

# PRELIMINARY DESIGN AND INTEGRATION OF EPICS OPERATION INTERFACE FOR THE TAIWAN PHOTON SOURCE

Y. S. Cheng, Jenny Chen, P. C. Chiu, C. H. Kuo, C. Y. Liao, K. T. Hsu, C. Y. Wu  
NSRRC, Hsinchu 30076, Taiwan

## Abstract

The TPS (Taiwan Photon Source) is the latest generation 3 GeV synchrotron light source which has been in construction since 2010. The EPICS framework is adopted as control system infrastructure for the TPS. The EPICS IOCs (Input Output Controller) and various database records have been gradually implemented to control and monitor available subsystems of the TPS at this moment. The subsystem includes timing, power supply, motion controller, miscellaneous Ethernet-compliant devices etc. Through EPICS PVs (Process Variables) channel access, remote access I/O data via Ethernet interface can be observed by the useable graphical toolkits, such as the EDM (Extensible Display Manager) and MATLAB. The operation interface mainly includes the function of setting, reading, save, restore and etc. Integration of operation interfaces will depend upon properties of each subsystem. In addition, the centralized management method is utilized to serve every client from file servers in order to maintain consistent versions of related EPICS files. The efforts will be summarized at this report.

## INTRODUCTION

The TPS [1] is a latest generation of high brightness synchrotron light source which has been under construction at the National Synchrotron Radiation Research Center (NSRRC) in Taiwan since 2010. It consists of a 150 MeV electron Linac, a 3 GeV booster synchrotron, and a 3 GeV storage ring.

The EPICS (Experimental Physics and Industrial Control System) is a set of open source software tools, libraries and applications developed collaboratively and used to create distributed soft real-time control systems for scientific instruments such as the particle accelerators, telescopes and other large scientific experiments [2]. In the field of accelerators, many facilities have good practical experiences for EPICS and adopt it as the accelerator control systems. Many resources and supports are available as well as numerous applications for accelerator have been developed.

As a result, the EPICS framework was also selected as control system infrastructure for the TPS project. The EPICS platform has been gradually built and tested to control and monitor the subsystems of TPS. The various database records can be created for accessing the I/O data and setting parameters at the IOC (Input Output Controller) layer. Adopting the EPICS channel access mechanism with specific toolkits, the data can be accessed between the IOCs and the clients.

During the implementation process of the EPICS support for various subsystems, the operation interfaces of each subsystem are also developed according to the different operation methods. To simulate the operation process, the various operation interfaces are needed to integrate. The centralized management of the EPICS related files is also adopted, and the mechanism of save and restore will be continuously developed. The efforts will be summarized as following.

## SOFTWARE ENVIRONMENT

The client consoles are adopted the Linux operation system. All of the EPICS base, modules and extensions are installed at the Linux system. The software versions are shown as Table 1 where most of applications are developed base upon this software environment.

Table 1: Software Environment of the Control Consoles

	Version
OS	RHEL 5.4 (32-bit) (kernel 2.6.18)
EPICS	base-3.14.10
Modules	asyn-4.11a StreamDevice-2.4
Extension	edm-1.12.xx labCA-3.1

## FILES MANAGEMENT

All EPICS related files at control consoles are mounted from the file server by using the NFS service [3] to simplify software version control. Various directories are created and saved into various versions of related files for various hosts and purposes. Various directories provide a mount point for hosts mounted according to various purposes. The directories include EPICS base, modules, extensions, saved data, temporary data and etc.

Several file servers are established to share the loading of NFS file service. The hosts mount specific file server according to its location and purpose. By loading testing, the NFS file service is divided into three parts. Two servers provide the NFS service for hosts of all cells; the other server is for engineer development.

To keep the same version at each NFS file server, the “rsync” function is used with the “crontab” schedule service [4]. One main NFS file server is established for development purpose; the other NFS file servers are adopted “rsync” function to synchronize all data from the main NFS file server with using the private Ethernet

communication. The system architecture is shown as the Fig. 1. The dedicated spare NFS file server is also established for redundant purpose, and the related data is synchronized from the main file for backup purpose.

