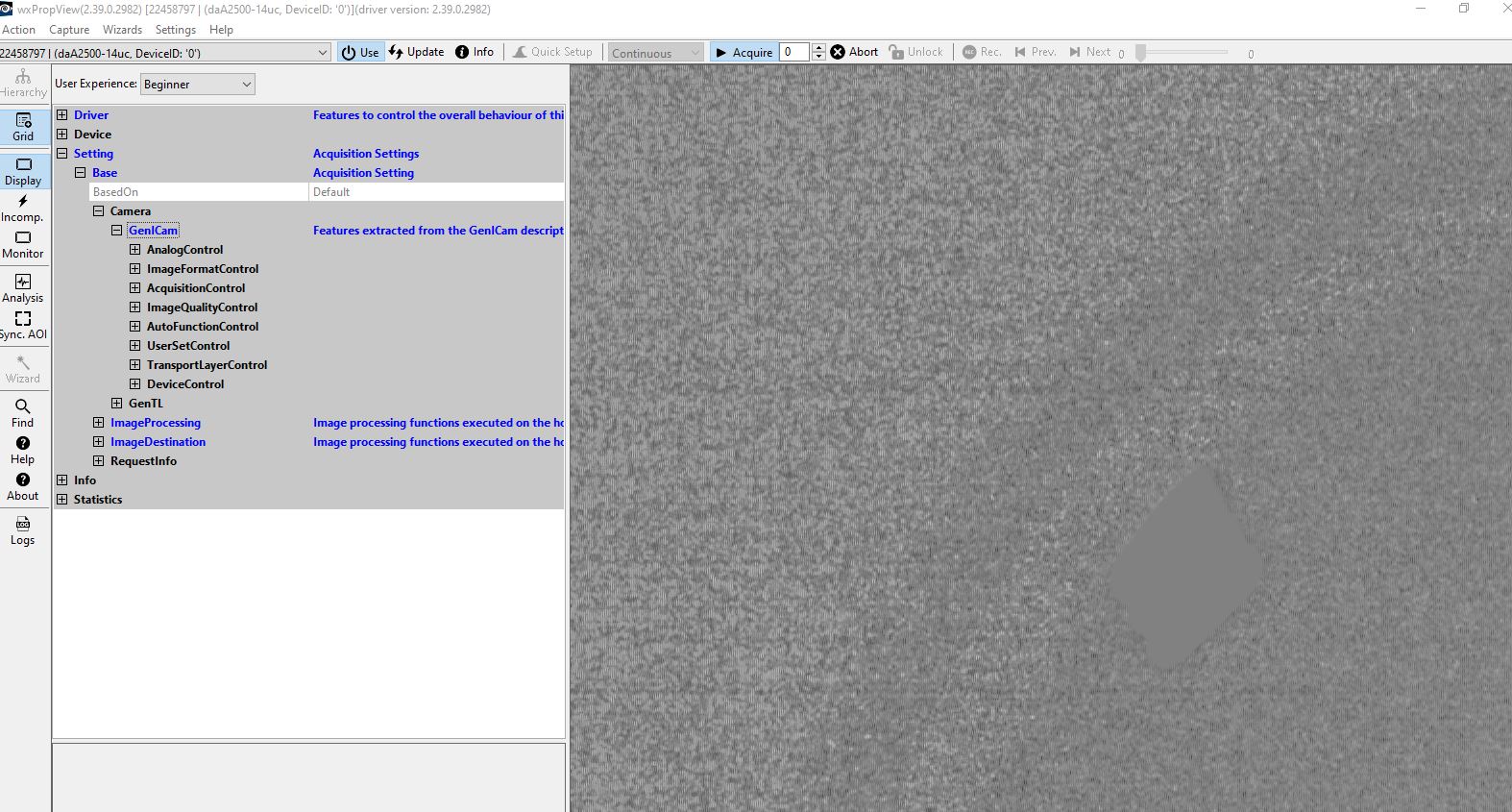
wxPropView

Tuolla ohjelmalla sain otettua kameralta kuvaa tietokoneelle

Painetaan ”Use”

