International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Analysis and Design of Real Time Data Acquisition System for Critical Analog and Digital Parameters in Nuclear Power Plant

Sachin Kumar Patidar

MS by Research (CSE), Dept of CSE, University College of Engineering, Osmania University, Hyderabad, 500007

ABSTRACT

Data is the basis of making any decision. Whether the Data is economical, political, environmental, or Process control related, it is required to be acquired in a precision manner and used for Control / monitoring. Data Acquisition Systems used in Power Plants determines the changing status and behavior of process of power generation, foreseeing impacts, and managing what can be controlled for the maximum benefit to the plant with minimum risks and human interaction. This paper provides the general ideas of developing the Real Time Data Acquisition System with scan rate up to 1 millisecond. First section of the paper concentrates on the introduction of DAS and RTDAS. In next section it exposes the analysis of RTDAS. Last section of the paper presents the design issues of the RTDAS. This paper mainly concentrates on RTDAS for critical analog (Disturbance Recorder) and digital (Event Sequence Recorder) parameters in nuclear power plant.

Keywords: Data Acquisition System, Real Time Data Acquisition System, Event Sequence Recorder, Disturbance Recorder, Recording Unit, viewing Unit

Data Acquisition System (DAS) A data acquisition system is a collection of hardware and software that connects to the physical world. **Real Time Data Acquisition System (RTDAS)**

Real Time Data Acquisition System (RTDAS) is a High-speed Data Acquisition System, which monitors and records various parameters important to Plant's safety and Operation. The recorded information is used to analyze/ study/ monitor the important parameters of a plant disturbance where pre disturbance and post disturbance data are required.

PROBLEM STATEMENT

Generation of power from Nuclear power plants needs monitoring of various safety parameters such as temperature, pressure; Heavy water loss. The Data acquired from the plant plays a pivotal role in deciding the safety of the nuclear power plant. Before the Advent of the computers the data is acquired manually.

Manually acquisition involved sensors were used to get vital information. There were meters, which were to be operated manually to get critical information regarding the value at a particular channel or regarding the particular safety parameter. The operator had to adjustor the sensor or meter so as to get the value at the particular channel and log down the values and other details. It was not possible to get the values of all the channels in real time. The logged data was not reliable as it was prone to human error.

In order to know the Trends of the various points one had to plot the values on the paper and see the trend at a particular channel. The availability of the information was confined to the place where recording was made. With the advent of the computers there aroused the thought of automating the data acquisition. First the Data Acquisition was done using single task operating system like DOS. Though DOS has support for interrupt It was not user friendly. With the coming of the WINDOWS the functionality of Data Acquisition Systems was not limited to mere acquisition of data from the sensors but it added many features like Graphical User Interface, better user control and so on. The Operator was able to get data on the fly for analyzing and making decision. But with the coming of LINUX which being Open source provided cost effective solution. Linux being more sophisticated, stable, efficient, robust, modular, and highly configurable offers full exploitation of reliable, deterministic real time data acquisition and presentation capabilities. In this current project Embedded Linux was used in the acquisition of various signals from the field of the nuclear plant. The Acquired data trends had been displayed using Graphical Trend, Tabular Trend.

SYSTEM ANALYSIS

Real time data acquisition system (RTDAS) is a Highspeed Data Acquisition System, which monitors and records various parameters important to Plant's safety and Operation. The recorded information is used to analyze/ study the important parameters of a plant disturbance where pre disturbance and post disturbance data are required. The main functions of RTDAS are:

1) Acquiredifferent types of Plant Inputs and process them.
2) Support normal recording, disturbance recording and event sequence recording.
3) User Friendly Display and Printing.
4) Support for Offline Analysis of recorded data.
5) Synchronization with Central Clock System.
6) Providerelayoutputcontactsforalarm indication.

Based on the analysis of functions to be implemented, Real Time Data Acquisition System is divided in to different subsystems based on the major functions to be implemented. The subsystems are Event Sequence Recorder (ESR), Disturbance Recorder (DR), Recording Unit (Server) and Viewing Unit (VU).

