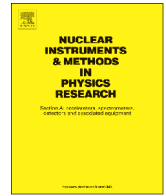




ELSEVIER

Contents lists available at ScienceDirect

Nuclear Instruments and Methods in Physics Research A

journal homepage: www.elsevier.com/locate/nima

On-line and off-line data analysis for the EUSO-TA experiment



Lech Wiktor Piotrowski ^{a,*}, Marco Casolino ^{a,b}, Livio Conti ^c, Toshikazu Ebisuzaki ^a,
 Claudio Fornaro ^c, Yoshiya Kawasaki ^a, Yusuke Hachisu ^a, Hitoshi Ohmori ^a,
 Cristian De Santis ^b, Kenji Shinozaki ^{d,a}, Yoshiyuki Takizawa ^a, Yoshihiro Uehara ^a

^a RIKEN, Wako, Japan^b INFN & Univ. Rome Tor Vergata, Rome, Italy^c International Telematic University UNINETTUNO, Rome, Italy^d Institute for Astronomy and Astrophysics, Kepler Center, University of Tübingen, Sand 6, D-72076 Tübingen, Germany

ARTICLE INFO

Article history:

Received 30 April 2014

Received in revised form

26 August 2014

Accepted 26 August 2014

Available online 15 November 2014

Keywords:

EUSO-TA

JEM-EUSO

Ultra-High Energy Cosmic Rays

Space instrument

Extensive Air Shower

Fluorescence telescope

ABSTRACT

We show the principles of the communication protocol, on-line calibration, off-line data format as well as basic visualisation and data analysis software implemented for the EUSO-TA on-ground experiment, being the first step towards implementation in a future space based mission. EUSO-TA is an on-ground detector for measuring UV (290–430 nm) light from extensive air showers induced by cosmic rays. It is a prototype experiment for the JEM-EUSO space-borne mission, built according to the same constraints of low mass, low power consumption and thus low computing power. Nevertheless, it needs to process a huge amount of data in short time, taking 2.5 μ s exposures for 2304 channels.

The low processing power and high time resolution require an efficient communication protocol and simple yet powerful algorithms for on-line analysis. The off-line data format is designed for storing a huge amount of data, at the same time allowing easy access, analysis and sharing. Its structure is scalable and adjustable to different experimental designs. It is independent of the data origin, whether it is hardware or a Monte-Carlo simulator. Use of object-oriented techniques and the ROOT framework allows rapid development of dedicated analysis software, such as a Python based quick-view program described herein. Basic capabilities of the software, such as display of the focal surface, light curves and calibration data are shown in this paper.

© 2014 Elsevier B.V. All rights reserved.

1. Introduction

Cosmic rays have proven to be an important area of science since their discovery at the beginning of 20th century by Wulf [1], Kolhrster [2], Pacini [3] and Hess [4]. Further research has extended our knowledge about this phenomenon significantly, revealing most of the energy spectrum, composition and sources of this radiation. However, the extremely high energy part of the energy spectrum remains mysterious. We are still not sure of the physics beyond the GZK cutoff energy and what in the Universe can accelerate particles to such huge velocities.

The main challenge in answering these questions is the very low flux of the most energetic cosmic rays reaching Earth. If one considers protons, nuclei and electrons, the rate for 10^9 eV particles is 10,000 per $\text{m}^2 \text{s}$, but above 10^{20} eV it drops to 1 per $\text{km}^2 \cdot 100 \text{ yr}$, which makes study in the highest energy regimes extremely difficult. To obtain a statistically significant result in an average experiment lifespan of 10–20 years, one has to monitor at least thousands of

square kilometres of ground. Experiments dedicated to this task, such as Pierre Auger Observatory in Argentina [5] and Telescope Array (TA) in the United States [6] have been constructed and have already yielded important results.

These detectors observe extensive air showers induced by the primary cosmic ray, with a hybrid method. They directly detect some of the secondary particles hitting surface based scintillation or water Cherenkov detectors and indirectly observe the cascade development with wide field of view (FoV) UV telescopes, measuring the fluorescence light produced by excitation of atmospheric nitrogen by the shower. The ground surface coverage of these facilities is limited by the possible extension of the detector network. The coverage of UV telescopes is related to the volume of the atmosphere containing showers that they can observe and limited mainly by the telescope FoV, shower proximity and energy. Introducing observations of shower-induced UV light from orbit drastically increases the observed volume of the atmosphere and is the main idea behind the Extreme Universe Space Observatory on the Japanese Experimental Module (JEM-EUSO) experiment.

The key part of the JEM-EUSO experiment is a UV telescope [7], consisting of a curved focal surface assembled from 4932 Multi-Anode Photomultiplier Tubes (MAPMTs) [8] with 64 pixels each, grouped by

* Corresponding author.

E-mail address: lech.piotrowski@riken.jp (L.W. Piotrowski).