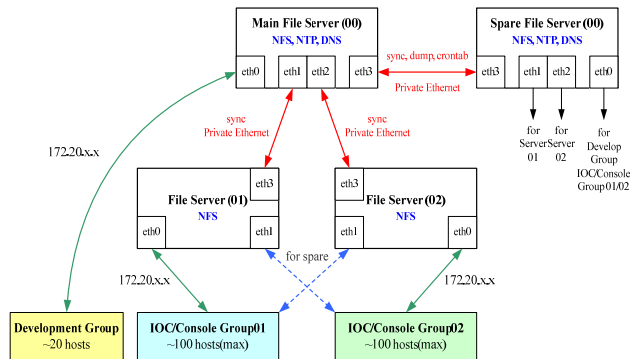


Figure 1: The system architecture of file servers for EPICS related files mounted with backup mechanism.

### SAVE AND RESTORE

To readily restore a set of the machine parameters for subsystems during operation as well as to optimize and record working points for different machine condition, the mechanism of save and restore is developed. The save and restore function is initially built by using the MATLAB with labCA [5]. The various files of grouped PVs (Process Variables) list are created for saving the respective parameter values of each subsystem. The file with PVs and saved parameters is also selectable for resume the settings. The preliminary interface of save and restore mechanism is shown as the Fig. 2. Both of consumed computer resource and process time are acceptable after preliminary testing.

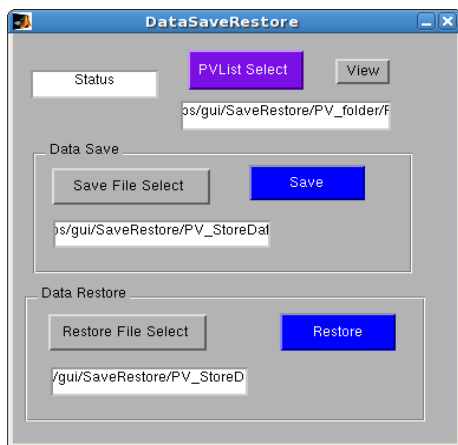


Figure 2: The operation interface of save and restore.

### PRELIMINARY CONTROL PAGE

At the development phase, the GUI of TPS control system adopts the EDM (Extensible Display Manager) toolkit to develop main graphical operation interface. The EDM pages of various subsystems are created and saved into the NFS file sever for client console operation.

The EDM is an interactive GUI builder and execution engine, the EPICS documentation uses the term display manager, and maintained by the ORNL EPICS community [6]. All “objects” of EDM are loaded from shared libraries. The EDM administrator can add and remove objects from the list of available objects without recompiling EDM itself. The objects are versioned, carefully coded objects can be upgraded without impacting existing displays.

The preliminary main control page is built by the EDM toolkit shown as the Fig. 3. All control pages can be launched from this GUI. All control components are located at the foreground of the TPS accelerator illustration. For example, the LTB (Linac to Booster) dedicated control page is linked from the main control page for operation shown as the Fig. 4. The related control parameters or components are also located at the same page for tuning easily, such as the control page for the TPS timing as shown in the Fig. 5. These pages will be continuously refined and developed for the future commissioning.

