IRSTI 29.03.77 https://doi.org/10.26577/phst20251228

# $^3$He neutron detector with Android smartphone integration

A.S. Kussainov[1, 2*], P. Arnqvist[3], N.P. Arnqvist[3],
N.O. Saduyev[1, 2], O. Kalikulov[1, 2], N. Yerezhep[1, 2],
A. Baktoraz[1, 2] and S. Utey[1, 2]

[1]Department of Physics and Technology, Al-Farabi Kazakh National University, Almaty, Kazakhstan
[2]Institute of Nuclear Physics, Almaty, Kazakhstan
[3]Department of Mathematics and Mathematical Statistics, Umeå University, Umeå, Sweden
*e-mail: arman.kussainov@gmail.com

*(Received September 22, 2025; received in revised form November 14, 2025; accepted December 5, 2025)*

We have developed a homemade neutron flux detection module with $^3$He tube hot-swap capability and control-rich Android software interface. Real-time data analysis is done by a smartphone with Android application interfaced with the detector via a USB cable. This setup can be used as a neutron and gamma ray background detector or as a compact, mobile $^3$He tubes calibration tool making it a cheap and easy-to-use alternative for the stationary setups. A fast neutron detection algorithm was implemented as a set of Java scripts and tested for real-time signal analysis. The modular structure of the device allows easy deployment and customization with further software development and regular upgrades. The current prototype was tested at the Nuclear Physics Research Institute under different neutron flux intensity conditions from the VVR-K water-cooled research reactor. Its simplicity and significantly lower cost, compared with conventional detector equipment, make it valuable for easy repetitive tasks with medium requirements for precision and neutron flux intensities.

**Keywords**: $^3$He detector, proportional counter, android application, functional clustering, neutron capture, USB interface.
**PACS numbers:** 07.77.−n, 07.05.Hd, 02.70.−c.

## 1. Introduction

Personal Android devices provide adequate computational power, availability [1], and comfort of use over stationary computers, contributing to the increasing number of their applications in daily use and in fundamental physics experiments. Fast and reliable interaction between the measurement tool and processing module is important [2]. Android devices could use their own internal sensors, like accelerometers, Hall and infrared sensors, etc, or work with an external hardware coupled through different interfaces to detect fast transient phenomena of varying nature [3].

There are studies reporting the onboard hardware such as smartphone CMOS camera [4] used for particles detection. In more complex cases, to facilitate the efficient interaction of the Android devices together with registration and detection hardware, they are coupled with the separate detectors and/or electronic assemblies through the wireless **[5]** interface or USB cable. Sometimes, the measurements are done at some remote facility equipped with server capability providing the user with internet access with static or dynamically updated records and measurement data. The typical example for the last one is the global cosmic ray neutron monitor networks supported by various institutions worldwide, see, for example, the range of neutron monitor stations maintained by the University of Delaware Bartol Research Institute [6].

With the particular hardware and experimental problem in mind, the individual solution is usually implemented within this range of choices. Although some components and modules, such as registration devices, signal amplifiers, ADCs, USB or wireless interfaces, etc, remain the same, the custom software finalizes them into a single measurement tool to complete the desired experiment objective.

Neutron physics was and still remains the import part of modern tool of fundamental research as well as a practical tool in material studies [7]. Nuclear and

ionizing radiation detectors provide vital information about the environment and transient process around [8], We employed the modular neutron/gamma ray detection setup, provided by the Cosmic Rays Physics Laboratory [6] at the Nuclear Physics Institute, Almaty, Kazakhstan. In this setup, the detector is equipped with an electronic hardware interface consisting of an amplifier, an ADC, and a USB 2.0 bus for data exchange between the electronics and Android devices (smartphone). The smartphone is used for data storage and processing. The ADC generates digital output for every neutron and neutron-like, usually gamma ray, detection event in a proportional $^3$He-filled SNC-18 (Slow Neutron Counter) detector tube; see Figure 1. These SNC-18 tubes are widely used in different stationary and mobile setups and can be found in old and new conditions in stock. High detection efficiency and low background noise are the benefits of using these Helium-3 detectors. $^3$He interaction with neutron flux is the separate part of the contemporary nuclear physics research [9].

In our case, the ADC is designed to sample the signal output from the detector amplifier at the rate of 5000 times per second [10]. The detector module's electronics receive the high-voltage power from any connected Android device configured in the USB host mode. The USB host mode [11] had been conveniently added to Android devices version 3.1 and higher.

As a further development, we completed the provided setup with custom Android software for data flow registration, processing, and storage. The set of Java scripts is used for the USB bus signal HEX to decimal conversion, basic shape recognition routines and events counting as well as for the processed data formatting and subsequent storage. Additional work has been done for developing R scripts [12] for retrospective signal patterns classification and filtering.

The assembled device, with the $^3$He detector tube attached to the electronics housing, and the smartphone, has a relatively compact and portable dimension of less than 50 $cm$; see Figure 2.