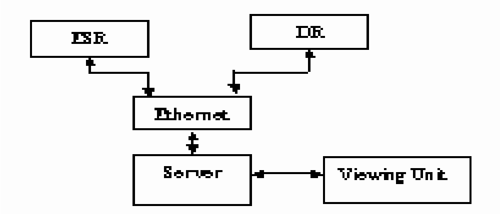

Figure4 Configuration Diagram of RTDAS

1. Event Sequence Recorder Subsystem Infrastructure Requirements

This subsystem caters to the ESR requirements of the RTDAS. ESR comprises of

- ESR I/O controller, an Industrial grade PentiumIV PC with a PCI Controller board, PCI DIO board and network interface ports
- I/O Bin with mother board, eight 32-Channel Digital Input boards and an ESR I/O Interface board
- A dedicated on-line printer (dot-matrix)
- Interconnecting cables and passive interfaceboards

Functional Requirements

ESR performs the acquisition of data from 256 digital channels, detects the events and records the events with their time of occurrence. The PCI controller board, which is a programmable and intelligent controller board, shall handle the main acquisition part by interfacing the 8 DI boards. The board shall be configured with Scan Period, time delays for each channel, Interrupt cycle time for data read, etc. The board shall accept auto-scan, Test 0, Test 1, Force All Read, etc. commands at run-time and dynamically perform the functions based on the commands. When configured for auto-scan (default mode), the board scans all the 8 cards for 256 channel data and detects the status at the configured scan rate. The events detected after acquisition and processing shall be stored in the resident memory. The main I/O controller, ESR PC, shall read the memory and record the events on to permanent storage.

ESR shall construct records with the Events detected sequentially and their time stamps. Each record shall have 4000 events and on collection of 4000 events, a record file, holding those 4000 events, will be created with a file name, preferably referring to the date of creation. These events as and detected shall be printed on to the ESR online printer.

ESR shall print all the events on the dedicated on-line ESR Printer. ESR shall evaluate a set (six) of expressions to trigger the Disturbance Recording INITIATINGEVENT. DR subsystem shall receive this triggering event and start the collection of post-event data of disturbance record. ESR shall drive a Digital Output specified, in PCI DIO board, for this DR INITIATINGEVENT and DR shall sense the contact to read the event. ESR shall drive a set of annunciations, such as Group Alarms (10 Nos.), ESR in Alarm, ESR Failed and ESR Network Link failure through the PCI DIO board. ESR shall send the events and status of the 256 inputs to the Server, where in the permanent storage of records shall be maintained and users can view the data, over the Ethernet at every second.

Performance Requirements

-ESR time Resolution shall not be more than 1 millisecond.

2. Disturbance Recorder subsystem Infrastructure requirements

This subsystem caters to the DR requirements of the RTDAS. DR comprises of

- DR I/O controller, an Industrial grade PentiumIV PC with a PCI I/O Interface board, PCI DIO board and network interface ports
- I/O Bin with mother board, five 32-Channel Analog Input boards and an I/O Interface
- Interconnecting cables and passive interfaceboards

Functional Requirements

Disturbance Recorder subsystem performs the acquisition of data from 128 Analog Inputs, verifies the data and converts the raw data to the Engineering Values. The main I/O controller, DR PC, shall generate the read cycles on the ADCs (in each 32channel Analog Input board) through the PCI I/O

Interface board and acquires the raw data from all the channels sequentially. It shall scan the all the channels at a rate of 200 milliseconds.

The default mode of operation of this recorder is Normal Recording. Five samples each of all analog inputs shall be averaged at every second and shall be recorded as normal record i.e. History.

