ADVACAM
Imaging the Unseen

PIXet EDU

Online version: https://wiki.advacam.cz/wiki/PIXet_EDU

Contents

Introduction 3
Requirements & Installation 3
Pixet Basic Interface Description 3
Title bar 3
Menu bar 3
File 3
View 4
Settings 4
Scripting 4
Quick access toolbar 5
Device list column 6
Image panel 6
Right panels 6
Image properties 6
Measurement 6
Image Info 6
Histogram 7
Spectra 7
Status bar 7
Getting started with the PIXet Basic 7
Collecting Data 7
Exporting data 9
Example scripts 10
Simpliest test 10
Simple test with error handling 10
Operation modes and calibration 11
More acq methods - Commandline version 12
Related 13

Introduction

The PIXet Basic is minimalist option of the PIXet Pro, intended to be used only with the MiniPIX-EDU device. The PIXet Pro switches to the Basic mode when the MiniPIX-EDU is connected.

Note: The MiniPIX-EDU is cheap and limited variant of the MiniPIX. It was intended for basic demonstrations like as described in the EDU Kit experiments cookbook

- Introduction video: https://www.youtube.com/watch?v=XOA1C65PZD4&t=3s
  https://www.youtube.com/watch?v=XOA1C65PZD4

Requirements & Installation

The PIXet Basic has some requirement and installation as the PIXet.

- See: PIXet: System Requirements
- See: PIXet: Installation

Pixet Basic Interface Description

1: Title bar
2: Menu bar
3: Quick access toolbar
4: Device list column
5: Image panel
6: Image properties
7: Measurement
8: Image info
9: Spectra
10: Status bar

Title bar

"Pixet Basic", {Pixet vesion}, "Build" {build_version}, (release date), "MiniPIX", {Chip ID}, "sn" {serial_number}

Menu bar

File

Open frame...

Opens a dialog that allows to open previously recorded frame files using a *.dsc file

Save data...
Opens a dialog that allows to save current frame in various file formats
Relevant file formats: .txt, .clog, .png, .tif, .tiff

Load factory config
Looks into (Pixet_folder)\factory\ and searches for MiniPIX-{chip_ID}.xml and configures the device by it.

View

Mirror Image
Toggle option, allows to display the detector window mirrored along Y axis

Rotate image
alows to rotate displayed data in detector window by 90°, 180° or 270° clockwise

Show grid
Toggle option, allows to display an overlay in detector window showing the pixel grid

Ensure aspect ratio
toggle option, keeps aspect ratio constant when resizing the Pixet window