Vaihdetaan ”Single Frame” (USB 2.0 portit eivät tue muuta kuin single frame kaappausta)

Painetaan ”Acquire”



## Python 3.7.6

Asensin python-3.7.6-amd64.exe

Pitää asentaa Administrator oikeuksilla

* Rastita ruutuun ”Add Python 3.7 to PATH”
* Customize installation
* Install for all users (myöhempi välilehti)
* Disable path length limit (myöhempi välilehti)

Python versio info:

**Python 3.7.6 (tags/v3.7.6:43364a7ae0, Dec 19 2019, 00:42:30) [MSC v.1916 64 bit (AMD64)] on win32**

## Camazing

Käynnistä komentokehote Administratorina (cmd.exe)

Asensin camazing-paketin Pythonille:

C:\WINDOWS\system32>pip install camazing

## Matplotlib

Käynnistä komentokehote Administratorina (cmd.exe)

Asensin matplotlib-paketti Pythonille:

C:\WINDOWS\system32>pip install matplotlib

## Git

Asensin Git-versionhallinnan FPIPY-paketin hakemista varten:

Git-2.28.0-64-bit.exe

## FPIPY

Käynnistä ”Git CMD”-komentorivi Administratorina.

Anna siellä asennuskomento FPIPY-paketille (tämä vaatii ensin Git-asennuksen):

C:\tldati\zyboz7\python>pip install git+https://github.com/silmae/fpipy.git

## Leddriver

Käynnistä komentokehote Administratorina (cmd.exe)

Mene kansioon, jossa löytyy leddriver-master.zip

Asensin leddriver-master-paketin:

C:\tldati\zyboz7\kamera\packets>pip install leddriver-master.zip

(Pitikö kopioida USB ID-muokkausten jälkeen korjattu versio system-hakemistoon? tarkista)

## Spectracular

Käynnistä komentokehote Administratorina (cmd.exe)

Mene kansioon, jossa löytyy spectracular-led\_version.zip

Asensin spectracular-paketin:

C:\tldati\zyboz7\kamera\packets>pip install spectracular-led\_version.zip

## Numpy sanity check error

Severin tietokoneella tuli Numpy ja Windowsin runtime:n kanssa yhteensopivuusongelma.

Tähän auttaa, jos pakotetaan Numpy vanhempaan versioon.

Käynnistä Administratorina komentokehote (cmd.exe)

pip install numpy==1.19.3

## LED-moduuli COM-porttinumero

LED-moduulin portti pitää tarkistaa.

Avaa Laitehallinta (Device Manager)

Katso laitehallinnasta ”Portit” alla, mikä on ”USB-laite” porttinumero, esimerkiksi ”COM6”

## LED-moduuli fixed file kopiointi

Kopioi support-paketista tämä tiedosto:

fixed\_package\_files\led\_driver.py

Windowsissa tänne:

C:\Program Files\Python37\Lib\site-packages\LEDDriver\led\_driver.py

Linuxissa tänne:

~/python3venv2/lib/python3.7/site-packages/LEDDriver/led\_driver.py

Zybo:ssa tänne

~/python3venv2/lib/python3.5/site-packages/LEDDriver/led\_driver.py

Tällöin tulee oikeat H019 kameran LED-moduulin USB ID:t käyttöön.

Kopiointi vaatii Administrator-oikeudet.

## FPIPY fixed file kopiointi

Kopioi support-paketista tämä tiedosto:

fixed\_package\_files\fpi\_driver.py

Tänne:

C:\Program Files\Python37\Lib\site-packages\spectracular\fpi\_driver.py

Linuxissa:

~/python3venv2/lib/python3.7/site-packages/spectracular/fpi\_driver.py

Tällöin tulee oikea H019 kameran tarvitsema MFPI-ajuri käyttöön.

