Python-based Software Design for Gamma Energy Spectrum Analysis

Xiong Zeng, Rui Shi, Zhou Wang, Minjun Liu, and Bo Wang

School of Computer Science and Engineering, Sichuan University of Science & Engineering, Yibin, Sichuan 644005, China

Abstract

The design of a system for γ spectroscopy using the Python programming language is introduced. The system uses a LaBr₃ detector, which has a better energy resolution and detection effect than the NaI detector, which is the most commonly used detector in traditional applications, and has a good prospect for development. The software system creates a database table through MySQL to store the information of nuclide, energy scale parameters, count rate, etc.; and utilizes the USB communication interface program and the digital multi-channel spectrometer to realize the data transmission. The software system collects the energy spectrum data from the spectrometer by using the communication interface program, and implements a series of operations on the data, such as smoothing, peak searching, peak area calculation, count rate calculation, spectral line display, etc., and the results give the measurement results of count rate of special nuclides, change point detection, and so on. Aiming at the LaBr3 detector and digital multi-channel spectrometer supporting software needs process control, real-time measurement, data acquisition, γ energy spectrum analysis and other functions, based on the PyQt5 framework design and development of a set of γ energy spectrum analysis software system. The system is based on serial communication to realize the data transmission between the equipment and the system, and realizes the functions of data acquisition, real-time measurement, visual measurement display, γ energy spectrum drawing, γ energy spectrum analysis, system setting, process control, nuclide identification, etc. After experimental testing, the software system has a simple interface, stable operation, friendly human-computer interaction, and it has good practicability and maintainability, and it has a certain value of use.

Keywords

Python; γ Energy Spectrum Analysis; Serial Communication; Change Point Detection.

1. Introduction

Currently, most of the γ energy spectrum analysis software is implemented through the design of programs such as C, C++, C#, etc. C language has a small amount of code, fast execution speed, and powerful functions, but it has a high risk, a long design cycle, a low degree of portability, and in the encapsulation of data, which leads to a large problem with the security of the data in the C language. object-oriented programming carried out in C++, the use of the language is very flexible, and the function is very powerful. C# has a powerful memory backup function and avoids memory leaks, equipped with rich libraries, which can speed up the development, but it is difficult to deal with multi-threaded applications and is not very flexible[1]. As for Python, as a simple, free, compatible, object-oriented programming language with rich libraries, standardized code, extensible and embeddable, Python has very rich third-party libraries on which to develop, thus greatly improving the development efficiency; Python's programming ideas are well suited to the logic of people's thinking

without the need to master the programming ideas of computers as C# does Python is designed to fit the logic of people's thinking without the need to master computer program ideas like C. It works seamlessly on a variety of platforms, including Linux, Windows, FreeBSD, Solaris, and so on.

Therefore, we can develop software using an environment integrated with PyCharm for the Python language. Using a large number of practical third-party libraries in Python, including: PyQt5 library for graphical interface development, and PySerial library for handling serial port operations, in order to realize the high-performance plotting capabilities, especially for real-time data plotting PyQtGraph library, to build the software, complete the communication interface operations, plotting spectral lines, and so on. This makes the energy spectrum analysis software design will be faster and simpler, the development cycle is shorter, and the human-machine interface is more friendly.

