

DEVELOPMENT OF ALARM AND MONITORING SYSTEM USING SMARTPHONE

W.S Cho*, Pohang Accelerator Laboratory, Pohang, Republic of Korea

Abstract

In order to find out the problem of the device remotely, we aimed to develop a new alarm system. The main functions of the alarm system are real-time monitoring of EPICS PV data, data storage, and data storage when an alarm occurs. In addition, an alarm is transmitted in real time through an app on the smartphone to communicate the situation to machine engineers of PLS-II. This system uses InfluxDB to store data. In addition, a new program written in Java language was developed so that data acquisition, analysis, and beam dump conditions can be known. furthermore Vue.js is used to develop together with node.js and web-based android and iOS-based smart phone applications, and user interface is serviced. Eventually, using this system, we were able to check the cause analysis and data in real time when an alarm occurs. In this paper, we introduce the design of an alarm system and the transmission of alarms to an application.

INTRODUCTION

The off-line method of the past alarm system sets threshold points for Low, High, and Interlock information, outputs a sound through the speaker when the range is out of range, and the operator immediately analyses the contents and solves the cause. This is a useful solution in field situations. However, engineers cannot see real-time data or alarm information generated by the device.

The new real-time online alarm method selects the EPICS PV [1] data requested by engineers or users in addition to the specified threshold and checks the data or receives an alarm with their smartphone. The new alarm system uses multiple user subscriptions based on EPICS PV. If the user wants to receive an alarm or there is data that he wants to monitor in real time, the PV is registered in the alarm system through the smartphone application. For example, the beam current data of the storage ring is unique. However, individual users may have different desired alarm setting conditions. That is, some users may want to be notified when the current is less than 100mA, and another user may want to receive an alarm when the current is less than 50mA. The alarm system is completely individualized for each user so that alarms are sent or stopped according to different conditions required for each as shown in Figure 1. And when an unexpected failure occurs, the PV connection of the EPICS IOC may be disconnected, so the EPICS PV connection information can also be included in the alarm. The smartphone application to receive the alarm can be used in both Android and iOS environments. We chose Web App because it is important to reduce development time and simplify maintenance. Among the many frameworks used in Web

App, Vue.js [2] is applied. HTML5 and Java script can be easily handled, templates are modular, and switching to mobile or desktop environments is simple. Data can be checked up to 5 hours ago from the present using a line chart. In addition, to check the data of EPICS PV, the data value displayed in the application is updated in real time once every 3 seconds using Web Socket.

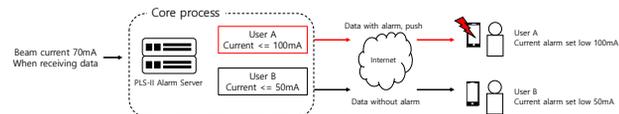


Figure 1 : Alarm system operation overview.

ALARM SYSTEM

Each embedded device provides the user with the necessary data. These data are integrated through the EPICS Gateway server. The alarm system connects to the gateway server through a network by a process called the integrated alarm system core and collects data at a rate of up to 10Hz according to the data cycle updated for each device.

The EPICS JCA library was used to collect PV connection status and data by connecting to the IOC through the network. Users who use the alarm system select a PV to receive real-time monitoring and alarm services through a smartphone application. And when parameters linked to PV are set and saved, data monitoring starts immediately in the core process. All data is stored in InfluxDB while monitoring data. And the data is processed by analysing the parameters set by each user, and if the alarm condition is met, an alarm is immediately prepared to be sent to the user. Alarm information is stored in an event queue running as an asynchronous thread inside the core process. When data exists in the event queue, it immediately analyses the event contents and stores the event log and data from the last hour or so in SQL and the file system. When all processing is completed, a PUSH notification service is sent to an Android or iOS smartphone through Google Firebase service. When the user receives the PUSH notification service, the notification of the installed application is generated, and when the user selects the corresponding event, analysis can be performed together with the past data.

Integrated Alarm System Core

Integrated alarm system core is developed in Java language and executed as a process inside O/S. This process is composed of 5 functions, and the role is divided for each function as shown in Table 1.