Kopiointi vaatii Administrator-oikeudet.

# Linux

## Tietokoneen tiedot

Ubuntu 18.04.1

uname -a  
Linux xilinx-ThinkCentre-M93p 5.4.0-42-generic #46~18.04.1-Ubuntu SMP  
Fri Jul 10 07:21:24 UTC 2020  
x86\_64 x86\_64 x86\_64 GNU/Linux

## Kameran ajurien asennus

Avaa tämä valmistajan ohjelmistosivu:

<http://static.matrix-vision.com/mvIMPACT_Acquire/2.39.0/>

Lataa ohjelmistosivulta nämä kaksi tiedostoa omalle tietokoneelle samaan ”Demokamera”-kansioon:

* install\_mvGenTL\_Acquire.sh
* mvGenTL\_Acquire-x86\_64\_ABI2-2.39.0.tgz

Kiinnitä kameran USB-johto

Aja seuraava komennot komentorivillä “Demokamera”-kansiossa:

sudo dmesg

Tämä komento näyttää, että löytyy “daA2500-14uc” laite

Aja seuraava komennot komentorivillä “Demokamera”-kansiossa:

chmod a+x install\_mvGenTL\_Acquire.sh  
./install\_mvGenTL\_Acquire.sh

Asennusohjelma kysyy, laitetaanko ”unattended mode” ja hyväksy se painamalla ”Enter”. Tällöin asennusohjelma asentaa kameran ajurit oletusasetuksilla. Asennukseen voi mennä 10 minuuttia.

Asennuksen jälkeen piti käynnistää tietokone uudelleen.

Aja komentorivillä ohjelma:

mvDeviceConfigure

Tämä ohjelma näyttää, että kamera löytyy USB:stä.

Aja komentorivillä ohjelma:

wxPropView

Tällä ohjelmalla voi ottaa kameralla kuvaa tietokoneelle

Paina ”Use”-painiketta

Vaihde ”Continuous”-asetus ”SingleFrame”-asetukseksi

Valitse ”Capture”-valikosta ”Acquire”

Paina muutamia kertoja ”Acquire”-nappia, niin kuvan pitäisi päivittyä näytöllä.

Kun LEDit ovat pois päältä, niin kuva on tumman harmaa.

## Ubuntu 18.04 Python3.7 virtuaaliympäristön asennus

notes.docx => Muistiinpanoja 2021.05.10

Spectracular vaatii Python 3.7!

sudo apt-get install python3.7

sudo apt-get install libpython3.7-dev

sudo apt-get install python3.7-venv

sudo apt install python3-pip (tätä ei ehkä tarvita?)

python3.7 –V

python3.7 -m venv python37venv

Käynnistä python3.7 virtuaaliympäristö:

**. python37venv/bin/activate**

python -V

* Pitäisi näyttää 3.7.5

pip -V

Ubuntu 20.04

* Python 3.8.10

## Python3.7 ohjelmistopakettien asennus

pip install numpy==1.18

pip install cython

pip install wheel

pip install camazing (pandas kääntäminen kestää kauan 15-20 min!)

pip install matplotlib

sudo apt-get install git

pip install git+https://github.com/silmae/fpipy.git

pip install leddriver-master.zip

pip install spectracular-led\_version.zip

## Käynnistä tietokone uudelleen

Käynnistä python3.7 ympäristö:

**. python37venv/bin/activate**

## Site-packages tiedostojen korjaus

### Kopioi korjattu FPIPY fixed file ”fpi\_driver.py”

Selvitetään site-packages

python -m site --user-site

Kopioi korjattu FPIPY driver-tiedosto site-packages kansioon:

~/python3venv2/lib/python3.7/site-packages/spectracular/fpi\_driver.py

Tämä korjaa käyttämään MFPI-ajuria, eikä virheellistä PFPI-ajuria.

