DATA-LOGGING & SUPERVISORY CONTROL IN WIRELESS SENSOR NETWORKS

By

CHOUDHURY ADITYA NARAYAN DAS

Presented to the Faculty of the Graduate School of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements for the Degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

THE UNIVERSITY OF TEXAS AT ARLINGTON

December 2005

ACKNOWLEDGEMENTS

I would like to thank many people who have helped me make this thesis possible. First of all, I sincerely thank my supervising professor, Dr. Frank Lewis for his invaluable guidance and constant motivation. It is because of his deep involvement in my thesis work, I could make this thesis possible. I would like to thank my supervising committee consisting of Dr. Dan Popa and Dr. K. Yeung for their interest in my thesis work and for taking their time to be on the thesis committee. I am grateful to all the ARRI-ACS members and especially Ms. Sarah Densmore for their support and help.

Above all, I thank my mother Kalyani Das and my father Kshirod Chandra Das for believing in my abilities, constantly providing the support I needed and being the greatest teachers of my life.

Last but not the least; I also would like to thank Dr. Vincenzo Giordano for his much needed suggestions during my work tenure at Mobile Wireless Sensor Networks Lab at Automation and Robotics Research Institute (ARRI- UTA).

This work was supported by National Instruments Lead User Grant NSF GRANTS IIS-0326505 and CNS-0421282 ARO grant M-47928-CI RIP-05075-1.

November 16, 2005

ABSTRACT

DATA-LOGGING & SUPERVISORY CONTROL IN

WIRELESS SENSOR NETWORKS

Publication No. ______

Choudhury Aditya Narayan Das, M.S.

The University of Texas at Arlington, 2005

Supervising Professor: Dr. Frank Lewis

This thesis mainly emphasizes on the inclusion & implementation of the Supervisory control unit in the wireless sensor networks with a brief overlook on the cross-bow sensors and mobile sentry robots. The aim of this research has been to design an application module which would be able to manipulate the data effectively and support the Discrete Event Controller in generating the rule-based tasks by solving major issues like data-logging, alarm & event reporting and security concerns. The PC-based data logging systems provide the most flexibility, customization, and integration to the Wireless Sensor Network research project.

This work has been supported by National Instruments Lead User Grant NSF GRANTS IIS-0326505 and CNS-0421282 ARO grant M-47928-CI RIP-05075-1.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

CHAPTER 1

INTRODUCTION

1.1. Introduction

Famous Greek philosopher Aristotle once stated, “Man is a social animal”. Man lives in society exercising mutual interdependency and maintaining a sense of togetherness amongst fellow human beings. The most beautiful gift that man has been blessed with is the power to express its feelings through unique and distinct words. These distinct and meaningful words which are promulgated with an intention to be discerned into a representation of facts, concepts, or instructions in a formalized manner suitable for communication, interpretation, or processing by humans are termed as “Data”. Data is information, knowledge, and conceptions, related to people, or things, obtained by observation, investigation, interpretation, visualization, and mental creation. Data are intangible and include numbers, words, symbols, ideas, concepts, and oral verbalization. While in ancient times the word “Data” was mainly within a confined definition of conversations, books or any other primitive ways such as paintings or stone carvings; in the current era of times, where science has made an astounding breakthrough, the definition of “Data” has changed significantly. It would not be a high-flown statement to say that today’s world lives and runs on data and advanced communication technologies. The word “Data” has now been upgraded into a term called “Statistical Data” which is defined as “Factual information such as measurements and statistics, especially information organized for analysis or used to reason or make decisions". Starting from day to day applications such as cell-phones, televisions and World Wide Web (www) to more complex and advanced operations like satellite & spacecraft communications, nuclear reactor control systems or fully and automatic systems; there are statistical data spread over a broad spectrum. Without the modern computers and advanced digital signal processing methods, it would have been impossible to acquire and control such huge amount of statistical data. Therefore it will be appropriate to associate the term "binary" or "digital" to the definition of statistical data. In such contexts the word data implies a binary, machine-readable representation of information.

A major aspect of concern is the efficient and controlled management of this huge amount of data. While there is a vast and diverse range of data all around, some of low importance & some highly critical, some secure and confidential & some un-secure, some raw & some processed; it becomes imperative that there should be some supervisory control system which would manage these widely varied data efficiently. The main objective of the supervisory control system would be to acquire data, extract only the necessary blocks of information from it, process it and derive conclusion from it and store the data in some storage system for future reference. These ideas lead to the evolution of "Data-logging & Supervisory Control" system.

