FERMILAB MAIN INJECTOR LCW CONTROL SYSTEM

Seung-chan Ahn†, FNAL, Batavia, IL 60510, USA

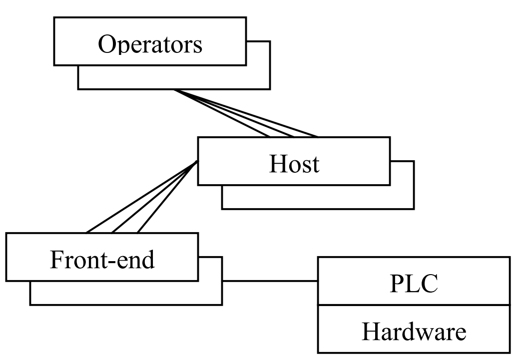
Abstract

The Low Conductivity Water (LCW) system cools the conventional magnets at the Fermilab Main Injector (MI) and their instrumentation support structure. This paper describes the LCW control system with an emphasis on the software layer closer to the user level.

INTRODUCTION

There are six service buildings along the perimeter of MI and they collectively use about 8000 gallons of de-ionized water. The de-ionization system is housed in a central utility building near main ring and another satellite system is utilized to further polish the water. The LCW system uses de-ionized water whose resistivity is greater than 10 MΩ, much greater than that of untreated water of the order of 10^2 to 10^3 Ω. The high resistivity of the cooling fluid renders a good thermal conductivity with a good electric insulation so as to prevent adverse effects like ground fault and other electrical interference. The LCW system facilitates efficient cooling of the conventional magnets and their instrumentation components such as electronics, bus, power supplies and others [1].

The purpose of LCW is to maintain the optimal level of pressure, temperature, flow rate and resistivity. It is done with the continuous monitoring of the LCW parameters and controls at various levels, from automatic control by Programmable Logic Control (PLC) to the operator initiated, order-sensitive chain of commands. The physical actions of these commands are usually switching on/off of valves and pumps. The functional diagram of the LCW system is shown below schematically.



The diagram illustrates multiple links between operators, host computers (VAX) and the front-end computers. Each PLC is connected to a single front-end computer and specific set of hardware elements. The control and monitor of LCW is distributed over a wide range of localities, from the main control room to the field operation.

†ahn@fnal.gov

SOFTWARE COMPONENTS

From the software point of view there are three functional components in the LCW system. They are described in the following starting from the lowest level.

PLC

At the lowest level is the PLC, in direct contact with the LCW hardware such as pumps and valves that control the water flow to maintain optimal system parameters. PLC does this by controlling actuators of the LCW hardware. PLC also controls sensors that report back physical attributes like flow rate, temperature, pressure and resistivity. PLC runs semi-autonomously and takes the proper action when predetermined conditions are met. Functionally speaking, for external users, the PLC is synonymous to the hardware elements. As such, when PLC is in certain states, it can be turned on after certain specified condition is met. This is typically done by forcing the user follow a certain sequence of operation.

The PLC is an autonomous computer in its own right; endowed with the ACNET, a Fermilab developed network protocol, network capacity. The PLC is equipped with safeguard features such as interlocks. PLC consists of a Gateway module and a variety of I/O modules. One programs PLC using languages expressible in various forms. The PLC can be programmed in graphics form by manipulating control/monitor components. It is called "Sequential programming". The PLC can also be programmed in a C-like language. [2]

Front-end Computer

From the LCW perspective the front-end computer is the bridge between PLC and host computers. All control command and monitor data go through the front-end computer. The front-end computer is built around Motorola 68000 microprocessor running on a real-time operating system VxWorks. The standard development environment offers efficient programming and quick download of the program to the front-end computer.

The role of intermediary is not the sole or even major function of the front-end computers, as it does a variety of other services as well. A typical front-end computer does more than serving LCW or any specific set of devices. An important function of the front-end computer is reporting alarms, based on the preset parameters. At 15 Hz the front-end computer scans all the devices in the list and checks their values against the preset parameters. If a device reading is out of tolerance for a preset period of time, then an alarm is issued along with relevant data to a central operational computer via ACNET and TCP/IP.

The front-end computer gets the list of devices to monitor immediately after the booting. The front-end computer does not issue alarms for all the devices it monitors, however. The list of devices to merely monitor and those to also report alarms are maintained in the central database built on a commercial database product Sybase, which runs on a Unix machine.

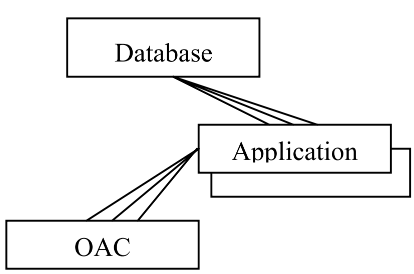
Host Computer

Operators in the main control room interact with the LCW system only through the host computers that are networked to consoles but physically isolated in an adjacent computer room. Others can also do the same on other host computers located throughout the laboratory.