* wscho@postech.ac.kr

Table 1: Integrated Alarm System Core Function

Function	Description
PV Status	Check PV status
Data Watcher	Data update interval monitoring
Event Queue	Alarm event information storage
Firebase	Message push service
Web Socket	Real-time data update support

PV status provides the user with IOC PV communication connection status information and PV data. If the PV communication connection state is disconnected, the IOC or internal network situation is not good. In this case, a notification is first sent to the user so that user can immediately check the IOC status or network status.

PV data watcher measures the time of data acquired at regular intervals. This is necessary because IOC works fine, but I have had experience with data not being updated. IOC acquires data through serial or ethernet connection with the device and transmits the data to PV. The general IOC data update period can be set through the SCAN of the EPICS record field. However, if the data is not updated for the time interval set by the user separately from the EPICS field information, it can be inferred that there is a problem in the connection between the IOC and the device. It is not easy to detect this communication problem so far, but through the newly developed alarm system, if data is not updated for a set period of time, it is possible to quickly identify any problem in the connection.

The event queue stores the user ID, time, value, and PV name in the database when the alarm condition is satisfied based on the collected data as shown in Figure 2. And it can be seen that the reason for the need for a user ID is that alarm sending conditions may be different for each user based on the same data.

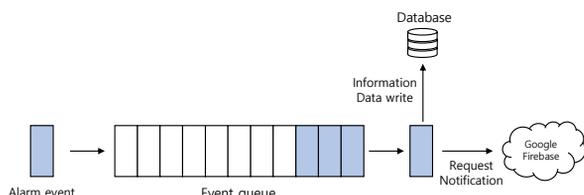


Figure 2: Event queue in alarm process.

Google Firebase [3] provides an integrated environment to send notifications including event contents to various client environments such as Android, iOS, and web browsers.

WebSocket is a technology for the purpose of communication between the user's web browser and the server. It is used in the alarm system of PLS-II and supports sending and receiving messages freely in both directions. Therefore, it supports the function to monitor the current PV data in real time to the user. When the user runs the smartphone application, the user can check the PV name registered in advance and the value updated in real time, so that the user can check the situation in real time.

The inside of the class of the core process consists of PV name, user information, value, time information, and various functions. When the user sets the PV to be monitored, the requested PV creates a new class inside the core process. If another user requests a previously used PV, user information is added from the existing class. When the class is created, PV connection is immediately proceeded, and the following pseudo code is executed according to the alarm range specified by the user.

Pseudo code 1: Alarm condition

```

Data stored in influxDB
if (data < low or data >= high) AND flag then
    if nowTime - lastTime > intervalTime then
        alarm push
    end if
end if
    
```

The pseudo code monitors the PV and when the value is updated, it is saved in influxDB. And according to the first conditional statement, if the range of values is within the range set by the user and it is true to receive an alarm, the next conditional statement is executed. After subtracting the last stored time from the current time, it is compared with the alarm update time set by the user to determine whether to send an alarm. This function uses a simple algorithm in order not to send a large amount of alarm to the user's application whenever the value satisfying the alarm condition changes. For example, the current time is 10 and the previously stored time is 5. And it is assumed that the user sets the interval time to 3. In this case, since the condition is true, an alarm is sent to the user. If the user's interval time setting value is 10, the range of the value satisfies the alarm condition, but it is not transmitted to the user.

Back-end (Server side)

Back-end performs a web server function that responds by processing user requests based on Node.js framework. Web server functions are divided into two main categories.

The user's smartphone application request is processed using RESTful API that is processed in HTTP protocol environment and performs commands such as POST and GET. Node.js can develop programs by distinguishing POST and GET commands from the same route address. When a user requests data, the GET function is used.

The user register function creates a token used by Google Firebase in the smartphone application when a new user is created. The token value is needed to send a notification to the user in the alarm system core process. When the token is sent to the web server through a specified route, the server issues a unique user ID that is a combination of alphanumeric characters and sends it to the application. Based on the issued user ID, the web server processes the inquiry request and provides data to the user.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2022). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI

The alarm history function can be checked by inquiring the alarm information previously stored by the alarm system core process. Alarm information is information stored in the file system based on the user ID and unique time of the PV registered by the user up to 5 hours ago. When the user selects the time information, the stored data is immediately loaded by the web server and the data is transmitted to the user application. Past data shows the entire PV registered by the user. That is, it is assumed that all PVs registered by the user are related.