Fig. 1. In the foreground, the EUSO-TA telescope in the final stage of assembling at the Black Rock Mesa site of the Telescope Array (TA) experiment. In the background, the housing for the TA fluorescence telescopes.

fours into the so-called EC-UNITS, which then are grouped by nines into Photodetector Modules (PDMs). This, in combination with a Fresnel lens based optics (three double sided, curved circular Fresnel lenses with 2.65 m maximal diameter, cut on two sides to 1.9 m minimal diameter), enables a $\pm 30^\circ$ FoV and 4.5' angular resolution. The detector converts photons into electrical impulses with ~ 2 ns width, which are then counted by the electronics during a single exposure of 2.5 μ s – the so-called Gate Time Unit (GTU).¹ Additionally, in case of a signal strong enough to saturate the pixels in photon counting mode, a KI mode – charge over time integration – is used. The KI consists of 8 channels per MAPMT [9]. The apparatus will be attached to the Japanese Experimental Module on the International Space Station [10]. Orbiting at an altitude of ~ 400 km, the resulting observational aperture is 1.4×10^5 km² on the ground, which is larger by a factor of ~ 46 than the Pierre Auger Observatory surface array. Changing the inclination of the telescope can further increase the aperture. These observational parameters give JEM-EUSO a chance to significantly extend the scientific knowledge in the area of Ultra-High Energy Cosmic Rays (UHECR) physics [11].

The JEM-EUSO launch will be preceded by prototype experiments, such as EUSO-TA, which will be used for testing the subsystems and estimating the performance of the final mission.

The EUSO-TA focal surface is equipped with one PDM composed of 36 MAPMTs, behind a set of two 1 m² Fresnel lenses, resulting in a $\pm 4^\circ$ FoV covered by 2304 pixels [12]. In future the FoV may be expanded with the addition of another PDM. The EUSO-TA on-ground experiment is being installed in Utah, United States at the site of Telescope Array (Fig. 1), which allows for comparing results from this limited focal surface and smaller lenses with the Telescope Array experiment.

The prototype experiments share the constraints of the final space-borne mission, such as requirements for low power consumption, low mass and high reliability. The detection system used in the prototypes is in general a scaled-down version of the JEM-EUSO system. The on-line and off-line analysis software, which is the main topic of this paper, has to be scalable and independent on the detector size. The on-line part must work efficiently with the available low processing power. The off-line analysis software must contain all the data coming from the experiment in an easily accessible, fast and reliable format,

¹ The 2.5 μ s length of the GTU roughly equals the time of movement of a shower through one pixel in the current optics design. This unit is a compromise between data stream size and shower reconstruction capabilities. Single EAS development occupies several dozens of GTUs.

allowing for a near real-time monitoring of the system and use of external tools.

2. Communication protocol for commands and data

The UV light reaching the telescopes is focused by the Fresnel lenses onto the focal surface and is measured by the MAPMT. The data stream handled by an electronics subsystem increases with each level of processing. Data from six MAPMTs is read-out by a single EC-ASIC (Elementary Cell Application Specific Integrated Circuit) [13] unit. Signal from six units is transferred to a Photodetector Module (PDM) and then to the Control Cluster Board (CCB) [14] and the Central Processing Unit (CPU) [15] (Fig. 2). The CPU merges the data acquired from all subsystems and stores it on a hard drive for off-line analysis (or prepares the data for sending to Earth in case of JEM-EUSO). A reverse flow of information is also required, where higher-level modules send requests for specific tasks to lower-level modules and retrieve feedback.

The exchange of information is made with encapsulated packets, organised in a “matroska” style. The amount of information contained in a packet increases with the level of processing (Fig. 3). A single packet of stored information contains photon counts for all photon counting channels (pixels) and KI channels, 1 Byte per channel, for 128 GTUs – around 332 KB for a single PDM consisting of 2304 photon counting pixels. That results in a data stream of about 1 GB/s of raw measurement data, increased slightly by additional information needed to be stored. It is difficult to estimate now how much will EUSO-TA work in a continuous data storing mode and how much in “trigger” mode, where only 1 or 2 packets of data are stored surrounding an interesting event.

Each processing unit has a C language structure for storing the information it provides. Pointers to the structures of the lower level units are stored in the higher level unit structure, which adds its own data. For conveying commands, packets containing appropriate C language structures are sent down from higher level units, each lower level unit ensuring that the request gets to all the requested subunits.

3. Data acquisition process

The EUSO-TA apparatus can work in two data acquisition modes. In the default mode it reacts to the internal trigger based on a multi-level algorithm constantly analysing incoming data. The Level 1 trigger is the fast trigger issued by the FPGA of the PDM board, which eliminates most of the background fluctuations by requiring a locally persistent signal above average background, lasting a few GTUs [7]. The more sophisticated Level 2 trigger is issued by the CCB. It follows the movement of the EAS spot inside the PDM over a predefined time window to distinguish the unique pattern of an EAS from the background [14].

The second acquisition mode is a reaction to an external trigger, which can be used for working in coincidence with other experiments. This is important for the EUSO-TA telescope which has very limited self triggering capabilities due to a small ($\pm 4^\circ$) FoV.² In the current configuration, the external trigger is received by a time synchronisation subsystem called clockboard and is directly passed to the CCB, with latency of order of microseconds. Additionally, the CPU may request snapshots of data or issue triggers to lower level subsystems on its own.

Obtaining proper measurements requires prior calibration of the apparatus, to properly count the number of photons hitting each

² Efficient distinction between false events and real extensive air showers requires an image of the shower on several exposures. The FoV and distance to showers of EUSO-TA makes it highly unlikely for the shower to appear on more than 1–2 images.

