

WEB BASED MACHINE STATUS DISPLAY FOR SIAM PHOTON SOURCE

N. Suradet, C. Thamtong, C. Preecha, S. Klinkhieo, P. Klysubun
SLRI, 111 University Avenue, Muang District, Nakhon Ratchasima, 30000, Thailand

Abstract

A new machine operation status broadcasting system has been developed for Siam Photon Source (SPS), a 1.2 GeV synchrotron light source in Thailand. The system is implemented using web-based interface, and broadcasts the information over the SPS website, mobile application, as well as local TV network within the SPS facility, allowing users as well as technical personnel to easily access a variety of information related to the machine via web browsers and other mediums. The new system also provides supporting message services for alarm, event notification, and other operational necessities. In this report, the design of web and mobile applications, which are based on HTML5, CSS3, and adopts PHP, AJAX, Bootstrap framework (for responsive design), jQuery, High charts JS, Twitter widget, and others, will be described. The details of the hardware and software configurations, users requirements and satisfactions, as well as suggestions on further improvements, will be presented.

INTRODUCTION

The Siam Photon Source (SPS) is a synchrotron light source operated by Synchrotron Light Research Institute (SLRI), and is located in Nakhon Ratchasima, Thailand. The first light was achieved back in December 2001. At present, the machine is operating at 1.2 GeV in decay mode with a maximum electron beam current of 150 mA. Three insertion devices, a permanent magnet planar undulator, a hybrid multipole wiggler, and a superconducting magnet wavelength shifter, are currently in operation, providing synchrotron radiation from infrared to hard x-rays to synchrotron light users.

The original machine operation status broadcasting system was developed back in 2000, providing the operation status of the machine, for e.g. beam current, beam lifetime, beam energy, to users, who can access the provided information through the internal cable TV system within the facility. Each display channel receives the machine status data from a LabVIEW program located on a computer server. Since this system was available only for on-site users, another system was developed in 2006 to provide the machine status information via the internet. The fundamental language used to create this web-based system was basic static HTML. The displayed beam current and lifetime chart was captured from a NI LabVIEW window.

This web-based system has two main disadvantages. First, it consumes quite a bit of the network bandwidth because the whole web page had to be constantly updated, and the size of the chart image was quite large. Secondly, the system cannot display the data in real-time. We found it necessary to develop a new system that is more robust, more responsive, and more accessible. The new system has to meet the following requirements:

- The core system is based on web technology.
- The web layout is able to present the contents clearly and accurately across multiple types of devices (PC, mobile phones, tablets, etc.) with diverse display resolutions.
- The data is constantly updated every 5 seconds, but the network traffic must be kept low.
- The beam current and beam lifetime chart is generated by the browser on the client side. The displayed data can be exported to a CSV file.
- The system is capable of broadcasting notification messages.

SOFTWARE ARCHITECTURE

The machine status data originates from a variety of sources. These sources/hardwares are interconnected via an assortment of interface standards (OPC, GPIB, RS-232, etc.). A data logging program written with LabVIEW and installed on an acquisition server is employed to continuously gather all the machine data and log them into a database. The logging interval is 5 seconds. Open source database MySQL [1] was chosen for our purpose. LabVIEW MySQL connector toolkit [2] allows LabVIEW to communicate with MySQL (version 4.1 or later) via the TCP/IP protocol. It is a part of the LAMP (Linux-Apache-MySQL-PHP) platform that has to be installed on the web server.

When the user opens the SPS machine status web page, the browser on the client side will make a request for the PHP webpage to the web server. The web server responds by sending HTML, JavaScript, and CSS scripts to the client for processing, so that the execution is performed by the client browser. We use AJAX (Asynchronous JavaScript And XML) [3,4] to help refresh the web page for updating the data. AJAX runs a background operation which extracts the data from the database in the XML/JSON data format every 5 seconds. It updates the data field of the web page without reloading the whole page, thereby substantially reducing the traffic demand on the network. Fig. 1 shows the architecture of the machine status broadcasting system.