Settings

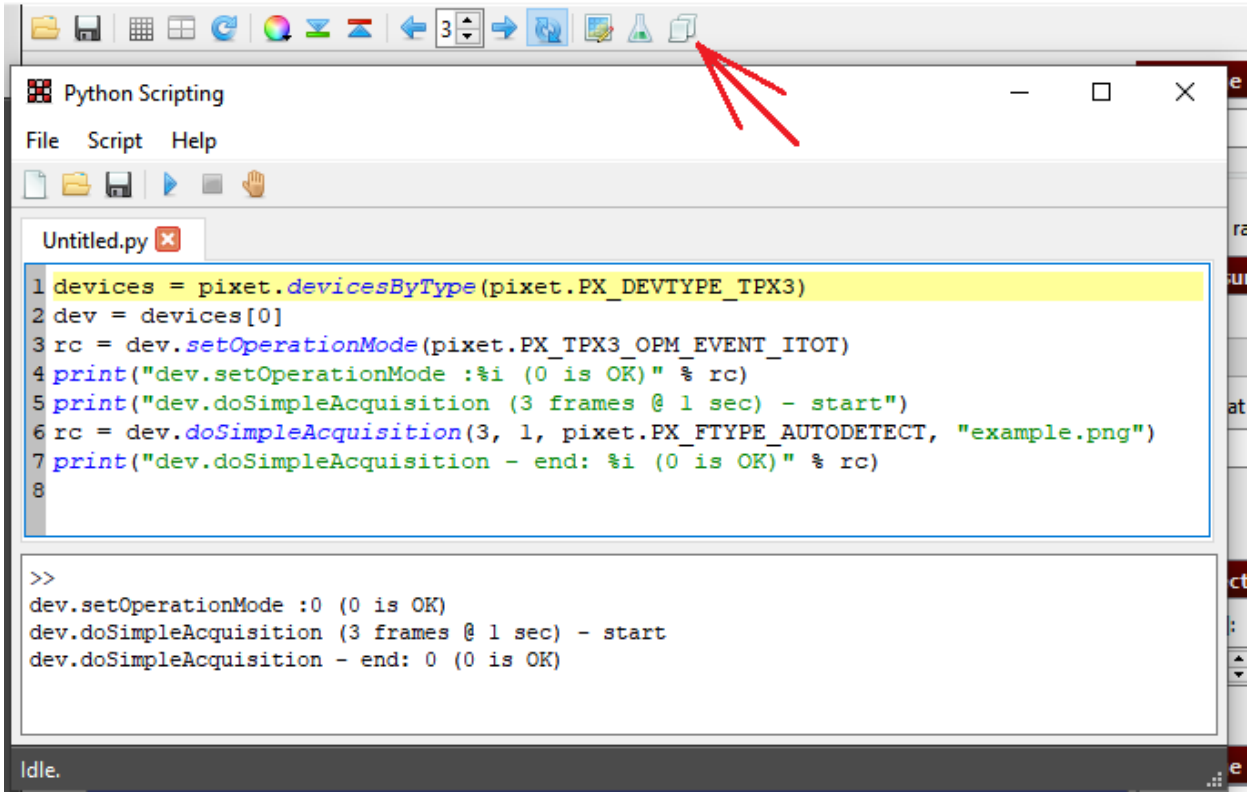
Change password for Pixet Pro
internal development option

Scripting

(Menu Scripting, or click on the icon with two papers at the toolbar.)

Python scripting opens Python scripting plugin, that enables to controll the device using Python

template_script.py runs an example script. If unaltered, performs 3x 1s acquisition and saves the frames into Pixet parent directory as .png files



Python plugin (Pixet Pro below it - Basic has fewer icons)

See

- #Example scripts
- Python API

Quick access toolbar



Quick access toolbar, left to right:

Open frame...

Save data...

Show grid

Rotate clockwise

Color scheme

Under warning

Over warning

Python scripting

Open frame... - see File --> Open frame...

Save data... - see File --> Save data...

Show grid - see View --> Show grid

Rotate clockwise - see View --> Rotate image

Color scheme
Allows selection of different colour representation for the measured data

Under warning
Toggle option, highlights pixels below min value in green

Over warning
Toggle option, highlights pixels above max value in red

Python scripting
see scripting --> Python scripting

Device list column

displays all currently connected devices (their type and chip ID)

Image panel

shows results of current measurement. Color scheme below the window can be changed using the color scheme icon. Slider at the bottom of the screen allows to browse different frames (if there are any).

Right panels

Image properties

---

allows to set min and max values for view of the current measurement

Measurement

Mode	Imaging - measures number of hits in each pixel during selected time period Tracking - measures energy of hits in each pixel during selected time period
Frames	set how many frames will be measured
Exposure [s]	set the exposure duration of each frame
Sum	if checked, sums all measured frames in the detector window
Play	starts the measurement. Automatically repeats when finished
Image	takes a single image with selected exposure time

Image Info

[X,Y]	gives X,Y coordinates of selected pixel or pixel under the mouse pointer
Count	gives number of hits in selected pixel or pixel under the mouse pointer
Min	min number of hits in image or selected area
Max	max number of hits in image or selected area
Pixel count	number of hit pixels in image or selected area
Total	total number of hits in image or selected area
Mean	mean number of hits in image or selected area

Histogram

(only if imaging mode is selected)

Spectra

(only if tracking mode is selected) Energy [keV]

Size [pix]