### Korjaa LED USB ID:t “led\_driver.py”

Selvitetään site-packages

python -m site --user-site

Muokkaa tähän tiedostoon oikeat ID:t:

~/python3venv2/lib/python3.7/site-packages/LEDDriver/led\_driver.py

LED\_IDS = [

# ( VID, PID) (and the same in decimal)

('1FC9', '0083'), (8137, 131),

]

"""Known VID:PID pairs of LED devices."""

LED\_HWIDS = [

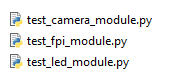
# Strings returned by read\_hardware\_id

'1000e016 aefba123 580267dd f5001982',

'10025018 af28a028 5a66a511 f5001983'

]

## Ajetaan moduulien testiskriptejä



**sudo chmod a+rwx /dev/ttyACM0**

**sudo chmod a+rwx /dev/ttyACM1**

### LED-testaus

Vedä irti LED-johto (valkoinen)

Kiinnitä LED-johto (valkoinen)

sudo dmesg

/dev/ttyACM0

cd Demokamera

. python37venv/bin/activate

python test\_led\_module.py

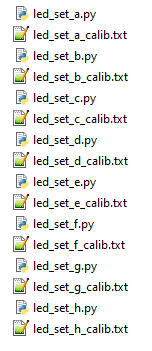
### MFPI-testaus

python test\_fpi\_module.py

### Kameramoduulin testaus

python test\_camera\_module.py

## Spektrikuvien ottaminen



**cd Demokamera**

**. python37venv/bin/activate**

**sudo chmod a+rwx /dev/ttyACM0**

**sudo chmod a+rwx /dev/ttyACM1**

**Pitää korjata LED-driverin oikea sarjaporttiasetus kuvausskripteihin!**

**python led\_set\_a.py**

**python led\_set\_h.py**

# Kameraliitännän testaus

Tein kolme erillistä Python-ohjelmaa, jolla voidaan testata kameran eri moduulien toimintaa yksitellen, esimerkiksi jos tietokoneessa on asennusvaiheessa vain yksi USB-portti.

python test\_camera\_module.py

* Tämä näyttää, että kamera löytyy camazing-paketin kautta

python test\_fpi\_module.py

* Tämä näyttää, että FPI-objektin luonti onnistuu

python test\_led\_module.py

* Tämä näyttää, että LED-ohjaus toimii, sytyttää LED:ejä.

## test\_camera\_module.py

from camazing import CameraList

cameras = CameraList()  
print(cameras)

camera = cameras[0]  
camera.initialize()

## test\_fpi\_module.py

import fpipy as fp  
import matplotlib

from spectracular.fpi\_driver import detectFPIDevices, createFPIDevice

FPI\_IDS = [  
 # ( VID, PID) (and the same in decimal)  
 ('1FC9', '0083'), (8137, 131),  
 ]

"""Known VID:PID pairs of FPI devices."""  
FPI\_HWIDS = [  
 # Strings returned by read\_hardware\_id  
 'd02b012 af380065 5b5bbeab f50019c1'  
]

fpi = createFPIDevice(detectFPIDevices(FPI\_IDS, FPI\_HWIDS)[0].device)  
print(fpi)

## test\_led\_module.py

import matplotlib  
from LEDDriver import detect\_LED\_devices, LEDDriver, LEDException

LED\_IDS = [  
 # ( VID, PID) (and the same in decimal)  
 ('1FC9', '0083'), (8137, 131),  
 ]  
"""Known VID:PID pairs of LED devices."""  
LED\_HWIDS = [  
 # Strings returned by read\_hardware\_id  
 '1000e016 aefba123 580267dd f5001982',  
 '10025018 af28a028 5a66a511 f5001983'  
]

ledportdevice = detect\_LED\_devices()  
print(ledportdevice[0])

led = LEDDriver('COM14')  
print(led)

led.open()  
print('Turning on LEDs')

# VIS  
#   
# 695,973845208697 … 738,7581953  
#  
# LED2/680nm, LED3/720nm, LED4/750nm, LED5/780nm  
# LED11/680nm, LED12/720nm, LED13/750nm, LED14/780nm  
# LED20/680nm, LED21/720nm, LED22/750nm, LED23/780nm  
#  
led.L(0b000011110000011110000011110)