2. Introduction to the γ Energy Spectrum Measurement System

Through the $LaBr_3$ probe for γ-ray detection, detection of γ-rays will be converted into a pulse signal, the use of digital multi-channel energy spectrometer for nuclear signal processing and analysis is converted into a discrete digital signal, FPGA for the extraction of spectral data to form an energy spectrum, and finally through the USB will be the transmission of spectral data to the spectral analysis software system, so as to achieve the spectrum of the energy spectrum smoothing, peak-seeking, energy scales, and other series of data processing, and the number of counting rate[2].
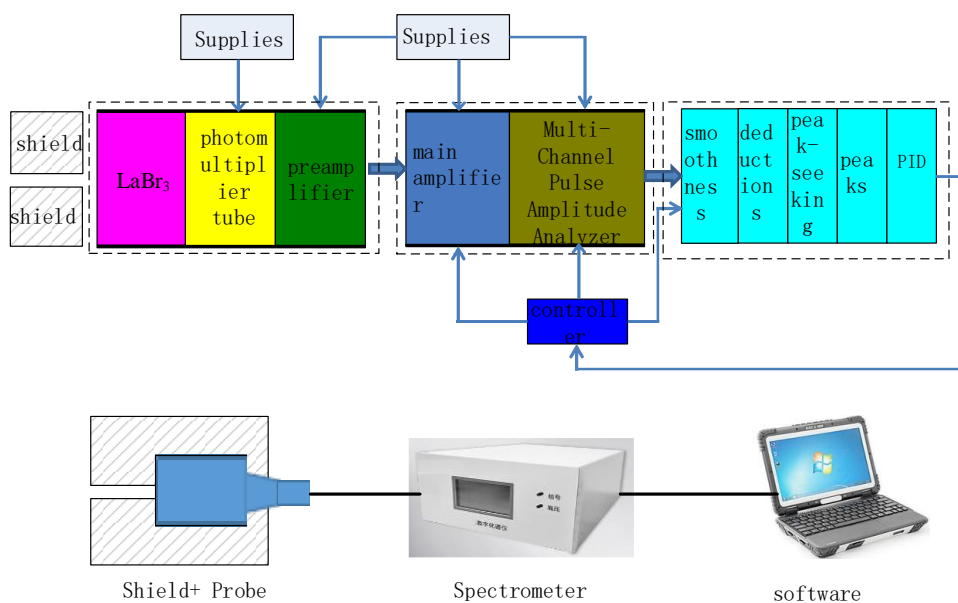

Fig. 1 Schematic diagram of the γ energy spectrum measurement system

3. Energy Spectrum Analysis Software Architecture Design

The general architecture of the software is shown in Fig. 2. It mainly involves the acquisition of γ energy spectrum data, energy spectrum analysis, energy spectrum display and other functions[3-6].

Analyzed from the internal processing of the software, the software should have the following main functions: serial port communication (read USB serial data), parameter call, spectral line processing, spectral line display, energy spectrum analysis, save the spectral data, database storage (nuclide information, count rate information, port information), and provide help and so on.

From the analysis of the software operating environment, it should be equipped with information storage data files such as: PDF help files, databases, .xls data files, etc., in addition to USB communication interface that can be realized with the digital multichannel spectrometer for data transmission.
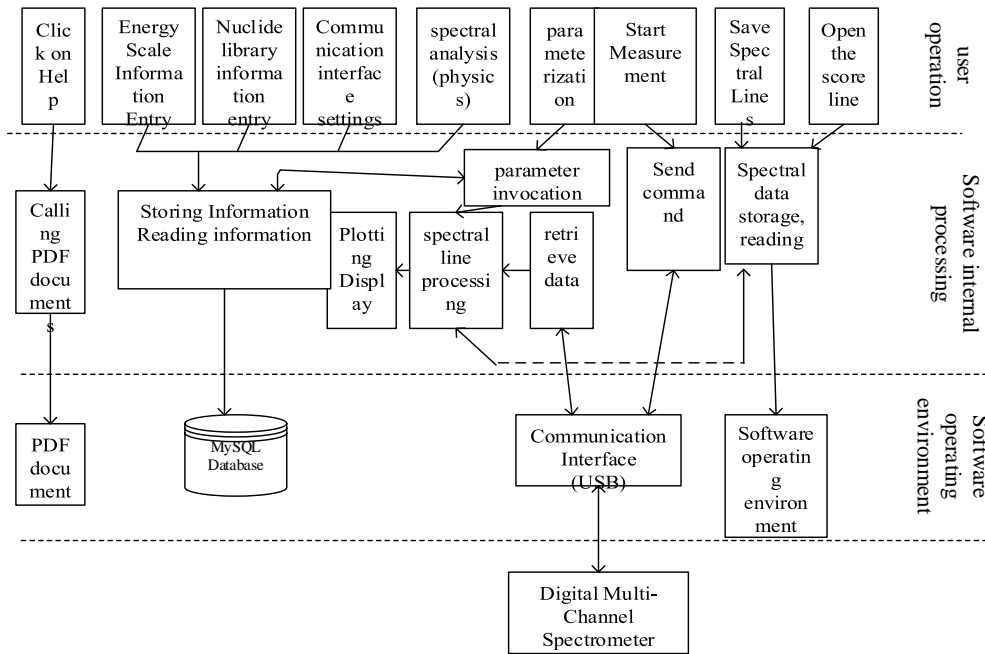
Fig. 2 Software Architecture Diagram