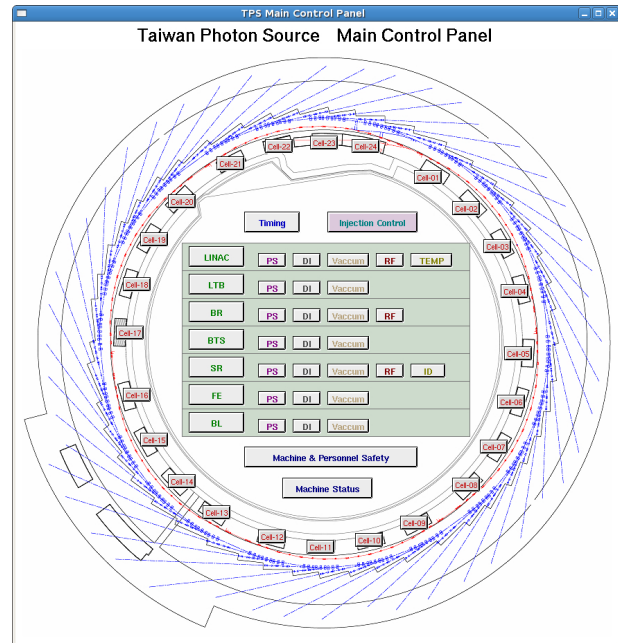


Figure 3: The preliminary TPS main control page.

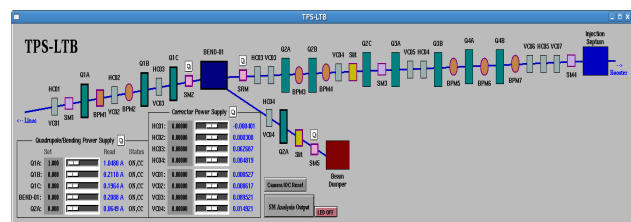


Figure 4: The LTB control page for operation test.

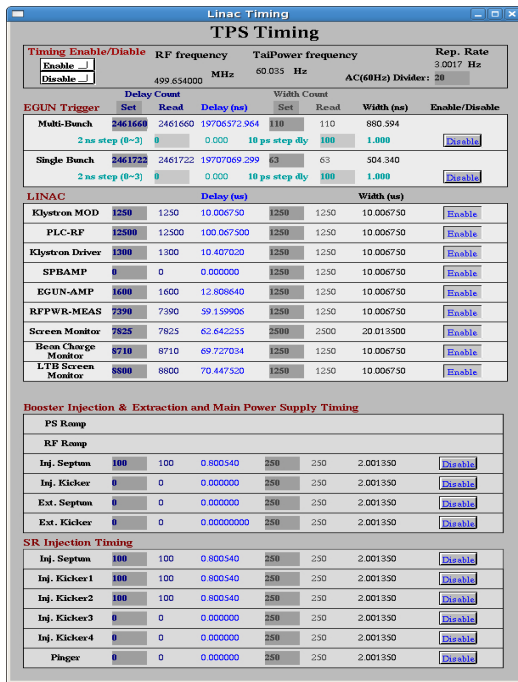


Figure 5: The overview control page for the TPS timing.

## POWER SUPPLY OPERATION INTERFACE

The EDM toolkit is also adopted to develop the operation interface of power supplies. The client console can operate the specific EDM page to access the data via PVs channel access. The preliminary GUI page of storage ring dipole, quadrupole and sextupole power supplies controls as shown in Fig. 6. The Fig. 7 shows the GUI page of storage ring vertical corrector, horizontal corrector and skew quadrupole power supplies controls. The macro name method was regularly used to switch each display page. The main control page was shown critical information for observing status easily, and the main operation process functions are also executed from the panel.

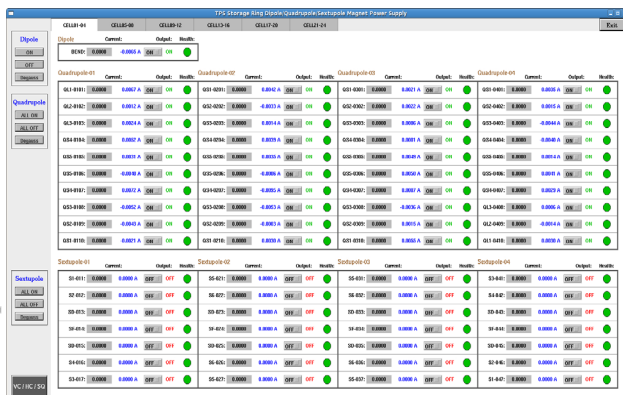


Figure 6: The control GUI of storage ring dipole, quadrupole and sextupole power supplies.