1.2. Objective

1.2.1. Overview

Data logging and recording is a very common measurement application. In its most basic form, data logging is the measurement and recording of physical or electrical parameters over a period of time. The data can be temperature, strain, displacement, flow, pressure, voltage, current, resistance, power, or any of a wide range of other parameters. Real-world data logging applications are typically more involved than just acquiring and recording signals, typically involving some combination of online analysis, offline analysis, display, report generation, and data sharing. Moreover, many data logging applications are beginning to require the acquisition and storage of other types of data.

Perception is a vital part of a smart environment, both as a way to sense the state of the space and a way to detect commands from its users. Without perception, ubiquitous computing would be cumbersome, and a smart environment would be impossible. With perception, an environment can come alive in its reactions to people and devices. All smart environments have some kind of sensing, be it cameras, microphones, light, temperature or pressure sensors in the floor, or other specialized sensors.

A wireless sensor network test bed consisting of several stationary cross bow sensors and two mobile sentry robots, has been set up in the Automation & Robotics Research Institute (ARRI) with an idea to build a smart environment. The environment is being constantly monitored by the unattended ground sensors which send the surrounding parameter data such as light, temperature, sound level, magnetism etc to a central control unit. The responsibility of the central control unit is to monitor the environment on the basis of the data sent by the sensors and control the environment with the help of the mobile sentry robots and the UGSs jointly by generating a set of rule-based tasks that been decided by a novel discrete event controller.

However there are some challenges in implementing the above concept in reality. Some of them are stated in the following section.

1.2.2. Real-life Challenges

The challenges faced in implementing the WSN with DEC are as follows:

- The UGSs send diverse data continuously to the central control unit. This data need to be processed, demodulated, identified and sorted-out before extracting any kind of substantial information from it.
- Once the data has been processed and information extracted from it, it might so happen that this data be required in future for reference and analysis. So the data must be stored in some kind of storage system.
- As the space in the storage system is limited, hence the data should be stored efficiently. Only the essential and significant part of the acquired data should be stored to use the entire space of the storage system for longer period of time. So there should be a filter system which would discern the acquired data and extract only the significant part of it for storage process.
- The stored data should be readily available and presented in such a manner so that decision can be made without much of a hassle. So there should be some system which would represent the stored data through some user friendly ways such as reports or trends or graphs.

- Now if the acquired data shows some kind of criticality which in turn represents some kind of instability in the environment then this situation need to be resolved immediately in order to maintain proper functionality. So there should be some kind of alarming system which would warn the DEC in the central control unit about the emergency and thus can initiate a sequence of tasks to resolve the situation and bring the system back to normalcy.
- Also the acquired and stored data should reach its proper destination quickly and efficiently. The World Wide Web (WWW) is the best way of communication these days. So there should be some way to publish the data over web to send it to the end user/uses.
- Lastly and most importantly data security is one major issue. As the wireless sensor networks handle large amount of raw data which could have been containing secure information of intrinsic and confidential value, hence this data must be protected from unauthorized access. Also the DEC must be protected from being operated by someone not having sufficient authorization. So there should be some security system which would set up a user authorization protocol and grant access privileges to handle the data, database or the discrete event controller itself.

In a nut shell, it can be said that there are various issues of concern in proper and secure implementation of the Wireless sensor network and the Discrete Event Controller. This thesis work puts light on these issues and discusses the methods of solution to the problems.

1.2.3. Contributions

This thesis work discusses the above mentioned real-life challenges in implementation of wireless sensor networks. Some of the contribution made by this work can be mentioned as follows:

- A proper PC based data acquisition system has been developed for efficient collection of data from the sensors.
- A data logging scheme and database has been developed.
- For intelligent storage of data in the database, a filtering method called "dead-band" has been incorporated.
- Report generation & historical data trending has been developed for data presentation.
- Alarming and event handling schemes are presented for dealing with emergency situations.
- A security system called "User Account Manager" has been designed to incorporate data safety by creating authorized user groups and granting access privileges.
- Web publication of current data & historical data and controlling the discrete event controller safely over the web has been developed and implemented.

This research work about the *data-logging and supervisory control (DSC) concept* has been done and implemented on the wireless sensor network test-bed at ARRI which uses an efficient matrix based discrete event controller for decision making.