4. Software Design and Implementation

(1) Interface design

The software is mainly divided into five interfaces, two main measurement interfaces: count rate display and energy spectrum display, and three sub-interfaces: energy scale interface, serial communication parameter setting interface and nuclide library information interface, as shown in Fig. 3.
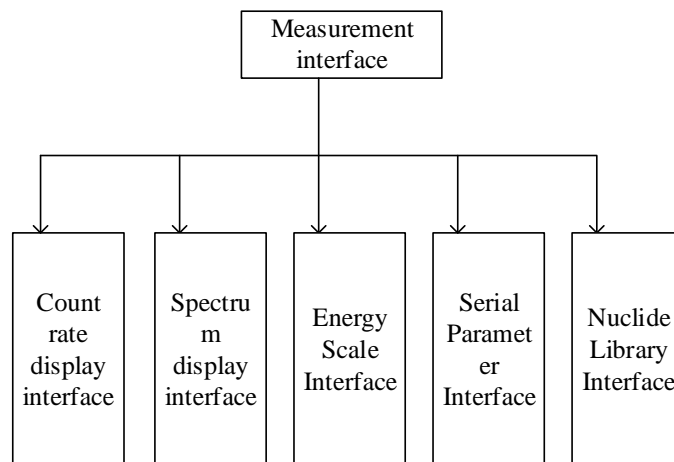

Fig. 3 Schematic diagram of the interface

Count rate display interface: includes function area, information display, serial port control and drawing area. There are five buttons in the function area: Start, Clear, Stop, Save Background and Deduct Background. The information display includes the count rate of individual nuclides and the total count rate. The serial port control includes selection of the open serial port, serial port number and baud rate. The top half of the plot area displays a scatter plot of count rates for individual nuclides and the bottom half displays a scatter plot of total count rates.
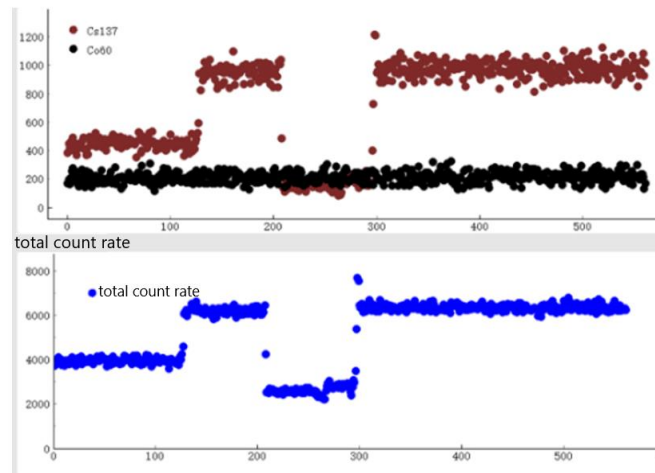
Fig. 4 Display of count rate