DR shall perform Disturbance Recording on demand, when triggered from ESR. ESR shall send an INITIATINGEVENT to DR to start the Disturbance Recording. A memory file for 8-minute data collection shall be maintained in DR. In each scan, at every 200 milliseconds, data of all 128 channels shall be copied in to this memory. When INITIATINGEVENT is received, data for the next 3 minutes (post-event data) shall be copied and at the end of third minute from the time of initiating event, 8-minute memory shall be copied on to a disk file with a file name referring to the time of INITIATINGEVENT. During the post event data collection, one more initiating shall also be processed.

DR shall send the normal record data of the 128 inputs to the Server over the Ethernet at every second, where in the History shall be maintained and facilities to view the data shall be provided. DR shall also send the Disturbance Record files to the Server for storing and viewing.

Performance Requirements

Normal Recording

- 5 samples shall be average at every second and recorded

Disturbance Recording

- Scan rate for Disturbance Recording shall 200 milliseconds
- Disturbance recording shall start with 200 milliseconds after detecting the Event

3 Server

RTDAS comprises of a Server, where in Graphical utilities for viewing the data shall be provided. All records such as 120 Hours Normal records (History), Disturbance Records, Event Records of 4000 events each, Fast Records and HRT Records shall be maintained in Hard disk.

Infrastructure requirements

- 512MB RAM
- Pentium-IV 3.2GHz PC
- 160GB HDD
- High-resolution 19” TFT LCD Monitor
- Keyboard and Mouse
- Network Interface boards

Functional Requirements

The following graphical utilities shall be provided on

Server: -

a. Graphical Trend Display
b. Tabular Trend Display
c. Bar Graph Display

d. ESR Inputs Display
e. Analog Inputs Display
f. Digital Outputs Display

Server shall maintain the database of all Analog Inputs, Digital Inputs, and Digital Outputs. The database shall include Point ID, Description, parameter process limits, Alarm settings, Contact type, time delays etc. They shall also maintain Groups for various displays, ESR Groups, etc. Server shall provide utilities to edit the parameters of any channel. Server shall provide the printing of various displays and data recorded on the Graphical Printers connected to them.

Server shall provide facility to transfer the data required/specified by the User on to a transferable media like floppy or USB disk.

4. Viewing Unit

RTDAS shall provide an Offline Viewing Unit to view and analyze the data transferred from Server.

Infrastructure requirements

The configuration of Viewing Unit shall be as similar to the Server.

Functional Requirements

The facilities provided on Viewing Unit are as follows:

a. Graphical History Display
b. Tabular History Display
c. Alarm Summary
d. ESR History

The data that shall be viewed or analyzed on this unit are:

a. One Hour data of the Normal Record (History)
b. Disturbance Record
c. ESR Event History

5. Interface Requirements Hardware Interfaces ESR On-line Printer

ESR shall be connected with a dedicated Dot matrix printer for on-line printing. ESR shall facilitate to start/stop the Event logging on to the printer. Event Logging shall be possible for all events or based on the group. Accordingly, the ESR shall provide the configuration utility. The printer shall be connected to the ESR on COM port and accordingly the print functions shall be designed. Appropriate printer drivers shall be installed and configured properly.

PCI to I/O Interface Board

PCI to I/O Interface board is used for interconnecting the PC to the I/O Bin. This provides an I/O Mapped I/O to the I/O Bin through its port configurations. The device driver for this shall be installed and configured. The Disturbance Recorder uses this board to interface the Analog Input bin.

PCI DIO Boards

PCI DIO boards provide Isolated 32 DI and 32 DO channels. These boards provide I/O Mapped I/O for 32 channels DI and DO. These boards are used for: -

-Driving All Annunciations

- CCS shall be connected to the PCI DIO cards as an Interrupt Input, such that it triggers an interrupt in the system to immediately synchronize the time
- INITIATINGEVENT communication

a. DisturbanceRecording INITIATINGEVENT from ESR to DR

b. Fast Recording INITIATINGEVENT From ESR to FR

Device drivers for these boards shall be installed and configured.