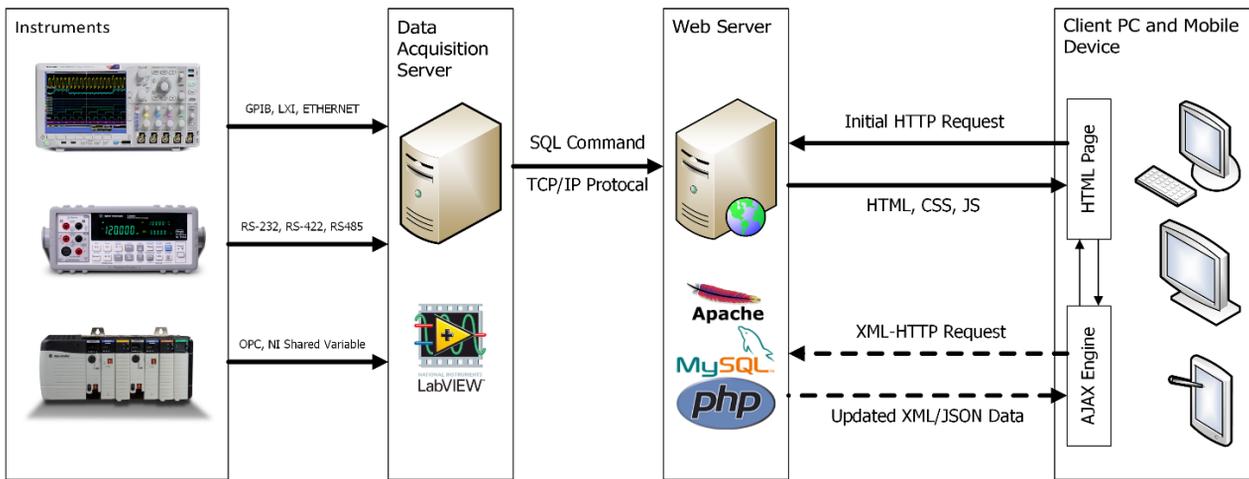


Figure 1: Machine status broadcasting system architecture.

DESIGN AND CODING

We employ various web-based technologies and frameworks to create the machine status webpage. These tools, which help make the code simpler and more efficient, are as follows.

Frontend Framework with Bootstrap

Regarding our aforementioned requirements that the new system has to be able to display the machine information correctly across all supported devices, and it must be relatively easy to develop, we employ the open source Bootstrap framework [5] to help manage the front-end. The Bootstrap package includes Scaffolding, Base CSS, Components, and JavaScript plug-ins. It handles display duty across all the devices with multiple resolutions, making the machine status webpage display-responsive, as shown in Fig. 2.

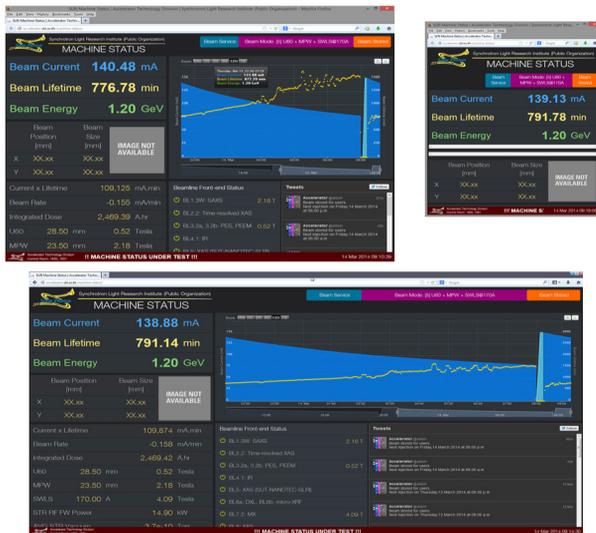


Figure 2: Machine status webpage at different screen resolutions. (Top left: 1280x1024, top right: 640x800, and bottom: 1920x1080)

Asynchronous Communication with AJAX

AJAX (Asynchronous JavaScript And XML) is a tool for creating fast and dynamic web pages. AJAX allows web pages to be updated asynchronously by exchanging small amounts of data with the server behind the scene. This means that it is possible to update specific parts of a web page without reloading the whole page, significantly reducing traffic load placed on both the server and the network.

Real-time Graph with Highcharts JS

The SPS beam current and beam lifetime chart which shows a historical record over 24-hour period contains the recorded values of the beam current, beam lifetime, beam energy, and the associated timestamp. We use Highcharts JS [6], which is a JavaScript framework for creating charts. Its process is performed on the client side (browser). A data buffer is created and stored in the MySQL database along with the associated timestamp, which is installed on the web server.