Energy spectrum display interface: includes time information area and energy spectrum information area. The time information area displays start time, measurement time, live time and dead time. The spectrum information area displays counts, channel addresses, and energy. The following status also synchronizes the display of time information and spectrum information, which is convenient to check the information when switching between the count rate display interface and the spectrum display interface.
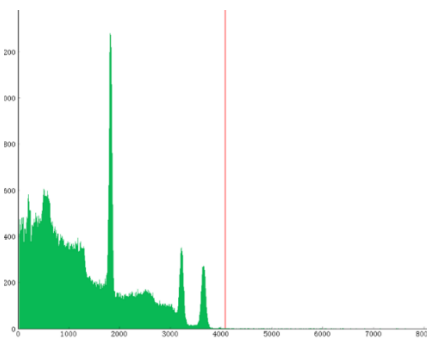

Fig. 5 Real-time energy spectrum

Energy scale interface and nuclide library interface: nuclide energy scale and data view are available. Serial parameter interface: set parameters for gain, bias, high voltage and trigger.
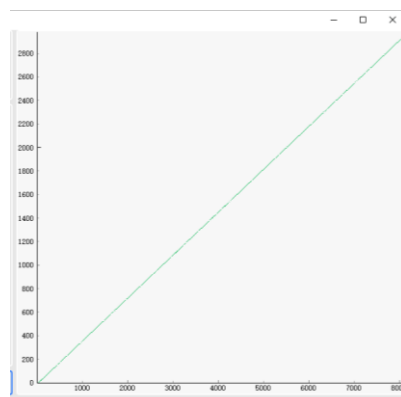

Fig. 6 Energy scale diagram

(2) Serial communications

The PySerial module encapsulates access to serial ports. It provides backends for Python running on Windows, OSX, Linux, BSD (which can be any POSIX-compatible system) and IronPython. The module named "serial" automatically selects the appropriate backend. serial.Serial('port number', 'baud rate', 'timeout setting', 'baud rate', 'timeout setting'). 'Timeout setting'), opens the serial port and gets the serial object in action. After establishing the connection, send to the digital multi-channel energy spectrometer to get the energy spectrum data, get the total bytes 26663, a total of 2051 small packets of data, remove the first 2 packets and the last 1 packet, the middle 2048 small packets for the 4096 channels of the energy spectrum data.

(3) Software processes

When the software is running, you need to click the button to open the serial port, after the serial port is opened, complete the initialization work such as parameter calling and serial port setting, click the start button to send the command to acquire the spectrum to the digital multi-channel spectrometer, and then wait for the data to be received. After the data is received successfully, make sure whether its format and the verification data format "AA 55 08 IDH IDL D0 D1 D2 D3 D4 D5 D6 D7" are the same, if they are the same, then the data packet is confirmed to be valid, otherwise it is discarded.

A histogram is drawn for the processed spectral data, and the software sends a command to read the spectra every 300ms to receive and process the data. Then, the energy spectrum data are smoothed, peak-seeking, scaled, etc., to identify their nuclides and calculate their count rates.
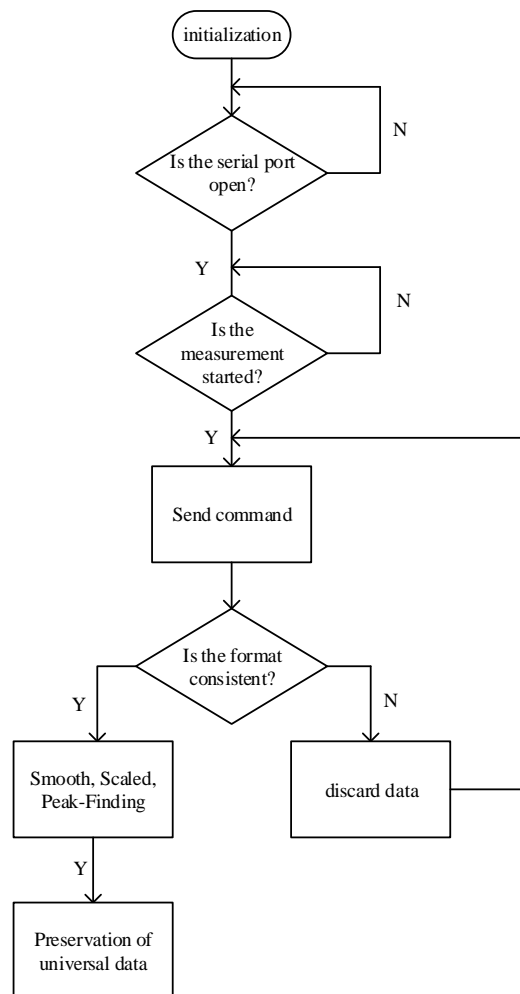