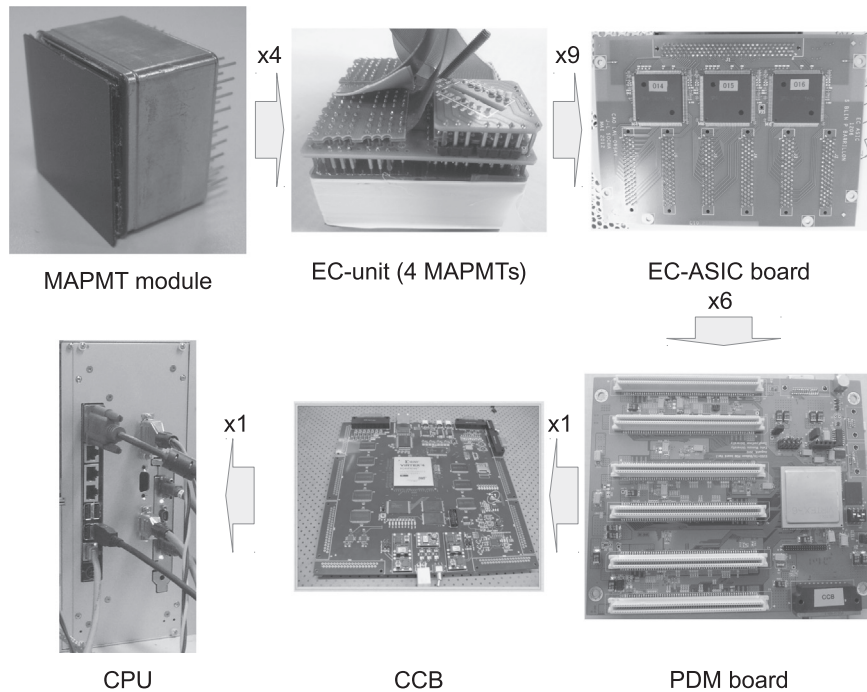


Fig. 2. Scheme of the signal flow in the EUSO-TA experiment. The labels over arrows show signal from how many subsystems of the previous level (the left or top side of an arrow) is included in the subsystem of the next level (the right or bottom side of an arrow). The subsystem multiplicity from the PDM board to the CCB and from the CCB to the CPU for the full JEM-EUSO mission are 8 and 18 respectively.

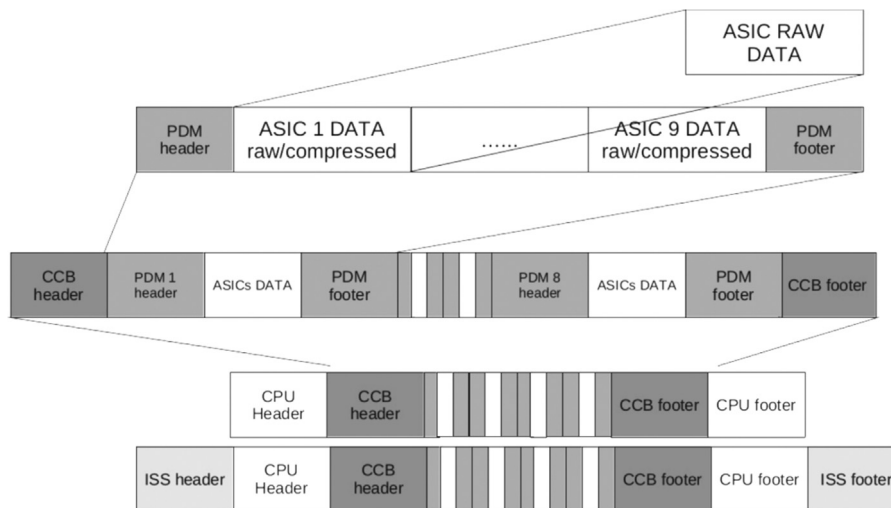


Fig. 3. A scheme of the “matroska” style packet organisation in JEM-EUSO experiments. Data packets from lower level subsystems are joined and encapsulated into higher level subsystem packet.

pixel. This requires estimating properties of pixels such as pixels' response to light functions and the efficiencies of the photon to signal conversion achieved with chosen settings. The first task is the absolute, off-line calibration which allows converting the number of counts in the electronics into the real photon flux. This requires laboratory measurements. The second task can be performed on-line and is designed for optimising the discrimination threshold between noise and photoelectrons, which varies as the instrument ages. For this task the CPU requests and analyses specific data for each pixel, which will be explained in more detail in Section 4. The frequency with which the on-line calibration will be performed has not yet been decided and will depend on the behaviour of subsystems.

Except acquiring data on a trigger and calibration request, the data acquisition process includes checking the status of the

apparatus elements and reaction to alarms concerning them – the so-called housekeeping. The final stage of the data acquisition process in EUSO-TA is storing the files onto a hard drive.

For the purpose of testing and debugging of the EUSO-TA system, two other acquisition modes are available – manual and semi-automatic. The manual mode simply sends low-level, hexadecimal commands to the CCB and clockboard subsystems through a CPU command line interface. This mode is used mainly for testing the functioning and communication between subsystems. The semi-automatic mode is designed for testing the cosmic-ray event data gathering and calibration. It is driven through a command line program invoked with parameters specifying the data acquisition mode, requested number of packets, some adjustable hardware parameters, etc.

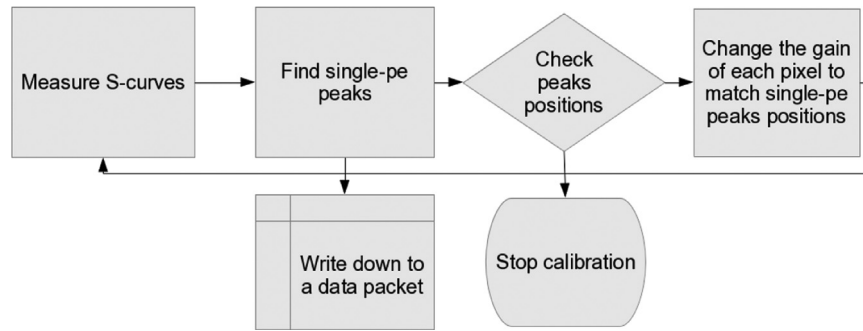


Fig. 4. The general diagram of the on-line calibration algorithm.