**Figure 1** – SNC18 Slow neutron counter tube.

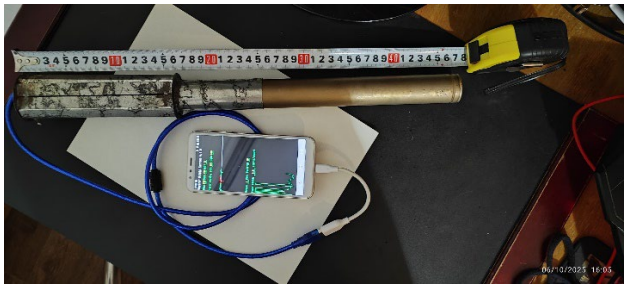


**Figure 2** – Assembled experimental setup running the data acquisition.

As a result of this project, a low-cost and simple neutron flux detection device is implemented. The open source software with algorithms for specific signal shapes' detection and recognition and the clean and efficient interface for user communication and measurement control are developed. Consistent measurement data were collected under different conditions. The future work for detection algorithms and device versatility improvements is outlined.

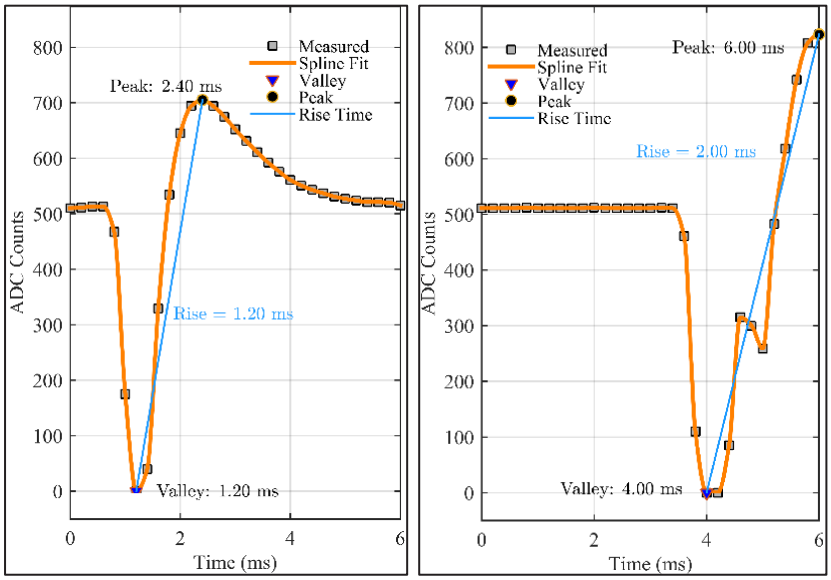
## 2. Materials and methods

*2.1 Signal shapes and events counting*

Figure 3 (a)-(b) shows the typical shapes that could be seen in the amplifier-generated signal, digitized by the ADC. It is sent further down the USB cable to a smartphone as the raw hexadecimal data sequence and ultimately received and converted by the software to the decimal representation. If not hidden by the noise and other unclassified events, the meaningful data event contains the well-noticeable voltage drop from about 500 a.u. value baseline, see Figure 3 (a). This drop is caused by the two heavy ions, triton and proton, from the reaction of neutron capture, see Eq. 1, producing the fast ionization of the detector gas

$$^3He + n \rightarrow {}^3H + p + 0.754MeV \quad (1)$$

Thus, the energy of the reaction goes to the kinetic energies of the proton and triton ions moving in the opposite direction and leaving a heavy trail of ion pairs. Their drift to the anode and cathode results in the following transient electric current in the detector's circuitry [13]. Unlike a single electron event produced by gamma-ray Compton scattering or photo-effect, the amplitude of a signal from a neutron capture is well above the noise level. The difference in amplitudes of gamma ray and neutron detection events is regularly used to discriminate between them in counting. Under a better resolution, for this shape, one could potentially see the two overlapping peaks from $^3$H and p ions [14]. We observe similar patterns, see Figure 3(b), but ours are primarily from the overlapping of two separate events, like the simultaneous detection of two neutrons or one neutron and one gamma quantum.




(a) (b)

**Figure 3** – (a) Typical signal shapes recorded from a neutron detection event in a $^3$He proportional counter. (b) Matlab generated plot for the data piece on the left with a 5-point moving average of 50 seconds.

The sampling frequency of our ADC is just not high enough to see the triton and proton events separately. Thus, depending on the event's location around the detector's physical volume and the detector's state after the previous event discharge, multiple events' shapes overlap and vary drastically in amplitudes, duration, and scaling factor.

Next, after the signal drop down almost to the zero level, the preamp shapes the electronics' response for the free charge surge dissipation, leading to the equally fast signal recovery with slight overshoot above the baseline. Subsequent return to the equilibrium is caused by the ion/electron drift and recombination processes being mediated by electronics and exhibited in the signal shape as an exponential decay.