Fig. 7 Software flow chart

(4) Analyzing Algorithm Implementation

The summing of the energy spectrum in the software is the count of the full spectrum, using the cumulative summation method. In order to reduce the randomness of the nuclear signal and the influence of statistical rise and fall on the analysis, the software supports the operation of spectral smoothing, and the method used is the 5-point smoothing formula in the least-squares method[7], as shown below:

$$\bar{y}_i = \frac{1}{35}(-3y_{i-2} + 12y_{i-1} + 17y_i + 12y_{i+1} - 3y_{i+2})$$

where i is the channel number, is the count value of channel i in the smoothed spectral line, and y_(i-2), y_(i-1), y_i, y_(i+1), y_(i+2) are the count values of the corresponding channel of the measured spectral line, respectively. The values are the counts on the corresponding channel sites of the measured spectral lines.

Two types of energy scales are utilized, the least squares linear fit and the least squares polynomial fit. When linear fitting is utilized, the inputs are at least two points and the form of the fit is shown below.

$$E = ax + b$$

where E is the energy after scaling; x is the tract site; a, b, and c are fitting constants.

(5) Bayesian Online Variable Point Detection

The time points at which the radioactive source count rate changes are detected using the movement of the radioactive source orbital platform are detected using Bayesian Online Change point Detection-Bayesian Online Change point Detection (BOCD)[8-10].

As shown in Fig. 8, which represents an example of a data series arriving at a time step, with time plotted on the x-axis and data values plotted on the y-axis, it can be observed that the mean value changes at the 5th time step and changes again at the 11th time step. So in essence, when the above data series occurs, we want to detect these change points at "t=5" and "t=11". Here g1, g2, and g3 can be thought of as three partitions of the data, separated by the change points.
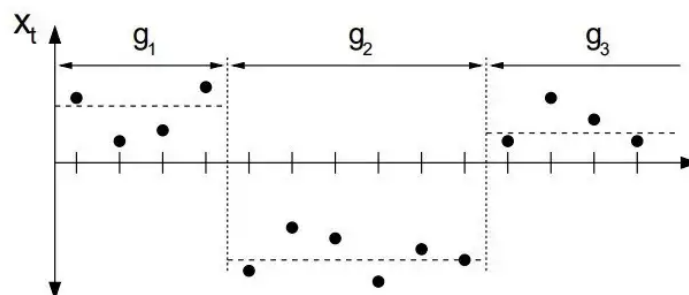

Fig. 8 Running length of data

BOCD models change point detection based on run length. After observing a previous data point, the run length only indicates whether the new data still belongs to the same partition. If so, the tour length for the next time step increases by one (indicated by the black line), otherwise it decreases to zero (indicated by the dashed line). This process is repeated at each time step, as shown in Fig. 9.
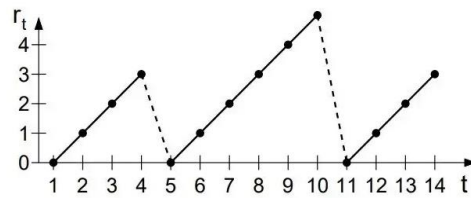
Fig. 9 Running length of data