4. On-line analysis of data

Due to the power consumption and mass restrictions for the space JEM-EUSO mission, the CPU of EUSO experiments has very limited processing capabilities. Therefore, the on-board software has to be characterised by low memory, low processing power and other resource requirements. To fulfil this demand, the on-line data analysis and management software is written in C/C++ using simple libraries only. On the other hand, the software must be fast, reliable and scalable to fulfil the scientific objectives of the full space mission.

4.1. Calibration

Most of the data getting past the Level 1 and 2 triggers is intended for off-line analysis. However, as mentioned above, standard data gathering modes require prior calibration. The off-line, absolute calibration [16] is needed for obtaining the number of photons hitting a pixel from the number of registered counts. The goal of the on-line calibration is to obtain a similar overall performance for the single photon detection for all apparatus' pixels by adjusting the preamplifier gain. In general, we want to reject the noise from the PMT and EC-ASIC electronics and accept more intense signals coming from a single photon. This requires analysis of counts vs. Digital to Analogue Converter (DAC) threshold for single photoelectron detection (the so-called S-curves), which is the main computational task for the data analysis of the CPU. This involves an iterative process of sending commands and acquiring data from lower level subsystems (Fig. 4).

On-line calibration involves finding a single photoelectron peak in the derivative of an S-curve (single photoelectron spectrum, Fig. 5), which due to time constraints and thus low statistics may be noisy. Therefore, more sophisticated, noise suppressing algorithms have to be used. Currently, we have obtained the best results using a Super Lanczos low-noise differentiator [17], but better procedures may be implemented in the future. The procedure of peak finding is simply locating the highest value of the “single photoelectron spectrum” which lies after the single photoelectron valley (the smallest value after the noise peak).

4.2. Event data

The calibration algorithm operates on data packets that contain the average of GTUs taken for each threshold setting for each pixel of the focal surface. The optimal number of GTUs in an average is a compromise between data taking time and the statistical uncertainty and is estimated³ to be of order of 10,000. In the case of real event data, the packet format is slightly different – it contains the count of

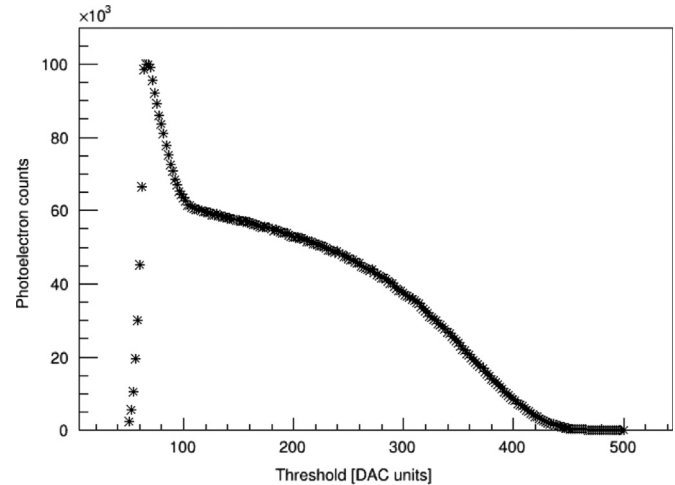


Fig. 5. A sample S-curve – a sum of photon counts for a number of measurements over the threshold for a single photoelectron detection (in Digital to Analogue Converter (DAC) units).

photons detected by each pixel of the focal surface in each GTU. Additionally, data from the KI is also supplied. The event data packets, each containing data for 128 consecutive GTUs, are added sequentially to a file, which also contains calibration packets as well as commands exchanged between the subsystems and other information.

5. Off-line data format

Sequential storing of binary packets coming directly from subsystems is very efficient in terms of writing speed, memory use, and bandwidth conservation. However, it is very inefficient when accessing random parts of data, a basic task in data analysis. Moreover, in case of any change in the format or experiment design, different data sets become incompatible and maintaining access to them becomes a very complicated task. Therefore, for data sharing and analysis purposes a different format has to be introduced. The requirements are as follows:

1. Fast access and search capabilities.
2. Ease of use.
3. Ease of sharing and distribution.
4. Popularity in the scientific community that will use the data.
5. Ability to handle very large data sets.

While requirements 1–3 can be fulfilled by several data formats, it seems that in the UHECR community the requirements 4 and 5 are fulfilled only by the CERN ROOT package [18] TTree format. More importantly, the ROOT framework is already used by most of the existing JEM-EUSO off-line analysis software. Therefore, the TTrees

³ The optimal signal for calibration is estimated to be around 1 photon for 100 GTUs, which gives a statistical uncertainty of around 10% for a point in an S-curve, fluctuations acceptable for single photoelectron peak determination purposes.