6. Software Interfaces Operating System

RTDAS shall use Linux Operating System in all the systems. The following Operating systems shall be used in various computers used in RTDAS.

- Embedded Linux in all I/O Systems.
- Red Hat Linux 9 in Server and Viewing Unit

Development Tools

- GNU GCC compiler, Makefile
- Xlib, Xmotif

Database

As the size of the database is small, Microsoft Access shall be sufficient for RTDAS. All the Database tables such as Analog Input Table, Digital Input Table, Digital Output Table, Group Tables and Color Tables etc. shall be created and updated as MS Access

Database files.

The database shall be accessed in to the software applications through ODBC protocol. Database shall be organized such that the tables are identifiable easily and it shall be possible to create the database using text files also when setting up the system. A proper DSN shall be configured which refers to the Plant Name, Unit No., and the System.

Libraries

- Glibc anduclibc

Communication Protocols

- UDP/IP

Inter Process Communication

- Shared memory concept shall be used on local system for sharing the data between different modules

- Socket programming shall be used for communication between various subsystems. The communication channels shall be DRÅÆServer and ESRÅÆServer.

SYSTEM DESIGN

The major sub-systems required to fulfill the requirements are:

1. I/O sub-system for scanning the field inputs and setting the digital/analog outputs.
2. Server for data acquisition, processing, controlling of network, centralized logging etc.
3. 3.Viewing Unit for operator interface
4. 4.Printers for printing alarms, and operator demanded reports.

1. Decomposition description of the system I/O Subsystem

The data acquired from the field is sent to ESR through DIIM's. Event Sequence Recorder scans 256 digital inputs at a scan rate of 1 millisecond. It compares the present status of all the inputs with their previous stored status and records the new status along with their time of occurrence. ESR constitutes an I/O controller, an I/O Bin with 8 Digital Input cards and Power Supplies. The I/O Controller is an Industrial Grade Computer, which interfaces I/O Bin and the LAN.

The system is designed and developed by ECIL for acquiring and processing inputs and generating outputs. The whole system is divided in to 4 subsystems:

- Event Sequence Recorder System
- Disturbance Recorder System
- Recording Unit (Server)
- Viewing Unit
- The basic functions performed by the I/O subsystems are as follows:oScanning the data (analog and digital) from the field.
- Transmission of raw data to Recording Unit.
- Recording the inputs sequentially with ESR.
- to record disturbances DR are triggered by ESR.
- Quality checking of signals.

Server (Recording Unit)

The Server receives data from I/O sub-systems (ESR, DR). The received data will be recorded and displayed through Viewing Unit.

The functions of the server are as follows:

- Control of network traffic.
- Regularly transmitting and receiving the health packets to/from other nodes over the network.
- Synchronizing to the central clock system and sending synchronizing signals to all other nodes at every minute.
- Receiving the various queries and processing these and sending back the required data to the nodes.

- Updating/maintaining dynamic database including history of data, alarm pages, alarm history etc., for 50 days.
- Processing of analog/digital/calculated variables.

Viewing Unit

These nodes contain total static data and dynamic data. The data gets updated every second through the network from server.

The functions of the Viewing unit are as follows:

- Maintain latest dynamic data, alarm pages, alarm history.
- Send health packets to server regularly.
- Receive time synchronization signals from server at every minute.
- Maintain display (temporary) groups generated by the operator.
- Based on the operator's request, it displays the current data taking from display node itself up to 24 hrs. and for history beyond 24 hrs., up to 30 days on day basis it sends request to server and gets relevant data from server for displaying.
- Various operational interactions will be carried out through the dialog boxes, which will be through any one of the display nodes.

Printers

Printers are used to print alarms, and operator demanded reports. All the printers in the System are network printers and are connected through print servers. But few of them are dedicated to Supplementary Control Room and head-office.