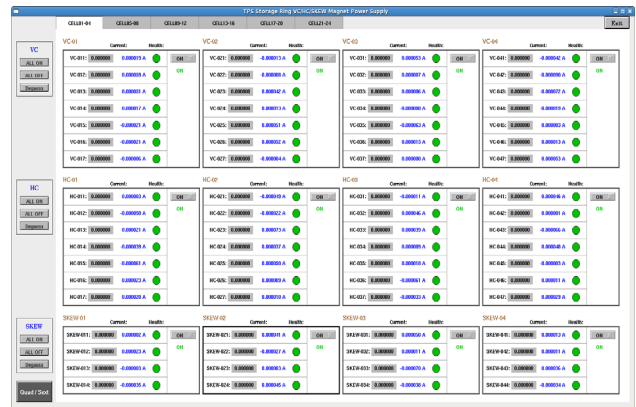


Figure 7: The control GUI of storage ring vertical corrector, horizontal corrector and skew quadrupole power supplies.

The MATLAB toolkit with labCA is adopted to develop the high level application programs for commissioning and diverse operational procedures. The applications include the specific overall power on/off control, degauss process, checking power supply status, operation performance analysis, operation statistics and etc. The various operation processes will be developed and tested according to the various operation modes. The detail control page of power supplies has the trend plot for observing. The Fig. 8 shows the current variation during the degauss process executed. The degauss application is also developed with the specific function of batch process to reduce the peak of power consumption for saving energy.

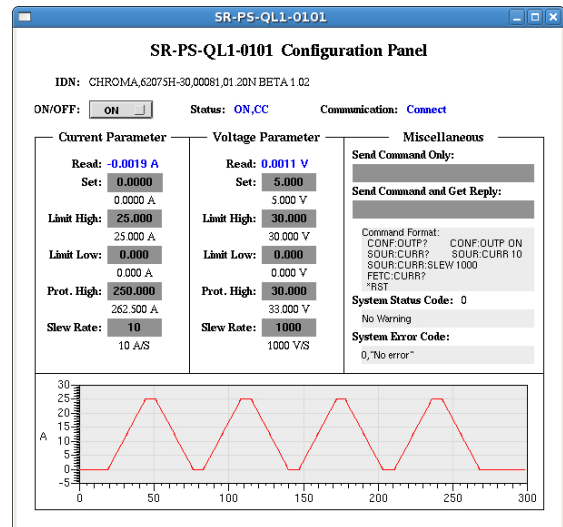


Figure 8: The current trend is shown at configure page of a quadrupole power supply during the degauss process test.

## ARCHIVE

To record the variation of specific subsystem for long time observation, the archive can be use to save data. The EPICS channel archiver [7] is under research and evaluation. The archive view is the java based toolkit, and

can be easy to use at the different operating system platforms. The server loading and saved files size of archive system will be estimated in the future.

### SUMMARY

The TPS control system adopts the EPICS framework as the control infrastructure. The EPICS supports of various subsystems are built gradually. The operation interfaces of various subsystems are also developed in the meantime. The various GUIs are designed according to the different operation methods. The management of the EPICS related files system is also established for maintaining files versions easily. The mechanism of save and restore are needed to apply parameters for operation process. Various operation interfaces will be integrated and improved continuously during the TPS control system construction.

### REFERENCES

- [1] <http://www.nsrcc.org.tw/english/tps.aspx>.
- [2] <http://www.aps.anl.gov/epics/>.
- [3] [http://en.wikipedia.org/wiki/Network\\_File\\_System\\_\(protocol\)](http://en.wikipedia.org/wiki/Network_File_System_(protocol)).
- [4] [http://kevin.vanzonneveld.net/techblog/article/synchronize\\_files\\_with\\_rsync/](http://kevin.vanzonneveld.net/techblog/article/synchronize_files_with_rsync/).
- [5] <http://www.slac.stanford.edu/~strauman/labca/>.
- [6] <http://ics-web.sns.ornl.gov/edm/>.
- [7] <http://ics-web.sns.ornl.gov/kasemir/archiver/>