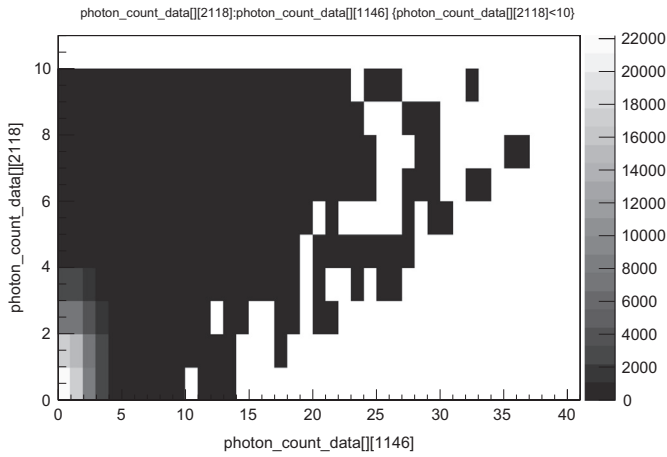


Fig. 6. An example histogram of dependence between two pixels value (test measurement) – a result of a single line command in the ROOT framework. Modification of the z-axis or labels requires additional commands or using the GUI.

were chosen as the basic format for data sharing and analysis in the JEM-EUSO collaboration.

The program for the conversion from hardware format to TTrees simply analyses all the packets in the file sequentially converting them and assigning them to separate TTrees for calibration, event data, etc. Then the task of obtaining, for example, measurements for a specific pixel is a matter of opening the file and invoking a simple command, such as:

```
root -l acquisition_laserrot.root
root [0] tevent->Scan("photon_count_data[[2118]]");
*****
* Row * Instance * photon_co*
*****
*      0 *          0 *          2 *
*      0 *          1 *          0 *
*      0 *          2 *          0 *
*      0 *          3 *          0 *
*      0 *          4 *          0 *
*      0 *          5 *          0 *
*      0 *          6 *          1 *
```

while, for example, drawing histogram of dependence between two pixels values with some cuts on one of the pixel values is equally simple (Fig. 6):

```
root [1] tevent ->Draw("photon_count_data[[2118]]:
photon_count_data[[1146]]",
"photon_count_data[[2118]] < 10", "colz");
```

The speed of conversion to the TTrees, as well as efficiency of the format allows for the development of a real-time monitoring system. The object-oriented design, C++ syntax and Python bindings make the development of an analysis software convenient. The experiment's TTrees have a versatile and scalable design, easily accommodating different experimental designs. These features make the format transparent – the data can be analysed in exactly the same way no matter if it was gathered in a real experiment or generated by a Monte Carlo simulator. The EUSO-TA TTrees include simple types only, therefore no ROOT dictionaries have to be distributed with files, which greatly improves the simplicity of analysis of the data.

6. Graphical data control and analysis

The data stored in TTrees can be read and visualised by the EUSO-TA quick-view software specially designed for this task. This allows for quick evaluation of the data acquisition status,

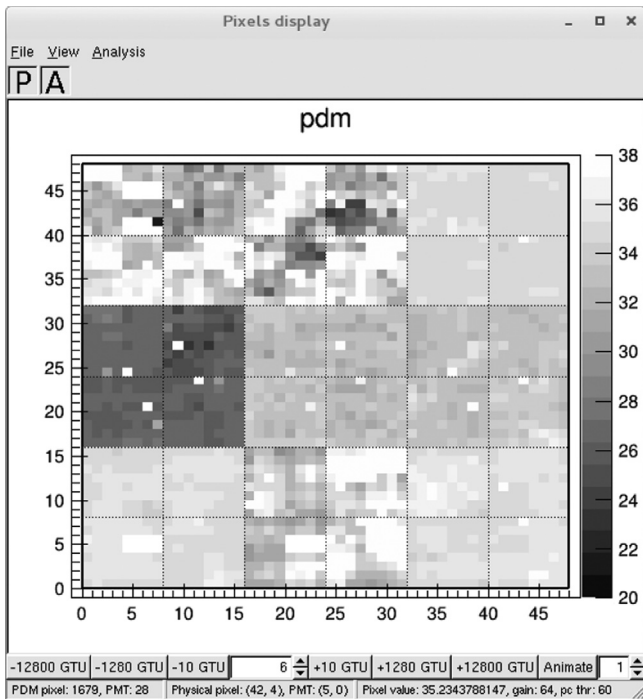


Fig. 7. Screenshot of the main window of the basic visualisation program, showing (not calibrated) photoelectron counts (grey scale) for a single PDM for a single GTU. The labels on x and y axes denote the pixel numbers.

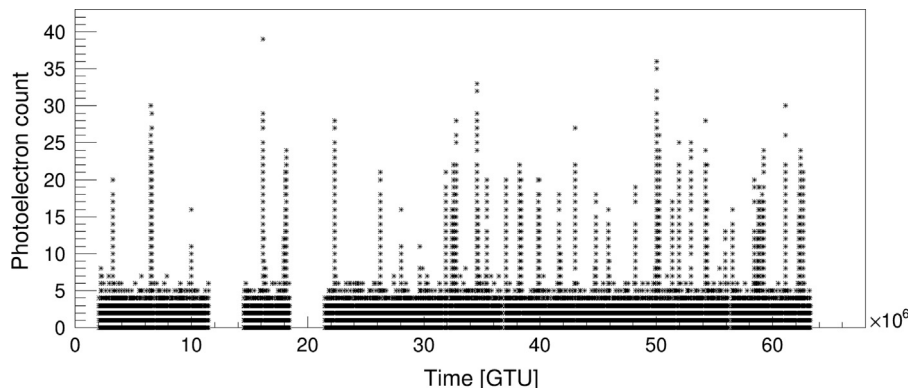


Fig. 8. Single pixel analysis. Top: the light curve (number of photoelectron counts vs. time) of a ~ 1 min observation of a spot from a moving laser projected on a wall. The spot was periodically entering and exiting the field of view. Bottom: the histogram of the photoelectron counts in the pixel.