The $X$ axis on the Figure 3 (a)-(b) is spanned by the value of the buffer size supported by the hardware and software drivers. Documented by the software driver's manual and correlated with the transmission rate and other parameters, the buffer size, set in the app by default, is 62 of the 8-byte words. The buffer is arranged as 31 couples of hexadecimal values. That is the ADC splits the three-digit voltage value into a pair of 8-byte words. Backwards concatenation of these two parts in a single 3-digit value is implemented in our app.

The ADC's 5000 $sec^{-1}$ fixed rate for the voltage sampling makes this buffer size equal to the 6.2 $msec$ period. Rather crude estimate for the count speed of the events of comparable amplitude, even without referring to the Raleigh criterion, gives us about 6 events per buffer size, or per 6.2 $msec$, that is about 1000 counts per second. This value is greatly reduced by the gas-filled detector tube recovery times and multiple other issues.

To provide the stand-alone detector setup with a data analysis module and implement the counting procedure, we have developed an Android application to process and record the data HEX sequence from the detector in real time. The free public drivers' library [15] was used to read the HEX data flow. Running on Android 3.1 version of the operating system and newer, it provides a communication interface with Arduinos and other USB serial hardware on Android. An extended list of the features provided for supported FTDI chip (our setup is using the one) devices includes open, close, setParameters, read and write, as well as many other flow control features.

Thus, the three parts of the waveform of interest are given by a) a sudden drop in a signal level below the baseline, followed by b) its rather speedy recovery above the average baseline level, usually half as high as the signal drop, completed by c) exponential-like decay.

We chose the event detection criteria based on the most prominent sudden drop below the noise level, corresponding to the transient current in the detector scheme. As soon as the data comes from a single 6.2 $msec$ buffer, the app searches through the voltage values sequence and finds the first available local minimum in the data. Further on it tests the presence of the fast recovery while staying well below the nose level. If all selected conditions and flags are satisfied, the event is classified as a neutron hit. As has been told, the neutron and gamma-ray pulse shapes have rather similar forms, though a definite distinction could be made depending on detector type, energy and other major parameters of the particle, see for example [16].

*2.2. Graphical user interface*

The app's user interface, as seen in Figure 4, provides the following sequence of interactive windows, saving and storage options, and text output.

The first screenshot, see Figure 4 (a), is activated as soon as the supported device with an FTDI chip is plugged in. This functionality is provided by the original driver's package. The minor alterations to the interface were made to suit our needs and to remove redundant, unsupported communications buttons and graphical interfaces.

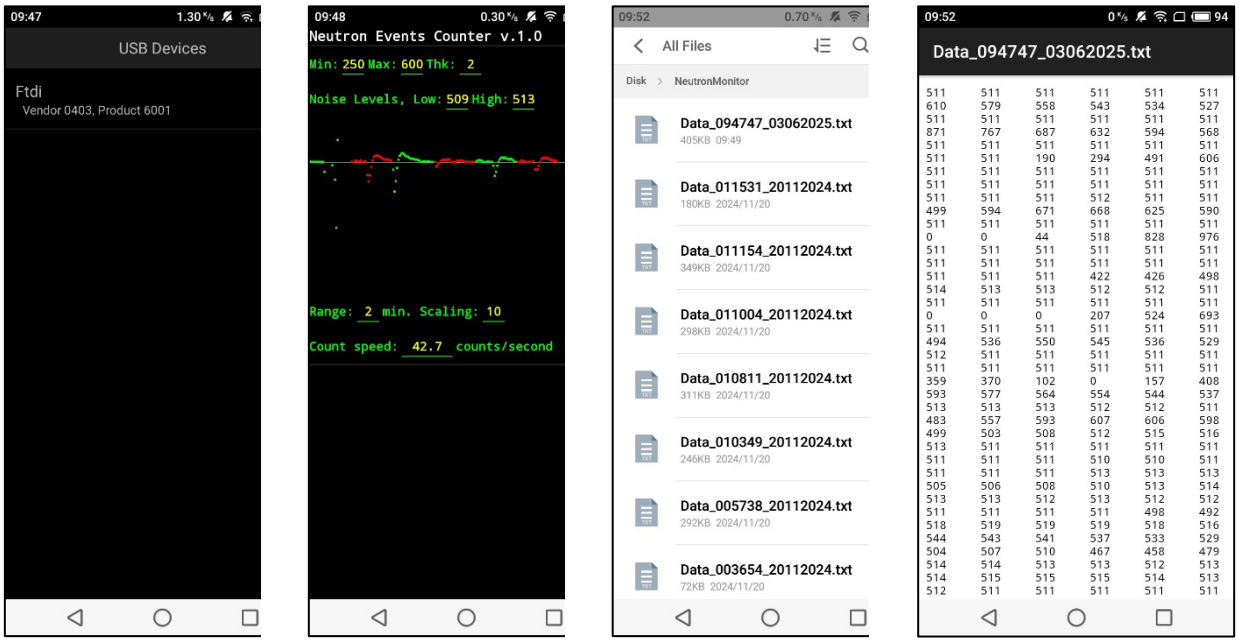
As soon as the user presses the button with the detected communication chip, the main screen with measurement, display tools and buttons becomes available, see Figure 4(b). Data acquisition starts immediately as soon as the proper voltage levels applied to the gas tube etc.