Timeline

Status bar

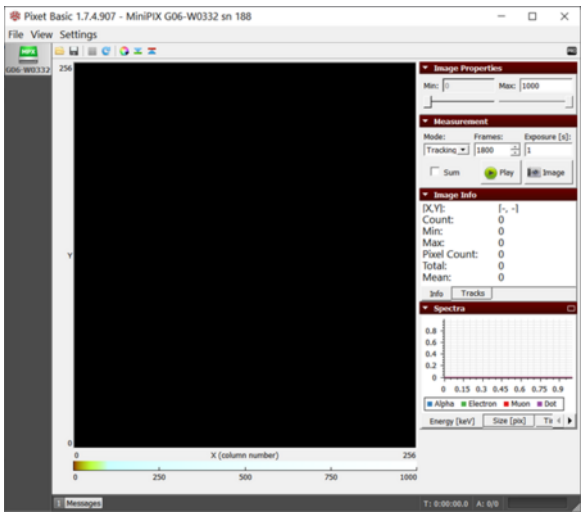
Messages	shows system messages. Broom icon = clear log. Toggle option to automatically open when new message
T	duration of current experiment
A	current/total no. of measurements. Does not work when repeating or when taking single images
Measurement progress bar	Shows progress of current measurement. Does not work when repeating or when taking single images

Getting started with the PIXet Basic

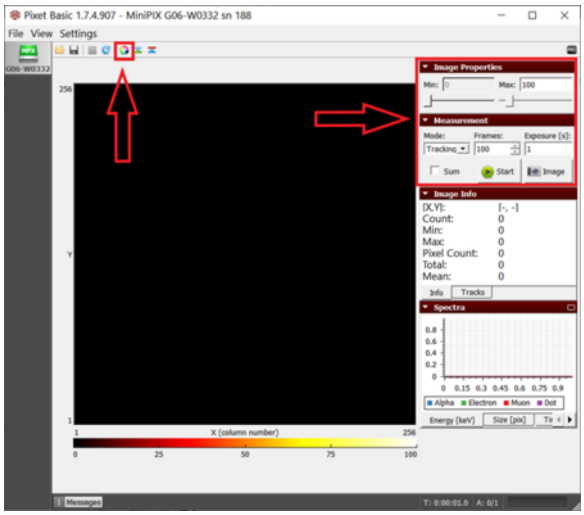
Quick intro to using the Pixet Basic program. For the Experiments cookbook, see: EDU Kit experiments cookbook

Collecting Data

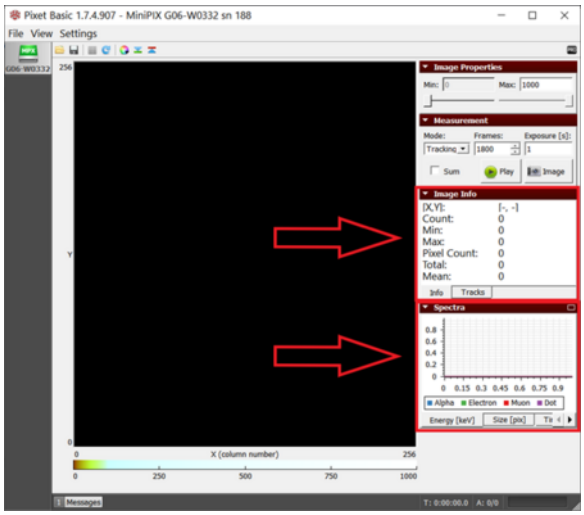
1. Connect the MiniPIX EDU camera, wait for 10 seconds and launch the software Pixet Basic.



2. The settings supposed to be modified while performing the experiments.
See right image: The two arrows show the only place to modify the settings.



3. The collected visual data will be displayed on the black screen and the additional data (counts, histograms, etc.) is displayed in the two sections- Image Info and Spectra.
See right image: The data of the *experiment will be displayed in the two sections*



4. The image info shows energy in the “Info” tab and counts of individual particles in the “Tracks” tab.

5. The spectra section shows the energy, size, and time of individual particles in respective tabs.
Note: To enlarge the spectra results, click on the rectangle at the top right corner of the spectra window.