The data is extracted from this database through a PHP query program. AJAX together with jQuery manage refreshing of the chart, which is a JSON data format, resulting in an auto-refresh real-time chart. The process is illustrated in Fig. 3.

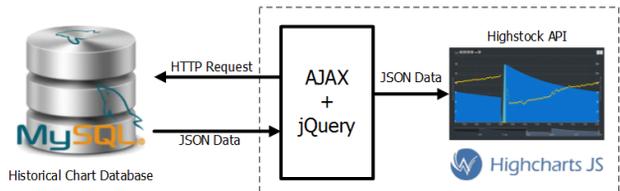


Figure 3: Software architecture of the auto-refresh real-time beam current and lifetime graphs.

Notification Broadcasting with Twitter

Incorporating a notification broadcasting into the system was deemed necessary from the beginning since it would be the main channel facilitating the communication from the control room to the users. For this we chose

Twitter [7], a social networking platform that allows one to send short messages (less than 140 characters in length) to followers. Twitter has a widget for displaying a message on the website and also delivering push notifications to a mobile device. The tool enables machine operators to send crucial machine status updates out instantaneously.

MOBILE APPLICATION

An iOS web application was written for iOS-based devices. The web app can be installed from the Safari web browser, and can be accessed later from the iOS home page, as shown in Figure 4(a). The program simply retrieves the machine data from the machine status webpage. When the condition that the user agent is a mobile device is detected, the system will display a specialized frontend framework containing the Slidebars [8] and a navigator menu, as shown in Figure 4(b). The machine status display remains on Bootstrap as mentioned earlier; however, the information is separated into multiple pages to fit the screen resolution of the mobile device, as shown in Figure 4(c).

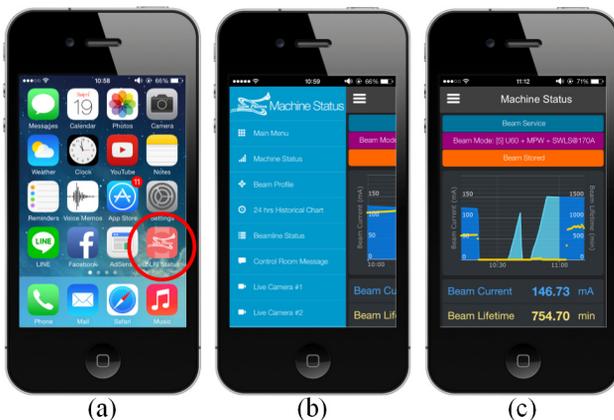


Figure 4: Mobile application: (a) SPS machine status app icon, (b) Slidebars navigator menu, and (c) SPS machine status on mobile device.

HARDWARE CONFIGURATION

Both the data acquisition server and the web server are based on two IBM System x3250 M4 servers having Intel Xeon E3-1270 3.4GHz (8MB L3cache), 8GB of system memory, 2xSATA drive of 250GB with RAID 1 (mirror) configuration, and gigabit ethernet. The operating systems are Windows 2008 Server and Ubuntu Linux, respectively.

CONCLUSION

The new SPS machine status broadcasting system has been in use for approximately 8 months since February 2014. The system is found to be robust, effective, and user-friendly. Future plan includes making it available for devices based on other platforms.

ACKNOWLEDGEMENT

The authors are thankful to all the members of Accelerator Technology Division and Technical Support teams for their support, constructive remarks, and feedbacks.

REFERENCES

- [1] MySQL website: <http://www.mysql.com>
- [2] LabVIEW TCP/IP Connector for MySQL Data base, <https://decibel.ni.com/content/docs/DOC-10453>
- [3] AJAX website: <http://www.w3schools.com/ajax>
- [4] Y. S. Cheng et al., "Upgrade the machine status broadcast system by PHP framework", THP113, Proceedings of ICALEPPCS2009, Kobe, Japan (2009), <http://jacow.org/>.
- [5] Bootstrap website: <http://getbootstrap.com>
- [6] High charts - Interactive JavaScript charts for your web pages, <http://www.highcharts.com>
- [7] Twitter website: <https://twitter.com>
- [8] Slidebars website: <http://plugins.adchsm.me/slidebars>