2. Software Architecture

Software Design for I/O System

Modules inthe I/O system are used to initiate thetask of scanning the parameters from the plant.

a) Driver for PCI I/O Interface
b) SCANTASK
c) IO LANCOMM

a) PCI I/O Interface Driver Design

The main purpose of this module is to made availability of the PCI I/O interface for the Application software to scan the data from plant through ECI/O system manufactured by Control and Automation division of ECIL.

The pci_driver structure

static structpci_drivereios_pci_driver =

{

Name:EIOS_DRIVER_NAME, probe : eios_pci_probe,

remove :eios_pci_remove,

```
id_table :eios_pci_table,
};
```

The pci_driver data structure is the core of hot-plug support. The structure is pretty small, being made of just a few methods and a device ID list.

char *name;

The name of the driver; it has informational value.

const struct pci_device_id *id_table;

An array that lists which devices are supported by this driver. static int __initeios_pci_init (void)

```
{
return   pci_register_driver (&eios_pci_driver);
  }
static void __exit eios_pci_exit (void)
{
 pci_unregister_driver (&eios_pci_driver);
}
module_init (eios_pci_init); module_exiteios_pci_exit);
MODULE_LICENSE ("GPL");
```

When driver is loaded into the kernel the execution starts at the function eios_pci_init. This function returns the no of devices detected for the driver. The user defined function eios_pci_probe is used to probe for the devices.

int pci_register_driver(struct pci_driver *drv); This function inserts the driver in a linked list that is maintained by the system. That's how compiled-in device drivers perform their initialization.

void pci_unregister_driver (struct pci_driver *drv); This function removes the driver from the linked list of known drivers. int register_chrdev (unsigned int major, const char *name, struct file_operations *fops); The return value indicates success or failure of the operation. A negative return code signals an error; a 0 or positive return code reports successful completion. The major argument is the major number being requested, name is the name of the device, which will appear in */proc/devices*, and fops is the pointer to an array of function pointers, used to invoke the driver's entry points.

The major number is a small integer that serves as the index into a static array of char drivers. Minor numbers, too, are eight-bit quantities. They aren't passed to *register_chr dev* because, as stated, they are only used by the driver itself.

b) Design of I/O software

The purpose of I/O software is to scan digital and analog field signals from plant and generate annunciations by writing data through digital outputs.

Event Sequence recorder

Create separate thread called scan threadto scan the digital data. Another thread called process thread for processing the raw data.

Steps to scan digital data from field.

1 Open the device file eiospci0 in /dev/ directory.
2 Configure the ports for read or write.
3 Enable the I/O bus through control port.
4 Reset buffers for digital output boards.
5 Read status register to check the slot address.
6 Signal read address data enable.
7 Read the status register for board type checking. 8. Signal read data enable
8 Read first 16 channels.
9 Select next 16 channels and read the data.
10 Compare present data with previous data. If any change detected generate event. Collect these events and generate record for every 4000 events.
11 Signal to process thread to start processing. In processing we can decide state of the particular input signal as either normal or alarm.
12 ESR evaluates a set of equations given by the user and initiate disturbance recording.

Disturbance Recorder

a. The main function of this module is the acquisition of data from 128 Analog Inputs, verifies the data and converts the raw data to the Engineering Values.

b. The default mode of operation of this recorder is Normal Recording. Five samples each analog input shall be averaged at every second and shall be recorded as normal record i.e. History.

c. Starts disturbance recording when receives initiating event from ESR.

Create a thread called scan thread to scan the analog data. Another thread called process thread for processing the raw data.

Steps to scan Analog input data from field.

1 Open the device file eiospci1 in /dev directory.
2 Configure the ports for read or write.
3 Enable the I/O bus through control port.
4 Read status register to check the slot address.
5 Signal read address data enable (RADEN).
6 Read the status register for board type checking.
7 Select the channel and latch the channel.
8 Signal write data enable (RDEN).
9 Set start conversion bit in control port.
10 Wait for End of Conversion bit to set.
11 Set Read data enable and read the data.

