miniPIXdaq: Data acquisition for *miniPIX EDU* pixel detector

The miniPIX EDU is a camera for radiation based on the Timepix pixel read-put chip with 256x256 radiation-sensitive pixels of 5x55µm² area and 300µm depth each. The chip is covered by a very thin foil to permit α and β radiation to reach the pixels. The device is enclosed in a sturdy aluminum housing with a USB 2.0 interface.

Other than single semi-conductor chips or simple Geiger counters, this device provides two-dimensional images of particle traces in the sensitive detector material. The high spatial resolution compared to the typical range of particles in silicon is useful to distinguish the different types of radiation and measure their deposited energies. α-particles are completely absorbed and deposit all of their energy in the sensitive area, allowing usage of the device as an energy spectrometer.

The vendor provides a ready-to-use program for different computer platforms as well as a software-development kit for own applications.

The code provided here is a minimalist example to read out single frames, i.e. a set of 256x256 measurements accumulated over a given, fixed time interval. Each frame is displayed as an image representation with a logarithmic color scale representing the deposited energy. The analysis of the recorded signals, i.e. clustering of pixels, energy determination and visualization, is achieved with standard open-source tools for data analysis. It is therefore well-suited to give high-school or university students detailed insights.

Getting ready for data taking

- get the code from gitlab @ kit
  `git clone https://gitlab.kit.edu/Guenter.Quast/mPIXdaq`

  This repository includes the python code and a minimalistic set of libraries provided by ADVACAM. `cd` to the `mPIXdaq` directory you just downloaded
- set up the USB interface of your computer to recognize the miniPIX EDU:
  `sudo install_driver_rules.sh` (to be done only once), then connect the *miniPIX EDU* device to your computer

Now everything is set up to enjoy your miniPIX EDU. Just run the *Python* program:
`./run-mPIXdaq.py`

Note: on some systems the current directory, ".", needs to be in the `LD_LIBRARY_PATH` so that the *Python* interface *pypixet* finds all *C* libraries. This is achieved by the first line in the Python script `run_mPIXdaq.py`. Starting the *Python* script by a different mechanism may not work without adjusting the environment variable `LD_LIBRARY_PATH`. In the *bash* shell, this is achieved by `export LD_LIBRARY_PATH=.` on the command line or in the initialization script.

Running the example script

Available options of the *Python* example to steer data taking and data archival to disk are shown by typing

`./run_mPICdaq.py --help`; the output is shown below:

```
usage: run_mPIXdaq.py [-h] [-v VERBOSITY] [-o OVERLAY] [-a ACQ_TIME] [-c ACQ_COUNT] [-f FILE]
                      [-t TIME] [--circularity_cut CIRCULARITY_CUT] [-r READFILE]

read, analyze and display data from miniPIX EDU device

options:
  -h, --help            show this help message and exit
  -v VERBOSITY, --verbosity VERBOSITY
                        verbosity level (1)
  -o OVERLAY, --overlay OVERLAY
                        number of frames to overlay in graph (25)
  -a ACQ_TIME, --acq_time ACQ_TIME
                        acquisition time/frame (0.2)
  -c ACQ_COUNT, --acq_count ACQ_COUNT
                        number of frames to add (1)
  -f FILE, --file FILE  file to store frame data
```

```
-t TIME, --time TIME  run time in seconds
--circularity_cut CIRCULARITY_CUT
                      cicrularity cut
-r READFILE, --readfile READFILE
                      file to read frame data
```

The default values are adjusted to situations with low rates, where frames from the *miniPIX* with an integration time of `acq_time = 0.2` are read. For the graphics display, `number_of_buffers=25` recent frames are overlaid, leading to an integration time of 5 s. These images represent a two-dimensional pixel map with a color code indicating the energy measured in each pixel.