Nearly all application programs on the host computers run in a controlled environment known as Console system. The Console offers a wide range of services to the application programs while performing the task of system maintenance. The service application can expect from the Console includes basic necessities like report of memory and I/O states. The Console also provides hooks for run-time functions. Roughly, as far as operation of the facility is concerned, the Console is somewhat similar to X-window system. Typical Console program will have user input capabilities and the user input it typically initiated with mouse action at various points in the display. Usually, the location of monitor is where the control occurs.

LCW programs at MI and Tevatron, while written and maintained separately, share much of the programming details, and within the Console environment one can switch back and forth between the two programs. LCW is further divided into a number of pages, corresponding to each physical LCW stations.

The main programs for LCW, one for MI and another for Tevatron conventional magnets, are linked as they share some features and are similar in their functionalities. Their user interface and overall architecture is also similar. Nearly all programs running on the host computer are a distributed system and the LCW host program is no exception, as shown below schematically without legs to external components.



1. User Interface

To users, the Console is organized in terms of “pages”. Pages with similarities are grouped together under a common heading such as “D-page”. The navigation over the set of pages is intuitive and efficient, although the look and feel may appear somewhat dated. The reason for this particular look and feel is because the Console graphics is essentially alphanumeric, i.e., it does the graphics with pre-defined graphics elements except for applications that require flexible pixel manipulation such as data presentations in time-display, scatter-plot and histograms. By judiciously choosing the modes of pixel manipulation, the Console performs the core tasks of the machine monitor and control quite efficiently. The most visible services a typical application receives from the Console are the management of visual data and links to other related programs.

Like most Console applications LCW host programs share the same display space for both input and output. User may send action command such as operation of valves and switches, and user may modify setting parameters that controls PLC operation. All user input is triggered by a mouse action on the designated place in the display. With extremely rare exception LCW displays data and accepts user command at the same area in the display area. Lastly, the Console decorates the LCW application with menu bar that provides additional services such as link to other utilities such as help, print, graphics and etc.

2. Utility Software for LCW

The two LCW programs running on the host computers obtain the device properties from the central database (DB). They can receive data from Open Access Client (OAC) or directly from the front-end computer.

From the application’s point of view DB is a collection of device attributes, e.g., a knob or meter of a physical device. A device can have multiple properties such as state, control, alarm, and reading but is not required to have them all. Since the device attributes are ultimately mapped to the front-end computer’s memory, the address is sufficient to establish a device. A utility program DABBEL, a homemade script language, is used to create, modify and delete devices in DB. DABBEL mediates users and DB engine. With utility programs like DABBEL it is relatively easy to create new devices corresponding to a physical device or software device that contains an aggregate of attributes. Currently there are about 200,000 devices collectively used to operate the accelerator facility.

An integral part of the host computer is OAC, which collects data from registered devices and serves it to applications. When application reads a large number of similar devices in an array, it can read them individually from the front-end computer or it can read the same data from OAC. When multiple applications or multiple instances of the same application require the same data it makes sense to delegate actual reading to OAC. This is the typical mode of the usage of applications. OAC helps reduce network traffic and offers other system related services for robust operations.

OUTLOOK

Fermilab Accelerator Control System had undertaken software system migration from the minicomputer system of pre-PC era to a new generation of commodity PC control system. The fundamental design of the new system is completed and a large-scale implementation of the new system has been made. As such, we are currently undergoing a switch to a new Java based system.

The most notable example is the new Java based Data Acquisition Engine (DAE) [3] that plays the critical role in the data flow throughout the system. The implementation is not limited to data flow but extended to an intelligent data manipulation such as calculation of certain quantities with inputs from multiple devices and re-direction of data between various contact points. An example is presented at this conference [4].

The new software environment will have significant impacts on the LCW host level software and possibly even the front-end computers. It is likely that the host program will be rewritten and/or translated into the new software environment to take advantage of all the benefits the new system offers. What will be retained is the functionality derived from the low end with the sensors and actuators, and familiarity at the high end with the human interaction of the system. All else in the mid-tier will be retooled in the new software environment to achieve enhanced performance with less maintenance cost.

REFERENCES

[1] http://www-bdnew.fnal.gov/operations/rookie_books /Main_Injector_RB_v1.pdf, Chapter 4: Water in the Main Injector

[2] K.C. Seino, Main Injector LCW (Low Conductivity Water) CONTROL SYSTEM. ICALEPCS 2001.

[3] http://www-bd.fnal.gov/controls/public/ JavaControls. htm K. Cahill and C. King: Guide to the Data Acquisition Engine.

[4] T. Bolshakov, A. Petrov and S. Lackey: Synoptic Display—A client-server system for graphical data representation, ICALEPCS 2003.