being one of the key elements in the system control. The main requirements for the software are as follows:

1. Convenience of use.
2. Simplicity of distribution and installation.
3. Simplicity of adding new features.

For convenience of use, the program functionality can be divided into two parts, dedicated to display and analysis of standard events acquisition data and of calibration data. The software has been written in Python with ROOT bindings – PyROOT. That allows for rapid development and easy use of an extensive library of Python scientific packages, such as NumPy or SciPy along the built-in ROOT functions. This is especially important for a program with basic data analysis capabilities – many of user requirements are not known at the time of designing the software and often need to be quickly implemented and used during hardware development or data acquisition sessions. In addition to different features required by different developers of the experiment, use of a program in different environments (different distributions of Linux, MacOSX and Windows just to name the most popular ones) is often problematic. We avoided this issue by the use of ROOT graphic user interface elements, which make the program almost platform independent, requiring only a properly configured installation of ROOT and Python.

6.1. Event analysis mode

The main task of the quick-view program is to provide a quick look into the data, displaying the photoelectron counts on the focal surface (Fig. 7). This allows an instantaneous discovery of errors related to the light detection. The main window allows selecting which GTU's data should be displayed and tracing the changes on the focal surface with time. A specific GTU can be selected, or the GTUs can be changed sequentially manually or as an animation. The software provides both geometrical and electronics coordinates of the pixels in real-time. The display is the ROOT 2D histogram, therefore all the standard ROOT methods are accessible, making it easy to zoom to a selected part of the focal surface, changing the colour scale of the histogram, fitting a mathematical function, smoothing, etc. The histogram can be saved in both graphical and the framework's format for further analysis with additional tools.

Clicking on a pixel brings up that pixel's analysis window. Currently, it includes the pixel's light curve (plot of photoelectron counts over time) and the photoelectron counts histogram. The latter allows estimating the background level or detecting

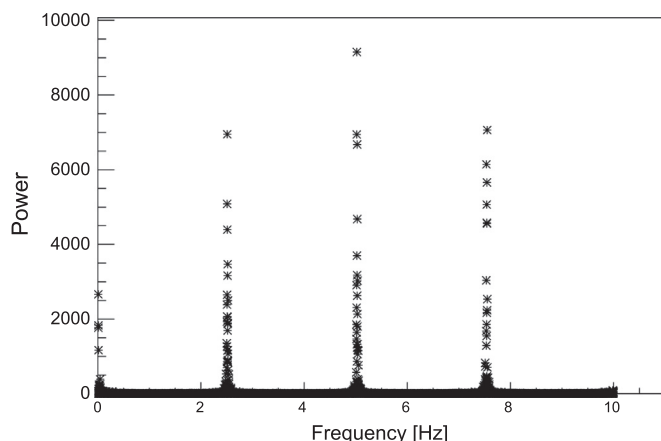


Fig. 9. A Lomb-Scargle periodogram of the light curve (Fig. 8) of a moving laser spot.

anomalies in pixel behaviour. The former is best suited for analysis of sources variable in time. Fig. 8 shows the display window with data from a test observation of a spot generated by a moving laser projected onto a wall. The increase of intensity when the laser was moving through the FoV of the experiment is clearly visible.

The pixel statistics histogram or the light curve can be fitted with any of the standard ROOT accepted functions. If requested, a Lomb-Scargle periodogram [19] of the displayed light curve can be calculated and displayed (Fig. 9), which in this case shows clear harmonics of the laser moving frequency. Currently, the periodogram is calculated using `scipy.signal.lombscargle` from the Python SciPy package [20].

6.2. Calibration analysis mode

When working with the calibration data, the main window displays the average photoelectron count for the focal surface pixels for a given threshold. In this case, a specific threshold can be selected, changed by fixed values or animated. This has already proven to be a valuable tool for detecting errors in electronics by analysing the anomalies of pixel response to light.

Similar to the behaviour in event analysis mode, clicking on a pixel brings up that pixel's analysis window, this time containing its S-curve (average photoelectron count over threshold). If requested, the photoelectron spectrum (first derivative of an S-curve) and the derivative of the spectrum are displayed (Fig. 10). An additional function looks for position of the single photoelectron peak and valley between them and the pedestal.

If the calibration was performed sequentially with different preamplifier gains in the same run, the curves for all gains can be displayed and analysed simultaneously (Fig. 11). Such data sets are useful both for determining the correct initial gain for the pixel and for obtaining data necessary for the on-line calibration procedure. The latter can be done with a peak finding function, which in this case finds the peak and the valley position for all the gains and attempts to find a dependence of the peak position in DAC units on the gain value (Fig. 12) utilising ROOT Minuit based fitting facilities. The procedure can be performed for all the pixels of the focal surface, providing statistics of the fits' results and allowing for deriving a general dependence.

6.3. The functions summary and the software performance

To summarise, current features of the basic visualisation and analysis program include the following:

- Visualisation of the focal surface⁴ for event data.
- Visualisation of the focal surface for calibration data.
- Browsing through GTUs/thresholds for focal surface visualisation.
- Display of light curve – photoelectron count over time (GTU) – for the chosen pixel.
- Display of an S-curve.
- Display of the first derivative and second derivative of the S-curve.
- Single photoelectron peak and valley finding.
- Calculation of a light curve periodogram.
- Analysis of a pixel's single photoelectron peak behaviour with gain change.
- Display of the focal surface pixels statistics.
- Providing both the electronics and geometrical coordinates of a pixel.