Data analysis consists of clustering of pixels in each frame and determination of cluster parameters, like the number of pixels, energy and circularity. The threshold on circularity is controlled by the parameter `circularity_cut` ranging from 0. for perfectly linear to 1. for perfectly circular clusters. Technically, the covariance matrix of the clusters is calculated, and the circularity is the ratio of the smaller and the larger of the two eigenvalues of the covariance matrix. This simple procedure already provides a good separation of α and β particles and of isolated pixels not assigned to clusters. The latter ones have a high probability of being produced in interactions of photons, while electrons from β radiation or from photon interactions produce tracks, and α particles produce large, circular clusters due to their very high ionization loss in the detector material.

To test the software without access to a miniPIX EDU device or to a radioactive source, a files with recorded data are provided. Use the option `--readfile data/BlackForestStone.npy.gz` to start a demonstration. Note that the analysis of the recorded pixel frames is done in real time and may take some time on slow computers.

Implementation Details

The data acquisition is based on the function *doSimpleIntegralAcquisition()* from the *ADVACAM Python* API, since more advanced modes are not available for the Medipix2 chip of the miniPIX EDU. A fixed number of frames (*qcq_counts*) with an adjustable accumulation time (*acq_time*) are read from the miniPIX device and added up.

The chosen readout mode is *PX_TPXMODE_TOT*, where “ToT” means “time over threshold”. This quantity is used as a very good
measure proportional to the deposited energy.

Note that this is highly non-linear for small signals near the detection threshold of the miniPIX, but becomes more linear at high values. Calibration constants are stored on the miniPIX device for each pixel, which are used to provide deposited energies per Pixel in units of keV.

The relevant libraries for device control are provided in directories `advacam_<arch>` for `x86` Linux, `arm32` and `arm64` and for Macintosh systems. The contents of a typical directory is:

```
__init__.py   # package initialization
pypixet.so    # the Pixet Python interface
minipix.so    # C library for pypixet
pxcore.so     # C library for pxpixet
pixet.ini     # initialization file, expected in same directory as pypixet
factory/      # initialization constants
```

Note that the copyright of these libraries belongs to ADVACAM. The libraries may be downloaded from their web page, ADVACAM DWONLOADS. They are provided here as a python package for convenience.

Data Analysis

The analysis shown in this example is intentionally very simple and based on standard libraries and functions. Pixels are clustered with the *scipi.cluster.DBSCAN* (Density-Based Spatial Clustering of Applications with Noise). The shape of the clusters is determined from the ratio of the smaller and the larger one of the two eigenvalues of the covariance matrix, which is calculated from the *x* and *y* coordinates of the pixels in a cluster. For circular clusters, as produced by α radiation, this ratio is close to one, while it is almost zero for the longer traces from β radiation.

The figure below shows the graphical display of the program and the typical distribution of the pixel and cluster energies and the number of pixels per cluster. The source used was a weakly radioactive stone from the Black Forest containing

a small amount of Uranium and its decay products. The pixel map shown in the figure was sampled over a time of two seconds. The histogram in the lower-right corner shows how cluster types discriminate different types of radiation.