Figure 4(b) shows the two main parts of the graphical interface. The top subwindow displays the captured signal shapes colored in green and red intermittently and plotted one by one in the order of detection. This was done to show that these are the different captured buffers selected from a data stream with the neutron detection event and they do not necessarily follow each other. The buffers with data pieces containing noise and signal, which do not meet the selection criteria, are discarded and are not displayed.

The bottom part of the graphical data display is the current count speed. The size of the time window is default 2 minutes but could be changed anytime by a user in interface. All the data fields above are interactive and, if changed, they affect the scale of the signal, the shapes' visibility, and the detection sensitivity of the algorithm. Besides, any part of the dynamically updated graphical interface could be turned off to help to free CPU and GPU load.

The next screenshot, see Figure 4 (c)-(d), shows the system's data storage default path and file's name containing unique time series identifier and timestamp. The last one, see Figure 4 (d), is the selected buffers data saved in decimal form as a text file. Alternatively, as seen in Figure 5(a)-(b), one can opt for the count speed vs. time output. This type of the data is vital in pulse shape and time series analysis [17]. At the present stage of development, it is hardcoded in order to avoid overcrowded user graphical interface.

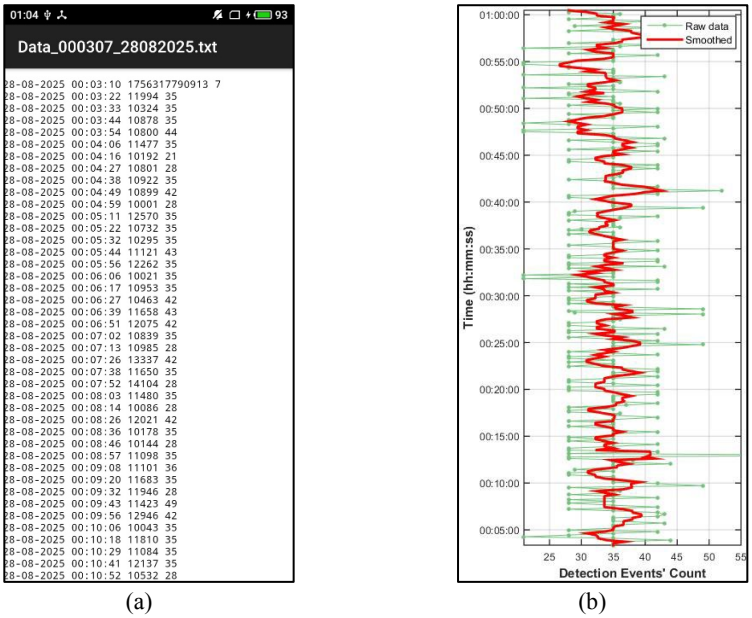
The conversion of hex to decimal data is built around internal Java functions. As mentioned, by original design, the three-digit number corresponding to the signal level is split by ADC into two parts and sent to the Android device, one part after another, in a single buffer of 32 numbers. Upon receiving, the two parts of a single data point are concatenated together, see *HexDump* part of the package, and sent for a further processing, that is the pattern search, see the event detection criteria discussed above.




(a) (b) (c) (d)

**Figure 4** – (a)-First window after the FTDI chip was detected and program is waiting to proceed to data acquisition; (b)-Main active acquisition window, with preselected by algorithm signal shapes, counting speed and options for different acquisition parameters; (c) – automatically generated file names with the time stamp; (d) - the content of the data file.

As discussed previously, the data processing is built around a single or several features selected as the neutron detection event. The whole buffer is scanned in a loop, and multiple events could be detected in a single buffer. A buffer with at least one event is displayed on the screen for control purposes. The micro USB port throughput of 480 Mbps for USB 2.0 is well above the required speed of the data flow. For the in depth analysis of the buffers with multiple detection events as well as for the overall data sequence statistics, we devised the functional clustering of recorded signal buffers.




(a) (b)

**Figure 5** – (a) Optional, data format for count speed vs time. (b) Matlab generated plot for the data piece on the left with 5-point moving average of 50 seconds.

*2.3. Functional clustering approach*

The clustering of the neutron detection signals is performed using a model-based functional clustering approach, as described in [18]. This method incorporates functional data analysis techniques to classify signal waveforms into homogeneous groups, ensuring an accurate distinction between neutron-induced events and background noise.

*2.3.1. Model formulation*

Let $N$ be the number of detected signal events, where each event $i$ is represented by a function $g_i(t)$ observed at discrete time points $t_{i1}, \dots, t_{in_i}$. The clustering model assumes that the data originate from a mixture of $G$ underlying groups, each characterized by a distinct functional pattern and potentially different covariance structures.