⁴ Currently for a single PDM board containing 36 PMTs for EUSO-TA purposes, but this can be easily extended in the future.

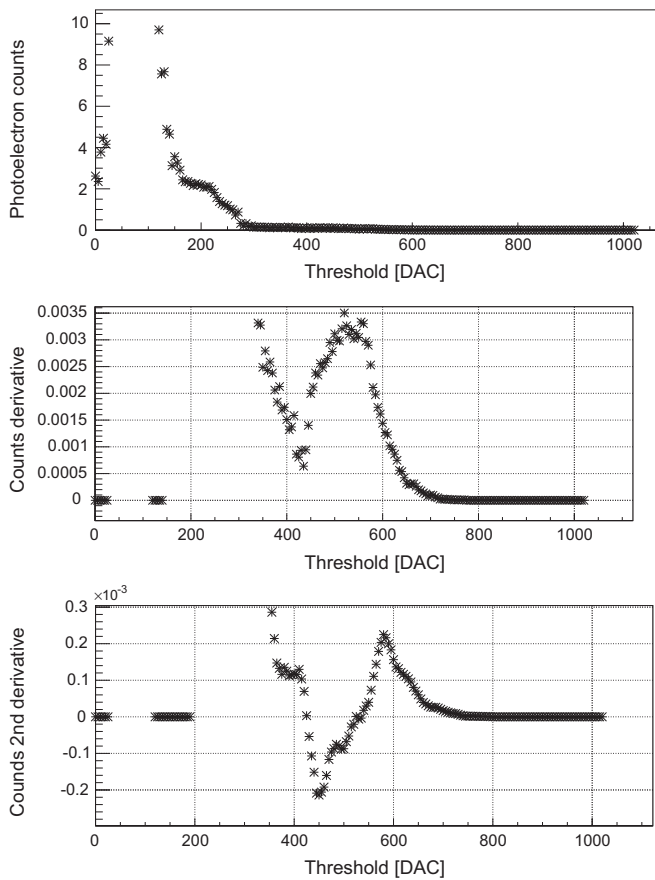


Fig. 10. Results of the analysis of the calibration for a single pixel (for a small statistics). The top plot shows an S-curve – the counts over threshold value plot, the middle is a “single photoelectron spectrum” – a derivative of an S-curve, and the bottom is the second derivative of an S-curve. The single photoelectron peak is clearly visible in the middle of the “single photoelectron spectrum”.

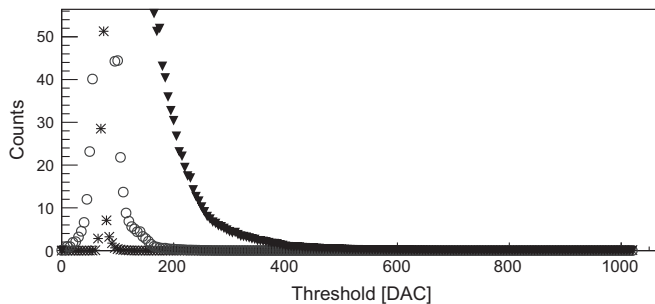


Fig. 11. Three S-curves (differing by the marker) obtained in a single run of calibration with different preamplifier gains.

The features currently under development include summing and averaging the photon count over specific time and different single photoelectron peak position estimation algorithms. The functionality will be extended as users' requests come in after testing these programs. In future we plan to include a visualisation of data and commands flow for system monitoring purposes.

The software has not yet been developed as a performance critical application, therefore no benchmarks have been calculated, however it performs very well on a modern day 64-bit Linux running an Intel Core i7-3840QM CPU @ 2.80 GHz. The speed is related mainly to the structure of the TTree, which in the current version does not include any indexes or has not been optimised in a dedicated way. Nevertheless, browsing through different GTUs/thresholds updates the focal surface display instantaneously. Drawing the graphs for specific

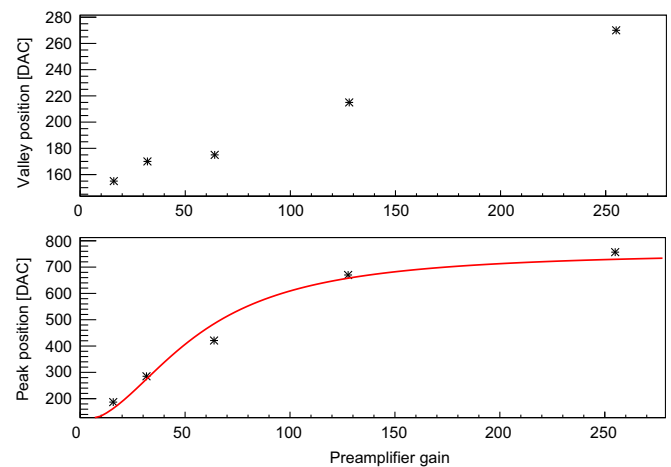


Fig. 12. The single photoelectron valley (top) and peak (bottom) positions vs. preamplifier gain. The curve is a fit of the dependency.

pixels carries a slight delay related to the large distance of data containing consecutive values for the same pixel, however calculating and drawing derivatives is instantaneous as the data already resides in memory. Therefore these small issues can be easily resolved with tuning of the TTree buffering.