# Kameran käyttö

## Kalibraatiotiedosto

Seuraavassa esimerkki kalibraatiotiedostosta.

index Npeaks SP1 SP2 SP3 PeakWL1 PeakWL2 PeakWL3 Sinv11 Sinv12 Sinv13 Sinv21 Sinv22 Sinv23 Sinv31 Sinv32 Sinv33 FWHM1 FWHM2 FWHM3

0 1 52164 0 0 554.0473012 0 0 -0.167038819 0.897291038 -1.106291748 0 0 0 0 0 0 20.36692413 0 0

1 1 52064 0 0 554.8201114 0 0 -0.116083144 0.941280185 -1.354217723 0 0 0 0 0 0 20.73087834 0 0

## Ledien valinta

Ledien valinta tehdään tämän tiedoston pohjalta:

AgMFPI-H019-SICSURFIS-Vaasa-cal-file-and-Led-selection-v01.xlsx

index Npeaks SP1 SP2 SP3 PeakWL1 PeakWL2 PeakWL3 Sinv11 Sinv12 Sinv13 Sinv21 Sinv22 Sinv23 Sinv31 Sinv32 Sinv33 FWHM1 FWHM2 FWHM3 Led 1/White LED 2/680 nm Led 3/720 nm Led 4/750 nm Led 5/780 nm Led 6/810 nm Led 7/850 nm Led 8/880 nm Led 9/940 nm Led 10/White Led 11/680 nm Led 12/720 nm Led 13/750 nm Led 14/780 nm Led 15/810 nm Led 16/850 nm Led 17/880 nm Led 18/940 nm Led 19/White Led 20/680 nm Led 21/720 nm Led 22/750 nm Led 23/780 nm Led 24/810 nm Led 25/850 nm Led 26/880 nm Led 27/940 nm

0 2 0 0 0 504,5181438 587,2723752 0 -0,801965603 -0,922764696 8,252884726 0,715143794 0,741234397 -1,240553872 0 0 0 13,53477982 16,25307694 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0

1 1 11127 0 0 584,577535 0 0 0,702570403 2,016995336 -4,32210678 0 0 0 0 0 0 15,84827112 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0

Tiedostossa on kalibraatiotiedoston rivejä vastaavat rivit ja jokaisella rivillä on lisäksi LED1…LED27 sarakkeet.

Näiden sarakkeiden sisältö pitää LED-ohjauskomentoa varten muuttaa seuraavaan muotoon:

led.L(0b000011110000011110000011110)

Tässä muodossa LEDit ovat järjestyksässä LED27…LED1.

Alla ote Python-ohjelmasta, jossa asetetaan 695 nm … 738 nm aallonpituuskaistoille sopivat LEDit päälle.

# VIS  
#   
# 695,973845208697 … 738,7581953  
#  
# LED2/680nm, LED3/720nm, LED4/750nm, LED5/780nm  
# LED11/680nm, LED12/720nm, LED13/750nm, LED14/780nm  
# LED20/680nm, LED21/720nm, LED22/750nm, LED23/780nm  
#  
led.L(0b000011110000011110000011110)

## Python-ohjelma

Seuraavassa esimerkki Python-ohjelmasta.