The observed signal values $y_{ij}$ at time $t_{ij}$ are modeled as:

$$y_{ij} = g_i(t_{ij}) + \epsilon_{ij}, i = 1, \dots, N, j = 1, \dots, n_i, \quad (2)$$

where $\epsilon_{ij}$ represents the normally distributed measurement noise: $\epsilon_{ij} \sim N(0, \sigma^2)$.

Each function $g_i(t)$ is expressed using a basis expansion:

$$g_i(t) = \phi_i^T(t)\eta_i, \quad (3)$$

where $\phi_i(t) = \left(\phi_{i1}(t), \dots, \phi_{ip}(t)\right)^T$ denotes a $p-$ longitudinal vector of the values of the known basis functions of B lines, and $\eta_i$ is a $p-$vector of unknown random coefficients. These coefficients are assumed to follow a Gaussian mixture model:

$$\eta_i = \mu_{z_i} + \gamma_i, \eta_i \vee z_i = k \sim N_p(\mu_k, \Gamma_k), \quad (4)$$

where $z_i$ is the latent cluster membership variable with prior probabilities $P(z_i = k) = \pi_k$ for $k = 1, \dots, G$, $\gamma_i$ represents signal-specific within-cluster variability. The variability within each cluster is modelled through the covariance matrix $\Gamma_k$, allowing for differences in structure across clusters. We use the R package fdaMocca [19, 20] to perform this model-based clustering method.

*2.3.2. Inference and estimation*

The model parameters $\theta = \{\pi_k, \mu_k, \Gamma_k, \sigma^2\}$ are estimated via the Expectation-Maximization (EM) algorithm, which iteratively updates cluster assignments and maximizes the observed likelihood. Given a fitted model, each event is assigned to the cluster that maximizes the posterior probability:

$$P(z_i = k \vee y_i, \theta) \propto f_k(y_i, \theta)\pi_k, \quad (5)$$

where $f_k(y_i, \theta)$ denotes the normal density given membership in cluster $k$.

The optimal number of clusters is chosen using criteria such as the Bayesian Information Criterion (BIC) or the Akaike Information Criterion (AIC). The performance of the clustering model is assessed using entropy-based measures and cross-validation.

## 3. Results and discussion

The software was initially built for Android version 9.0 and lower (Lollipop), with the possibility of upgrading to a higher version following the recent changes in the security policies for the Android software. It performs well with moderate count speed values. Nevertheless, as expected, the events tend to overlap as soon as the neutron flux increases; see Figure 3(b). This puts a rough estimate of the count speed as 1000 events per second, based on the length of the transmitted buffer. Overlapping, caused by the digitizing hardware specification is aggravated by an extended hardware geometry that could register the multiple events simultaneously. In this case, the other counting algorithm is under development designated to analyze the overlapping signal shapes and increase the potential counting speed of the device. Additionally, the clustering analysis is offered to classify these events retrospectively or on the fly.

It was found that, in long-run experiments, it is practical not to rely on the internal battery capacity of the Android device but on an extra power supply connected through a USB split cable connected to the main power supply.

In the regime where only counting speed is of interest, the counted event data is dampened for every 10 seconds or faster, which is not a problem for internal memory or for micro-SD card read/write speed.

In the case of older devices with limited memory and graphics processing power, the graphical output for the registered signal shape could also be disabled. In addition to counting performance, a preliminary clustering analysis was performed to explore the structure of the recorded signal shapes.

Nuclear physics experiment also relies on the historic data postprocessing, if the experiment is irreproducible, one-of-a-lifetime or for some other reasons, when some efforts are made to clean the data, classify the observed patterns or correct the data acquisition artefacts [21,22].

The cluster analysis of the recorded signals, see Figure 6, resulted in the identification of the eight groups that reflect different types of the signal patterns. The number 8 of the clusters, as an input parameter, has been selected as a preliminary guess to observe the set of the features that could be found in the signal. At this stage these features include but not limited to the combinations of the full signals of different amplitudes or their parts, rear or head, for the classical neutron or gamma ray hit detection events plus the noise. Shape and amplitude analysis are available in the method making it a good tool to differentiate between gamma and neutron events.

Although this is a preliminary analysis, a general grouping can be observed. Some clusters correspond to typical neutron-like signals with a sharp drop followed by recovery, others capture low-amplitude events more consistent with background noise or gamma interactions, and a few clusters represent overlapping or mixed signals where two events occur close in time. The future work will target the overlapped signals decomposition as well.