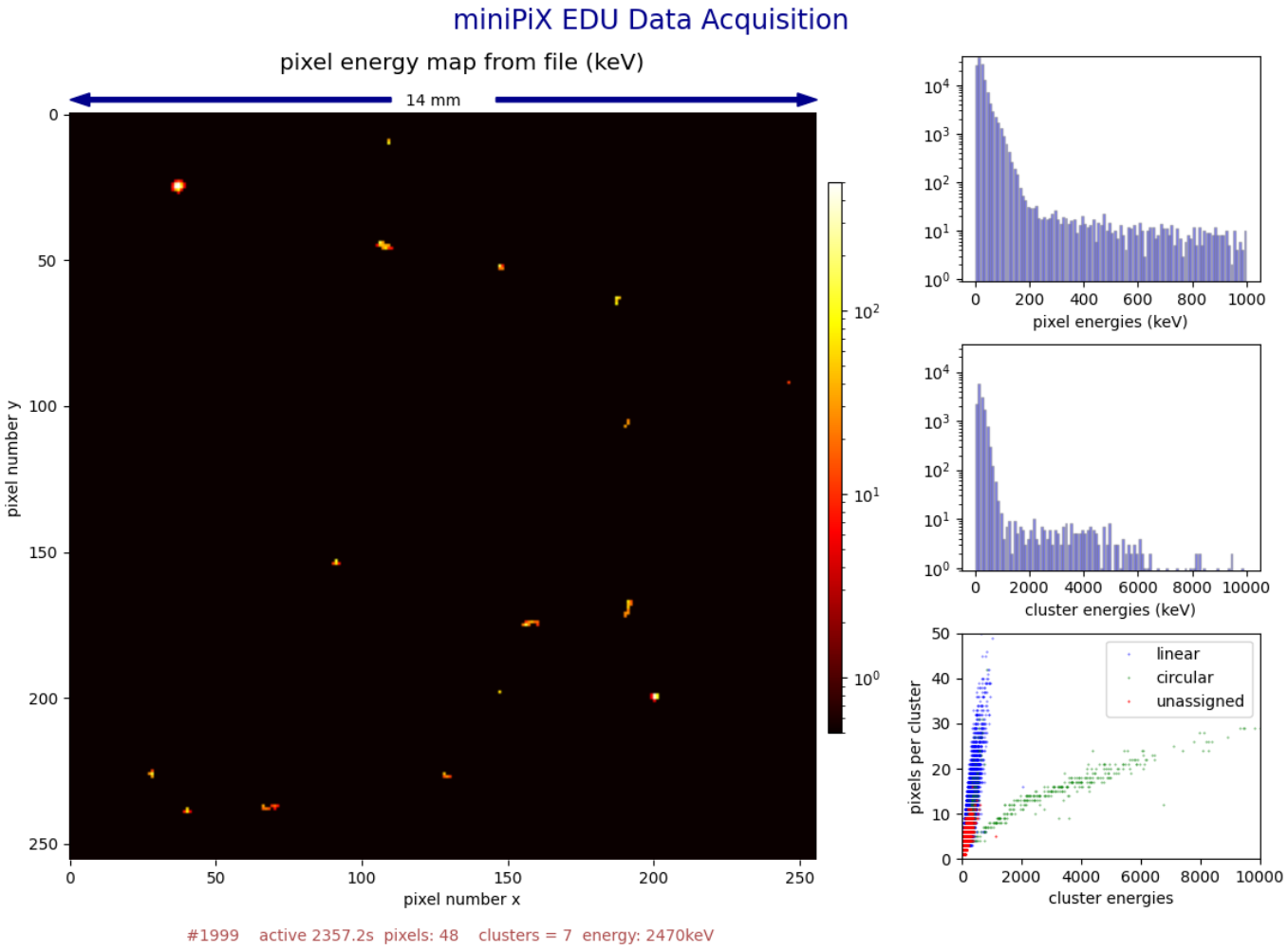

Abbildung 1: The graphical display of miniPIXdaq

Sensor Details

The miniPIX EDU is based on the *Timepix" hybrid silicon pixel device, consisting of a semiconductor detector chip segmented into 256 x 256 square pixels with a pitch of 55 mm that is bump-bonded to the readout chip. Each element of the pixel matrix is connected to its own preamplifier, discriminator and digital counter integrated on the readout chip.

The built-in Medipix2 variant of the chip is operated in the so-called "frame mode", i.e. all pixels are read out at the same time, providing one frame consisting of the deposited energies per pixel collected during the acquisition time. If operated in time-over-threshold (ToT) mode, returned pixel readings represent the time the signal is over a given threshold in counts of the chip clock (appr. 10 MHz). *ToT* is linearly related to the energy deposition for large deposits exceeding 50 keV. The functional dependence on the deposited energy E, including threshold effects, is approximated by the following function

$$ToT = a\,E + b - c/(E - t)$$

Approximate values of the calibrations constants are a = 1.6, b=23, c=23 and t=4.3. Each pixel has its individual calibration stored on the chip, which is optionally applied. The calibration is reliable up to pixel energies of one MeV. Higher pixel energies may result when frames with short acquisition time are summed up. For details, see the article by *J. Jakubek, "Precise energy calibration of pixel detector working in time-over-threshold mode", NIM A 633 (2011), 5262-5265*.