12 Signal to process thread to start processing. In processing we can decide state of the particular input signal as either normal or alarm.
13 When ESR initiates it starts disturbance recording of 3 minutes pre disturbance data and 5 minutes post disturbance data. The collected data is maintained as records.

c) IO LANCOMM

a. The main function of this module is to collect dynamic data from I/O subsystems by triggering an event at every one-second from server.

b. Through query thread I/O messages are read and processed which further updates the I/O common segment

Stepsin IOLANCOMM

a. Create UDP socket.

b. Bind the socket to one of the ports.

c. Wait for receive the request from server and Send the requested data to server

Software design for Server

Server is the coordinator of the subsystems in RTDAS. All subsystems are connected to server.

The modules in the server includes

- Server LANCOMM
- Database
- Shared memory
- Processing.
- Record generation.
- Graphical user Interfaces.
- Main application.

1. Server LANCOMM Module.

The purpose of the module is to send request to I/O subsystems for every second and get dynamic data and update shared memory.

Steps in Server LANCOMM

a. Create UDP socket.

b. Bind the socket to one of the ports.

c. Send request to I/O subsystems for every one second.

d. Receive data from I/O subsystems.

2. Database

Mysql is a Database server. A database is a data storage feature. It can be used to store, sort, arrange, and display information. MySQL is a functional feature on its own. MySQL database is used to store the static data for example process high, process low, description etc. The data stored is used by the shared memory for processing the data. The MySQL database server is the world's most popular open-source database.

MySQL is an attractive alternative to higher-cost, more complex database technology. Its award-winning speed, scalability and reliability make it the right choice for corporate IT departments, Web developers and packaged software vendors.

3. Shared Memory

Shared memory is the fastest form of IPC available. Once the memory is mapped into the address space of the processes that are sharing the memory region, no kernel involvement occurs in passing data between the processes. Shared memory contains static data about field signals taken from database and dynamic raw data filled by server LANCOMM process.

4. Processing

Raw data taken from field is processed for further analysis and to display on man machine interfaces. Shared memory is the fastest form of IPC available. Once the memory is mapped into the address space of the processes that are sharing the memory region, no kernel involvement occurs in passing data between the processes.

5. Recording

The purpose of the module is to record the field data for one day in a file called one record. For easy retrieval of data from the record we write data with time stamps according to offsets taken for each second.

6. Graphical User interfaces

The Graphical User Interfaces for Real time Data acquisition are developed on Xlib and Xmotif as development tools.

Bar Graph display, Tabular trend display and graphical trend display are developed using Xlib.

7. Main Application

The purpose the module is to start all the applications (IOLancomm, processing, recording) by linking all the user defined libraries (Database, Shared memory processing and recording) and providing displays.

Conclusion

The “Real Time Data Acquisition System” project was successfully developed and tested. We achieved the real time performance of scanning digital inputs in one millisecond. And we developed the project without using any third-party tools. In the I/O system we used embedded linux for data acquisition. On Server the Acquired data is processed and displayed using neatly built Graphical User Interfaces.

The “**REAL TIME DATA ACQUISITION**” project development has given exposure in working with the live environment following the Software Engineering Standards. This project development also given an opportunity to gain more technical knowledge, working with the Linux environment, X Windows System and writing device drivers. This work has given the confidence and incited interest to work more on this kind of projects. We have successfully implemented the user requirements and looking forward for working on the future enhancements.

References

1 Building Embedded Linux Systems by Karim Yaghmour.
2 Bar, Moshe. *Linux Internals*. McGraw-Hill. 2000. This terse book by *Byte* columnist Moshe Bar covers much of how the Linux kernel works, and includes a number of 2.4 features.
3 Linux Device Drivers by *Alessandro and Rubini*
4 Stevens, Richard. *Unix Network Programming*. P T R Prentice-Hall. 1990. Perhaps the definitive book on the Unix network programming API.
5 Xlib – C Language X Interface by James Gettys and Robert W. Sheflier.