Exporting data

1. Select the energy graph in spectra and right-click on the histogram in the spectra section and click on Save to File.

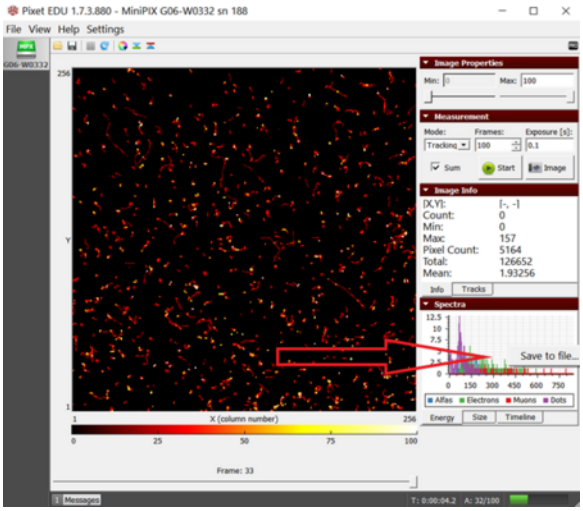


Figure 4. The histogram of collected data is being exported as ASCII

2. Enter the file name and select ASCII Vertical in Save as type drop-down menu. Click on Save.

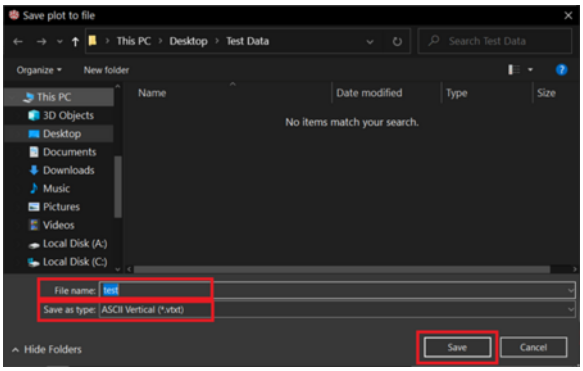


Figure 5. The type of file in the save window should be vertical to make sure that we get the whole histogram

3. Next open Excel. Click on Open > Browse and browse to the saved file. Select All Files from the drop-down menu next to the File name field, select the file and click open.

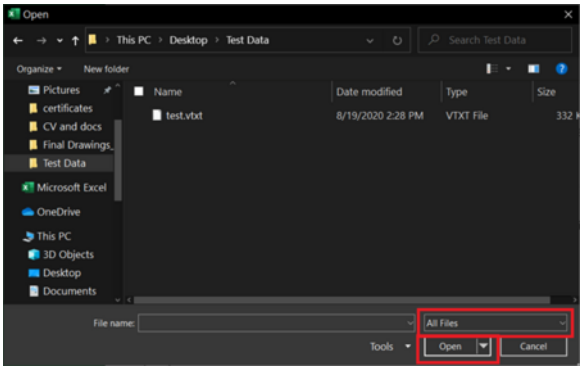


Figure 6. The Open window of excel showing the “vtxt” file

4. Select Delimited in the Text Import Wizard window and click on Finish.

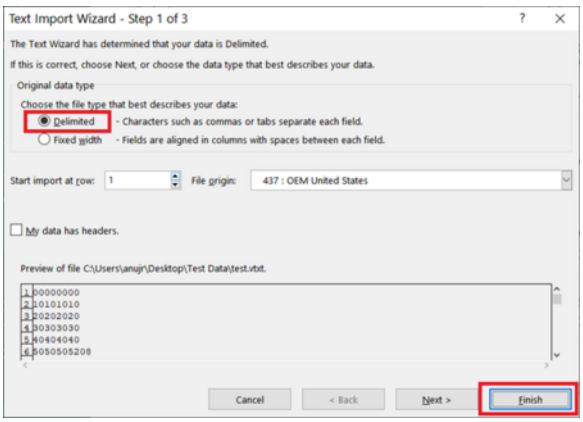


Figure 7. Excel delimiting the data to sort it in different columns

5. The first two columns of sheet are the list of the energy of alpha particles, the next two columns are for electrons, the next two are muons and the last two are dots.

6. Now select the first two columns and click on the Insert menu. In the Charts section, click on the Scatter chart and select Scatter with Straight Lines to get the histogram.