Python-ohjelma ajetaan komennolla:

c:\tldati\zyboz7\kamera\python>python ilkka18\_vis\_multiple.py

Esimerkki Python-ohjelma:

import fpipy as fp  
import matplotlib

from camazing import CameraList  
from spectracular.fpi\_driver import detectFPIDevices, createFPIDevice  
from spectracular.hsi import HSI

from LEDDriver import detect\_LED\_devices, LEDDriver, LEDException

LED\_IDS = [  
 # ( VID, PID) (and the same in decimal)  
 ('1FC9', '0083'), (8137, 131),  
 ]

"""Known VID:PID pairs of LED devices."""  
LED\_HWIDS = [  
 # Strings returned by read\_hardware\_id  
 '1000e016 aefba123 580267dd f5001982',  
 '10025018 af28a028 5a66a511 f5001983'  
]

ledportdevice = detect\_LED\_devices()  
print(ledportdevice[0])

led = LEDDriver('COM14')  
print(led)

led.open()

print('Turning off LEDs')  
led.L(0)

cameras = CameraList()  
print(cameras)

camera = cameras[0]  
camera.initialize()

FPI\_IDS = [  
 # ( VID, PID) (and the same in decimal)  
 ('1FC9', '0083'), (8137, 131),  
 ]

"""Known VID:PID pairs of FPI devices."""  
FPI\_HWIDS = [  
 # Strings returned by read\_hardware\_id  
 'd02b012 af380065 5b5bbeab f50019c1'  
]

fpi = createFPIDevice(detectFPIDevices(FPI\_IDS, FPI\_HWIDS)[0].device)  
print(fpi)

hsi = HSI(camera, fpi)  
print(hsi)  
hsi.read\_calibration\_file('AgMFPI-H019-SICSURFIS-Vaasa-VIS-cal-file-v02\_korjattu\_pieni\_695nm\_multiple.txt')

camera["TriggerMode"].value = "On"  
camera["TriggerSource"].value = "Software"  
camera["ExposureAuto"].value = "Off"  
camera["PixelFormat"].value = "BayerGB12"  
# camera["ExposureTime"].value = 100000  
# camera["ExposureTime"].value = 25000  
camera["ExposureTime"].value = 10000  
camera["BalanceWhiteAuto"].value = "Off"  
camera["Gamma"].value = 1  
camera["Gain"].value = 1.9382002601  
camera["GainAuto"].value = "Off"

input("Put the lens cap on")  
hsi.take\_dark\_reference()  
print(hsi.dataset.dark)

input("Take the lens cap off and set image (only for radiance)")

print('Turning on LEDs')

# VIS  
#   
# 695,973845208697 … 738,7581953  
#  
# LED2/680nm, LED3/720nm, LED4/750nm, LED5/780nm  
# LED11/680nm, LED12/720nm, LED13/750nm, LED14/780nm  
# LED20/680nm, LED21/720nm, LED22/750nm, LED23/780nm  
#  
led.L(0b000011110000011110000011110)

print('Capturing cube')  
raw = hsi.capture\_cube()

print('Turning off LEDs')  
led.L(0)

print('Calculating radiance')  
rad = fp.raw\_to\_radiance(raw)  
print(rad)

print('Extracting single frame from cube and saving to PNG')  
test = rad["radiance"]  
testdata = test.data

imagelastindex = 33

#  
# Save radiance images  
#  
print('Start saving radiance images')  
for x in range(0, imagelastindex):  
 rad1 = testdata[:,:,x]  
 matplotlib.image.imsave('rad\_' + str(x) + '.png', rad1)

import matplotlib.pyplot as plt  
plt.gray()

#  
# Save raw images and demosaic images  
#  
print('Start saving raw data')

for x in range(0, 2):

# Raw data values  
 dn1 = raw.dn.isel(index=x)  
 matplotlib.image.imsave('raw\_' + str(x) + '.png', dn1)

# Demosaic to get three colour channels  
 dm1 = fp.demosaic(dn1, 'BayerGB', 'bilinear')  
 dm1\_red = dm1[:,:,0]  
 dm1\_green = dm1[:,:,1]  
 dm1\_blue = dm1[:,:,2]

matplotlib.image.imsave('raw\_' + str(x) + '\_demosaic\_red.png', dm1\_red)  
 matplotlib.image.imsave('raw\_' + str(x) + '\_demosaic\_green.png', dm1\_green)  
 matplotlib.image.imsave('raw\_' + str(x) + '\_demosaic\_blue.png', dm1\_blue)