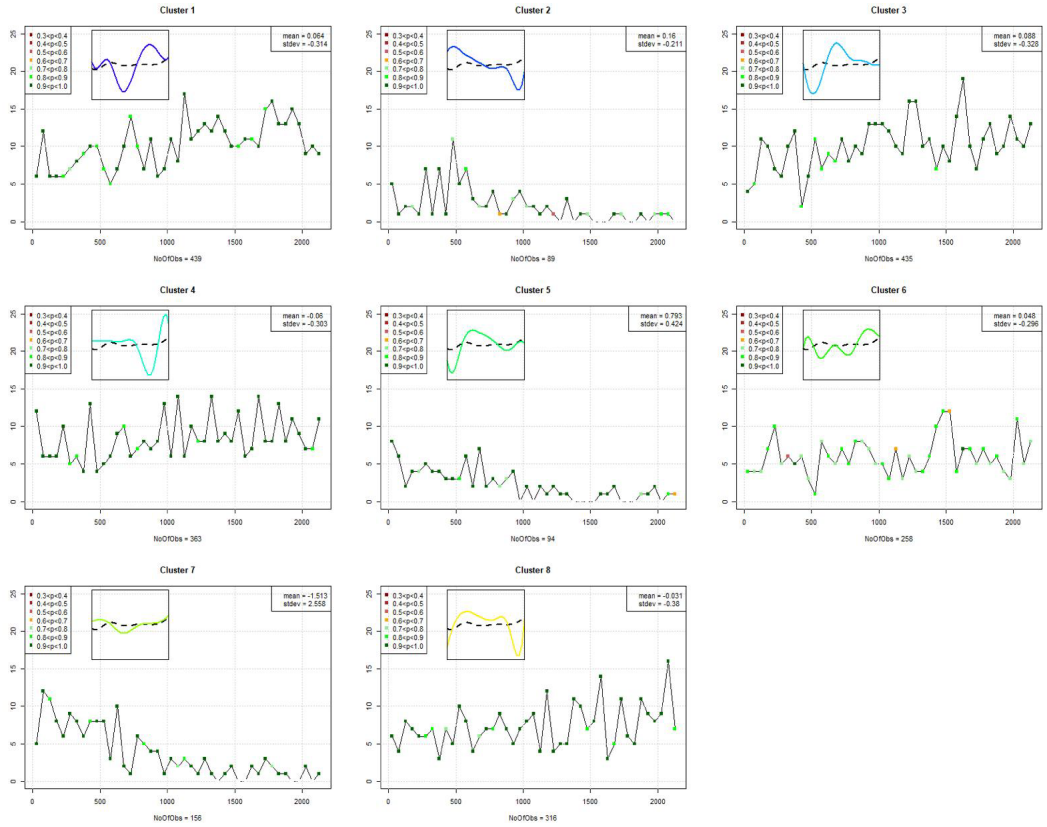



**Figure 6** – Estimated dynamics of the eight clusters of standardized signals. The posterior probabilities illustrate the uncertainty in cluster assignment and frequency over time. The colored curves in the sub-window plots represent the cluster-specific mean signals, while the dashed curve denotes the overall mean signal. The upper-right box provides the average of the mean and standard deviation of the signals within each cluster. The p-values are posterior probabilities showing the likelihood that a particular observation belongs to a cluster. Circumstantial visual interpretation of the clusters is the following, from left to the right, from top to bottom: 1. Cluster comprises signals with a sharp initial drop followed by a stable recovery, corresponding to well-formed neutron-like events; 2. Cluster includes moderate-amplitude signals with periodic fluctuations near the baseline, typical of background oscillations or low-energy interactions; 3. Cluster contains short-duration pulses with a distinct early peak and rapid decay, representing fast transient detections; 4. Cluster displays nearly constant low-amplitude traces with minimal variance, interpreted as background noise or incomplete events; 5. Cluster groups high-intensity pulses characterized by steep leading edges and pronounced recovery tails, consistent with strong single neutron captures; 6. Cluster shows overlapping waveforms of mixed amplitude, indicating partially superposed or coincident events within the buffer; 7. Cluster gathers a small number of very high-variance signals with large positive excursions, reflecting saturated or composite detections from simultaneous neutron and gamma interactions; 8. Cluster consists of weak, symmetric pulses oscillating around the baseline, most likely representing residual electronic noise or minor after-pulses of the detector circuit.

Although the exact assignment of clusters to physical processes requires further validation, preliminary results indicate that the method can successfully separate signal shapes into meaningful groups. This provides an important step toward distinguishing neutron-induced events from noise and composite signals, improving the interpretability of the data, and laying the groundwork for a more refined classification in future studies.

Estimated dynamics of the five clusters of standardized signals. The posterior probabilities illustrate the uncertainty in cluster assignment and frequency over time. The colored curves in the sub-window plots show the cluster mean signals, and the dashed curve is the overall mean signal. The upper right box gives the average of the mean and standard deviation of the signals in the cluster.

## 4. Conclusions

We combined the low-cost and universally available components such as $^3$He detector tubes, ADC boards, and USB interfaces with Android drivers and our software, for data transfer and handling, to produce a simple and reliable measurement tool with a flexible interface and objectives.

The neutron flux, transformed by a detector tube into an electrical signal, is digitized and sent through the data cable to a smartphone. The raw data, after being received from a signal buffer, is searched simultaneously for the characteristic shapes of the neutron detection event and treated further for hits counting or signal amplitude measurements to produce the power spectrum (yet to be implemented).