Example scripts

Simpliest test

This is the simpliest test.

- Run it from the integrated Python console
- Using first device
- Starting sigle acquisition
- Using actual operation mode
- Saving data to the Pixet directory

```
dev = pixet.devices()[0]
dev.doSimpleAcquisition(1, 1, pixet.PX_FTYPE_AUTODETECT, "example.png")
```

Simple test with error handling

Basic error handlng implemented

- Shows the process starting message.
- Shows a return code.
- If error, shows error message.

```
dev = pixet.devices()[0]
```

```
print("dev.doSimpleAcquisition...")
rc = dev.doSimpleAcquisition(1, 1, pixet.PX_FTYPE_AUTODETECT, "example.png")
print(f"dev.doSimpleAcquisition end: {rc} (0 is OK)")
if rc!=0: print(dev.lastError())
```

Operation modes and calibration

- Creates the output directory
- Shows list of devices
- Tests all combinations of OPM and ToT calibration on/off
- Measures and saves png images and txt numeric data
- Error handling
- Shows output path

Notes

- The script was intended to be run in the integrated Python console
- MiniPIX-EDU-Timepix (Other than Timepix has other OPM management)

```
import os

dirName = "test-files"
print("---------------------------------")
try:
    os.mkdir(dirName)
    print(f"Directory '{dirName}' created successfully.")
except FileExistsError:
    print(f"Directory '{dirName}' already exists.")
except PermissionError:
    print(f"Error: Permission denied: Unable to create '{dirName}'.")
except Exception as e:
    print(f"Error: An error occurred: {e}")

devices = pixet.devices()
devCnt = len(devices)
if devCnt==0 or devices[0].fullName()=="FileDevice 0":
    print("No devices found")
    devCnt = 0
else:
    print("Devices list:")

for n in range(devCnt):
    dev = devices[n]
```

```
    print(f"- {n}: {dev.fullName()}, {dev.sensorType(0)}")

dev = devices[0]

def test(opm, opmn, calib):
    rc = dev.useCalibration(calib)
    print(f". dev.useCalibration({calib}): {rc} (0 is OK)")
    if rc!=0: print(dev.lastError())
    for ext in [".png", ".txt"]:
        print(f". dev.doSimpleAcquisition(3 frames, 1 sec, '{ext}')...")
        rc = dev.doSimpleAcquisition(3, 1, pixet.PX_FTYPE_AUTODETECT, f"{dirName}/edu-
c{calib}-o{opmn}{ext}")
        print(f". dev.doSimpleAcquisition - end: {rc} (0 is OK)")
        if rc!=0: print(dev.lastError())
modes = [
    [pixet.PX_TPXMODE_MEDIPIX, "MEDIPIX"],
    [pixet.PX_TPXMODE_TOT,     "TOT"],
    [pixet.PX_TPXMODE_1HIT,    "1HIT"],
    [pixet.PX_TPXMODE_TIMEPIX, "TIMEPIX"]
]

if devCnt>0:
    pixcfg = dev.pixCfg()
    for opm in modes:
        print(f"OPM: {opm} ----------------------------")
        rc = pixcfg.setModeAll(opm[0])
        print(f". pixcfg.setModeAll({opm[1]}): {rc} (0 is OK)")
        if rc!=0: print(dev.lastError())
        for calibOnOff in [0, 1]:
            test(opm[0], opm[1], calibOnOff)
    fullPath = os.path.join(pixet.appDir(), dirName)
    print(f"See the '{fullPath}' for output files")
print("Test end")
```

More acq methods - Commandline version

Tpx version of device-tpx3-frames-manyAcqs.py:
https://advacam.com/examples/device-tpx-frames-manyAcqs.py
(For using with MiniPIX-EDU comment/delete temperature and other params unsupported on EDU)

Related

- MiniPIX-EDU
- EDU Kit experiments cookbook
- Pixet Basic introduction video: https://www.youtube.com/watch?v=XOA1C65PZD4&t=3s
  https://www.youtube.com/watch?v=XOA1C65PZD4
- The PIXet Pro program