The overhead of using Python is more visible when using analysis functions, such as fitting the data or calculating a periodogram. The latter, for a data set of $\sim 130,000$ points takes about 1.5 min, but performing a mathematical analysis on such a big data set is quite unlikely in our experiment and has been made mainly for test purposes. However, a dedicated analysis software where performance is critical is being written in C++.

7. Conclusions

We have shown the basic scheme of on-line and off-line data flow and analysis in the EUSO-TA experiment. The on-line software utilising a low-power CPU fulfils the established requirements for a low resource consumption. This has been achieved mainly by implementing a calibration algorithm based only on simple mathematical operations, not requiring numerical fitting, therefore significantly reducing the number of needed CPU operations. The accuracy is improved by employing low-noise derivatives, however the final effectiveness remains to be determined in future laboratory tests.

The internal, hardware based format of data uses minimal processing power, but is inconvenient for off-line data analysis purposes. We designed and implemented an off-line data format that fulfils crucial requirements of fast and easy random access to very large data sets. The data files can be distributed without any additions and analysed directly with tools available on most computers used in this branch of science.

The off-line data format has been tested with the EUSO-TA hardware in laboratory conditions. We successfully stored data up to several gigabytes in a single file. However, the quality of handling of larger data sets remains to be tested. The data storing speed is sufficient – on the used hardware conversion of the original data was faster than its acquisition speed, opening a possibility to develop a real-time monitoring system. The scalability of the software has been tested with simple artificially generated data. The test will be extended in future with Monte-Carlo packages, which are also needed to prove the transparency of the design to the origin of data. The format contains all the data generated by the EUSO-TA CPU, but lacks storing of the so-called housekeeping data, which are produced by a different subsystem. This remains to be implemented.

The access to off-line data has been tested on different machines operating on Fedora and Scientific Linux, and on MacOSX. The data was displayed without any noticeable delays both directly with ROOT interpreter and Python based quick-view. In all cases, the software required only working installation of ROOT with PyROOT bindings, NumPy and SciPy libraries, greatly enhancing its accessibility. Testing on Windows ROOT installations is pending.

The presented software fulfils all the crucial requirements, which has been proved by gathering, sharing and analysing data from experiment in laboratory conditions. It has already become necessary in hardware tuning and data gathering. The ease of use, performance better than in initial requirements and quickness of adding additional functions make it the first step towards an implementation in a space-based experiment, most likely without many modifications.

Acknowledgement

The authors would like to thank the JEM-EUSO collaboration for the support. This work is partially supported by JSPS KAKENHI Grant no. 24244042.

L.W.P. acknowledges the support from JSPS Postdoctoral Fellowship for Foreign Researchers.

References

- [1] T. Wulf, *Physikalische Zeitschrift Chemie* (1909).
- [2] W. Kolhrster, *Physikalische Zeitschrift* 14 (1913) 1153.
- [3] D. Pacini, *Il Nuovo Cimento Serie VI* 3 (1912) 93.
- [4] V. Hess, *Physikalische Zeitschrift* 13 (1912) 1084.
- [5] J. Abraham, et al., *Nuclear Instruments and Methods in Physics Research Section A* 523 (2004) 50.
- [6] H. Kawai, et al., *Nuclear Physics B* 175–176 (2008) 221.
- [7] J.H. Adams Jr., et al., *Astroparticle Physics* 44 (2013) 76.
- [8] H. Prieto, et al., in: Proceedings of the 33rd International Cosmic Ray Conference, 2013, [arxiv:1307.7071](https://arxiv.org/abs/1307.7071), p. 95.
- [9] S. Blin, P. Barrillon, C. de La Taille, *Journal of Instrumentation* 5 (2010) 12.
- [10] J.H. Adams Jr., et al., in: Proceedings of the 33rd International Cosmic Ray Conference, 2013, [arxiv:1307.7071](https://arxiv.org/abs/1307.7071), p. 91.
- [11] K. Shinozaki, et al., in: Proceedings of the 33rd International Cosmic Ray Conference, 2013, [arxiv:1307.7071](https://arxiv.org/abs/1307.7071), p. 31.
- [12] Y. Takizawa, et al., in: Proceedings of the 33rd International Cosmic Ray Conference, 2013, [arxiv:1307.7071](https://arxiv.org/abs/1307.7071), p. 107.
- [13] H. Miyamoto, et al., in: Proceedings of the 33rd International Cosmic Ray Conference Proceedings, 2013, [arxiv:1307.7071](https://arxiv.org/abs/1307.7071), p. 103.
- [14] J. Bayer, et al., in: Proceedings of the 33rd International Cosmic Ray Conference, 2013, [arxiv:1307.7071](https://arxiv.org/abs/1307.7071), p. 99.
- [15] F. Kajino, et al., in: Proceedings of the 33rd International Cosmic Ray Conference, 2013, [arxiv:1307.7071](https://arxiv.org/abs/1307.7071), p. 11.
- [16] P. Gorodetzky, et al., in: Proceedings of the 33rd International Cosmic Ray Conference, 2013, [arxiv:1307.7071](https://arxiv.org/abs/1307.7071), p. 71.
- [17] Richard Wesley Hamming, "Digital Filters", 1977.
- [18] CERN ROOT framework (<http://root.cern.ch>).
- [19] N.R. Lomb, *Astrophysical and Space Science* 39 (2013) 447.
- [20] E. Jones, et al., (<http://www.scipy.org>), 2001.