Create and delete PV monitor registers or deletes the PV to be monitored by the user. When user registers PV, alias, formula application, LOW, HIGH, alarm interval, Y-axis log scale, alarm reception, etc. can be set. And the user PV update function can modify the values set during registration.

User PV list receives the list of all PVs registered by the user. If you use this function, you can check the real-time data along with the PV name in conjunction with the WebSocket session.

It maintains security between the server and the application. When providing web services, external attacks occur, so security must be set. Two-way communication data must be encrypted by setting SSL (Secure Socket Layer) and WSS (WebSocket Secure) as security settings. And it protects from external threats by altering HTTP server information and showing other server information to the attacker.

Front-end (Client side)

Vue.js is a front-end framework and supports SPA (Single Page Application), so it can play the role of a smartphone application. And since it is possible to access the DOM (Document Object Model) interface in an easy way, it is possible to selectively update only the parts that need to be changed in interaction with the user. In addition, since it uses a modular method, it can be easily applied to other templates if a specific page or function is developed.

As a security function of Vue.js, the history function displays an error screen when an arbitrary route is connected without going through a set route.

In the layout design of the smartphone application, as shown in Figure 3, three bottom buttons are applied: HOME, HISTORY, and SETTING.

HOME shows a list of all PVs registered by the user and values that change in real time. If user select a PV name or an alias, user can check the data up to 5 hours ago in a graph. Therefore, even if a specific alarm situation does not occur, users can easily check the real-time information they want anytime, anywhere.

HISTORY shows the past records of alarms that have occurred to the user by alarm occurrence time, value, and PV name. If the user selects this list, user can check the historical data stored before the alarm in a graph.

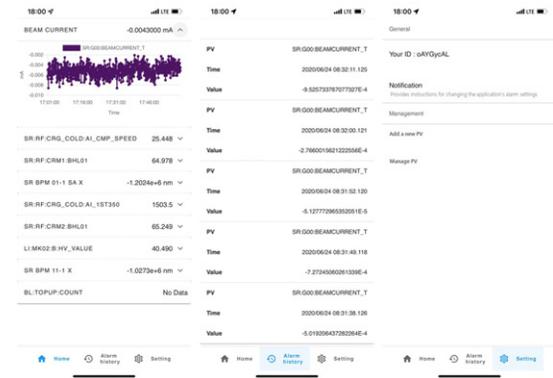


Figure 3: Smart phone application view (iOS).

SETTING can change the user's application usage settings. When an alarm occurs to the user, it is possible to turn on/off the function that the alarm information is exposed at the top of the smartphone screen through the notification function of the application. In addition, the function to add, edit, or delete PVs to be monitored can be performed.

CONCLUSION

In this paper, we introduced a system that supports the real-time alarm function based on a smartphone application. It has become possible to support monitoring functions by individualizing the data of devices that users want and to respond quickly to failures. The read/write performance of InfluxDB time series data in the PLS-II alarm system shows satisfactory performance, but the clustering system is replaced and We plan to upgrade the performance of the data archiving system of PLS-II to collect more data through integration with the alarm system. Currently, the notification function is supported by using a simple algorithm that compares the specified threshold, but it aims to further improve the system through ML (Machine Learning) in the future. The main goal of ML is to predict and detect abnormal operation of the device in advance to strengthen the protection function. When the abnormality detection technology reaches the required level, it will be possible to develop a better system than now by interworking with the alarm system.

ACKNOWLEDGEMENTS

I would like to thank the members of the PLS-II Accelerator divisions for their comments on the system and for testing the program.

REFERENCES

- [1] EPICS, <https://epics.anl.gov/>
- [2] Vue.js, <https://vuejs.org>
- [3] Google Firebase, <https://firebase.google.com>