Additionally, by employing the robust estimation approach, the model effectively differentiates neutron signal patterns from background noise and gamma ray events, ensuring accurate classification and enhanced detection performance. At present, this advanced retrospective analysis is done, using the R language, separately on a PC. Nevertheless, it could be ported to an Android device for on-the-fly or retrospective data filtering and signal analysis in future development.

**Acknowledgments.** This research has been funded by the Science Committee of the Ministry of Education and Science of the Republic of Kazakhstan (Grant No. BR21881930).

**References**

1. Ishikawa N., Nadeem, Nomura H., Yoda Y., Uetsuki O., Fukunaga K., Nagoya S., Sawara J., Ishihata H., Senoguchi J. High-performance distributed computing with smartphones // In: Zeinalipour D. et al. (eds.) Euro-Par 2023 International Workshops, Limassol, Cyprus, August 28–September 1, 2023, Revised Selected Papers, Part II. – Cham: Springer, 2024. – (Lecture Notes in Computer Science; Vol. 14352). – Pp. 229–232. https://doi.org/10.1007/978-3-031-48803-0

2. Temirzhanov A., Sadykov B., Zholdybayev T., Duisebayev B., Ussabayeva G., Kerimkulov Z. STM32F407 microcontroller based multichannel analyzer for spectroscopy // Physical Sciences and Technology. – 2024. – Vol. 11, No. 3–4. – Pp. 21–28. https://doi.org/10.26577/phst2024v11i2b03

3. Budden B. S., Stonehill L. C., Dallmann N., Baginski M. J., Best D. J., Smith M. B., Graham S. A., Dathy C., Frank J. M., McClish M. A. A $Cs_2LiYCl_6$:Ce-based advanced radiation monitoring device // Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment. – 2015. – Vol. 784. – Pp. 97–104. https://doi.org/10.1016/j.nima.2014.11.051

4. Bourbeau J., Campos F., Gallay B., Meehan M., Morgan R., Peacock J., Pizzuto A., Schneider C., Simons A. L., Vandenbroucke J., Winter M. Particle identification in smartphone camera images using the Distributed Electronic Cosmic-ray Observatory // In: Proceedings of the 36th International Cosmic Ray Conference (ICRC 2019), Madison, USA, July 24–August 1, 2019. – PoS(ICRC2019)390. – Published online: August 17, 2025. – https://pos.sissa.it/PoS(ICRC2019)390

5. Bee Research Pty Ltd. $^3$He Neutron Detector GS-Neutron-3: High-Efficiency $^3$He Neutron Tube with Built-in Bluetooth [Electronic resource]. – Sydney, Australia: Gammaspectacular. – Available at: https://www.gammaspectacular.com/blue/gamma-spectroscopy/neutron-detectors/gs-neutron-3 (Available on December 16, 2025).

6. University of Delaware, Bartol Research Institute. BARTOL Neutron Monitors Program. – Available at: https://neutronm.bartol.udel.edu/Welcome.html (Available on December 16, 2025).

7. Muhametuly B., Duysebayev A. CCD camera in a neutron imaging system for real time and tomography investigations // Physical Sciences and Technology. – 2016. – Vol. 2, No. 2. – Pp. 18–27. https://doi.org/10.26577/2409-6121-2015-2-2-18-27

8. Dyachkov V. V., Zaripova Y. A., Yushkov A. V., Shakirov A. L., Bigeldiyeva M. T., Abramov K. E. Periodic variations in time of atmospheric alpha and beta radioactive nanoparticles // Physical Sciences and Technology. – 2019. – Vol. 6, No. 1–2. – Pp. 11–17. https://doi.org/10.26577/phst-2019-1-p6

9. Kunakov S. K., Shapiyeva A. Fission fragments and primary electrons' energy distribution in helium-3 plasma irradiated by neutron flux // Physical Sciences and Technology. – 2019. – Vol. 5, No. 3–4. – Pp. 78–82. https://doi.org/10.26577/phst-2018-2-160

10. Saduev N. Study of the Processes of interaction of neutrons and charged particles with beryllium nuclei [Master's thesis, electronic resource]. – Al-Farabi Kazakh National University, 2014. – 116 p.

11. Android Developers. USB Host Overview [Electronic resource]. – 23 May 2025. – Available at: https://developer.android.com/develop/connectivity/usb/host (accessed 16.12.2025).

12. R Core Team. R: A Language and Environment for Statistical Computing [Electronic resource]. – Vienna, Austria: R Foundation for Statistical Computing, 2025. – Available at: https://www.R-project.org/ (accessed 16.12.2025).