5. Software Field Testing

After connecting the device to the PC, test the software system, the test site is shown in Fig. 10.
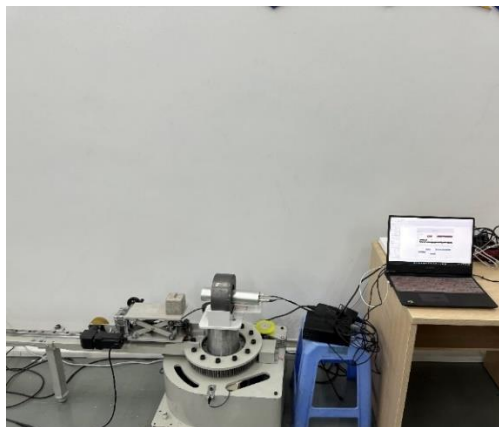

Fig. 10 Test site

Open the software system, select the serial port number of the lower device, open the serial port and transmit the command to the lower device, which receives the command and feeds back the data to the PC. Put the radioactive source on the track platform (take Cs137 as an example), through the track control software, move the track platform, measure the radioactive source during the moving process, plot the blue scatter plot (count rate every 300ms), the red dotted line is the detected variable point, extract the time information of the detection point and record it. The horizontal coordinate is the time and the vertical coordinate is the count rate, as shown in Fig. 11.
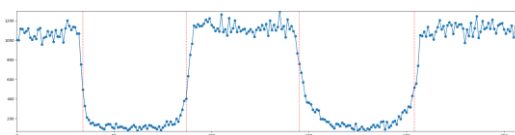

Fig. 11 Test result graph

6. Conclusion

Under the integrated compilation environment of PyCharm, by using Python programming language and its powerful third-party libraries to realize software development, it greatly reduces the research and development cycle, and successfully designs the γ energy spectrum analysis software with good human-computer interface, data processing and display functions, safe, reliable and easy-to-operate, which can be popularized and applied in the digital γ energy spectrum measurement system.

Acknowledgments

References

[1] WANG Lin, WANG Lei, TOU Xiangguo, LIN Mu. Design of γ energy spectrum analysis software based on C#[J]. Nuclear Electronics and Detection Technology, 2013, 33(09): 1080-1084.

[2] Luo Q. Development of software for low background γ energy spectrum analysis [D]. Chengdu University of Technology, 2019.

[3] Yao Xiaoyu, Tou Xiangguo, Shi Rui, Li Qiang. Qt and Matlab hybrid programming method for designing alpha energy spectrum analysis software[J]. Nuclear Electronics and Detection Technology, 2018, 38(01): 70-75.

[4] YANG Yue, LUO Xujia, LI Chu. Design and development of nuclear energy spectrum data processing software based on Matlab[J]. Jiangxi Chemical Industry, 2019.

[5] WU Hexi, LIU Yujuan, LIU Libo, WEI Qianglin, YAN Chuan, GE Liangquan. Design of teaching platform for digital nuclear energy spectrometer[J]. Nuclear Electronics and Detection Technology, 2015, 35(08): 814-816+836.

[6] Zhu Hao. Research on nuclear energy spectrum data acquisition and processing based on Matlab [D]. Nanhua University, 2017.

[7] Y.T. Zhang, Y.M. Jia. Analysis and implementation of algebraic polynomial curve fitting in least squares[J]. Computer and Digital Engineering, 2017, 45(04): 637-639+654.

[8] Bai Qiannan. Research on video summary generation method based on variable point detection [D]. Xi'an University of Technology, 2019.

[9] Guo Weijuan. A Bayesian approach for detecting the mean change points of normally distributed sequences[J]. Journal of Hubei Second Normal College, 2020, 37(08): 8-11.

[10] Habibi Reza. Bayesian Online Change Point Detection in Finance[J]. Financial Internet Quarterly, 2022, 17(4).