13. Shalev S., Hopstone P. Empirical expressions for gas multiplication // Nuclear Instruments and Methods. – 1978. – Vol. 155, No. 1–2. – P. 237–247. https://doi.org/10.1016/0029-554X(78)90209-4

14. Vasilyev I. A., Djilkibaev R. M., Hlustin D. V. Investigation of the pulse shape in a ³He counter under neutron irradiation // Instruments and Experimental Techniques. – 2020. – Vol. 63, No. 2. – P. 146–153. https://doi.org/10.1134/S0020441220010236

15. Wakerly M., Morich K. Usb-serial-for-android: Android USB Host Serial Driver Library for CDC, FTDI, Arduino and Other Devices [Software, electronic resource]. – GitHub, Version 3.9.0, 10 March 2025. – Available at: https://github.com/kai-morich/usb-serial-for-android (Available on December 16, 2025).

16. Lee D. W., Stonehill L. C., Klimenko A., Terry J. R., Tornga S. R. Pulse-shape analysis of $Cs_2LiYCl_6$:Ce scintillator for neutron and gamma-ray discrimination // Nuclear instruments and methods in physics research section A. – 2012. – Vol. 664, No. 1. – Pp. 1–5. https://doi.org/10.1016/j.nima.2011.10.013

17. Yamaki S., Morimoto K., Kaji D., Wakabayashi Y., Takeyama M., Tanaka K., Tanaka T., Baba H., Yamaguchi H., Suzuki T., Morita K. R&D status of pulse shape analysis for short-lived decay of superheavy elements in GARIS-II // Physical Sciences and Technology. – 2017. – Vol. 3, No. 1. – Pp. 12–16. https://doi.org/10.26577/phst-2016-1-88

18. Arnqvist P., Sjöstedt de Luna S. Model Based Functional Clustering of Varved Lake Sediments [Preprint, electronic resource]. – arXiv:1904.10265, 2019. – Available at: https://arxiv.org/abs/1904.10265 (Available on December 16, 2025)

19. Arnqvist P., Sjöstedt de Luna S., Pya N. fdaMocca: an R package for model-based clustering for functional data with covariates // In: Skiadas C. H., Skiadas C. (eds.) Quantitative methods and data Analysis in applied demography. Vol. 2: Data, Models, Risk and Surveys. – Cham: Springer, 2025. – (The Springer Series on Demographic Methods and Population Analysis; Vol. 58). – Pp. 95–108. https://doi.org/10.1007/978-3-031-82279-7

20. Pya N., Arnqvist P., Sjöstedt de Luna S. fdaMocca: Model-Based Clustering for Functional Data with Covariates [Software, electronic resource]. – R package version 0.1-2. – CRAN, 31 March 2025. – Available at: https://CRAN.R-project.org/package=fdaMocca (Available on December 16, 2025).

21. Kussainov A. S., Saduev N. O., Em M. A., Mukhatay M. A., Myrzabek Y. T. Simple, contrast-based compartmentalization in 3D reconstruction from CT images // Physical Sciences and Technology. – 2020. – Vol. 7, No. 1–2. – Pp. 38–42. https://doi.org/10.26577/phst.2020.v7.i1.06

22. Aghabozorgi S., Seyed Shirkhorshidi A., Wah T. Y. Time-series clustering – a decade review // Information Systems. – 2015. – Vol. 53. – Pp. 16–38. https://doi.org/10.1016/j.is.2015.04.007

***Information about authors:***

*Arman S. Kussainov (corresponding author) – Senior Lecturer at the Faculty of Physics and Technology, Al-Farabi Kazakh National University (Almaty, Kazakhstan, e-mail: arman.kussainov@gmail.com).*

*Per Arnqvist – Associate Professor at the Department of Mathematics and Mathematical Statistics, Umeå University (Umeå, Sweden, e-mail: per.arnqvist@umu.se).*

*Natalya Pya Arnqvist – Associate Professor, Department of Mathematics and Mathematical Statistics, Umeå University (Umeå, Sweden, e-mail: natalya.pya@umu.se).*

*Nurzhan Saduyev – Deputy Director General for Science of the Institute of Nuclear Physics (Almaty, Kazakhstan, e-mail: n.sadyev@inp.kz).*

*Orazaly Kalikulov – Head of Cosmic Ray Physics Laboratory, Institute of Nuclear Physics (Almaty, Kazakhstan, e-mail: o.kalikulov@inp.kz).*

*Nurzhan Yerezhep – Lecturer at the Physics and Technology Department, Al-Farabi Kazakh National University, Researcher at the Cosmic Ray Physics Laboratory, Institute of Nuclear Physics (Almaty, Kazakhstan, e-mail: n.yerezhep@inp.kz).*

*Shynbolat Utey – Lecturer at the Faculty of Physics and Technology, Al-Farabi Kazakh National University, Researcher at the Cosmic Ray Physics Laboratory, Institute of Nuclear Physics (Almaty, Kazakhstan, e-mail: sh.utey@inp.kz).*

*Aliya Baktoraz – Lecturer at the Faculty of Physics and Technology, Al-Farabi Kazakh National University (Almaty, Kazakhstan, e-mail: baktoraz.aliya@kaznu.kz).*