This thesis includes the study and implementation of various data acquisition and data-logging techniques, historical and current data trending, issue-based alarm generation and acknowledgments and setting up of different levels of security. In addition to it, studies also made about the capability of the supervisory control unit to connect to PLC and industrial device networks, OPC servers & clients and a wide selection of other device servers. This thesis also introduces the web-based maintenance of the wireless sensor networks.

CHAPTER 2

WIRELESS SENSOR NETWORKS

2.1. Wireless Sensor Networks

A wireless sensor network, as the name suggests, is a network of sensory devices which communicate with each other and outer world with the aid of some wireless link. The main idea of building a wireless sensor network is to design a smart environment which provides an efficient unmanned monitoring system complemented with a decision making system which generates a set of tasks to regulate or manipulate the environment. The wireless sensor network test-bed (WSN) at Automation & Robotics Research Institute has been built with this envision. It comprises of a set of unattended ground sensors (UGS) and mobile sentry robots.

2.1.1. UGSs

UGS consist of a variety of sensor technologies that are packaged for deployment and perform the mission of remote target detection, location and/or recognition. Ideally, the UGS are small, low cost and robust, and are expected to last in the field for extended periods of time after deployment. They are capable of transmitting target information back to a remote operator. These devices could be used to perform various mission tasks including perimeter defense, border patrol and surveillance, target acquisition, and situation awareness.

The UGSs that are being used in the WSN test-bed are Berkley mode crossbow sensors.

Figure 1. Cross-bow MICA2 & MICA2DOT sensor family

Motes are new and advanced computing devices which are also conceptually known as smart dust. The basic idea of motes is not much hard to conceive. The core of a mote is a small, low-cost, low-power computer. The computer monitors one or more sensors. It is easy to imagine all sorts of sensors, including sensors for temperature, light, sound, position, acceleration, vibration, stress, weight, pressure, humidity, etc. Not all mote applications require sensors, but sensing applications are very common. The computer connects to the outside world with a radio link. The most common radio links allow a mote to transmit at a distance of something like 10 to 200 feet (3 to 61 meters). Power consumption, size and cost are the barriers to longer distances. Since a fundamental concept with motes is tiny size (and associated tiny cost), small and low-power radios are normal. Motes can either run off of batteries, or they can tap into the power grid in certain applications. As motes shrink in size and power consumption, it is possible to imagine solar power or even something exotic like vibration power to keep them running. All of these parts are packaged together in the smallest container possible. The biggest advantage of using motes is that they can be used in large numbers which communicate with each other and form an ad hoc network.

The crossbow sensors that are being used in the WSN test-bed are MICA & MICA2DOT which are third generation tiny wireless platforms for smart sensors; designed specifically for deeply embedded sensor networks.

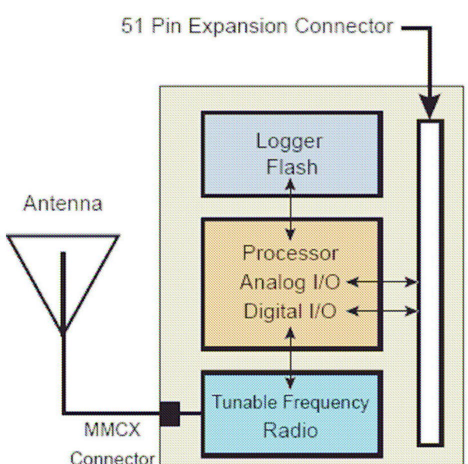


Figure 2. MICA2 Functional Block Diagram

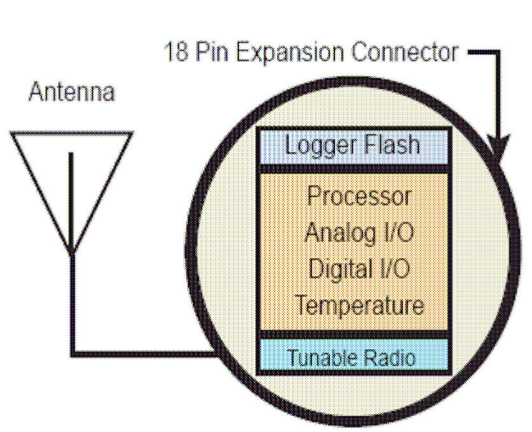


Figure 3. MICA2DOT Functional Block Diagram

Base Stations:

A base station allows the aggregation of sensor network data onto a PC or other computer platform. Any MICA2 Processor/Radio board (MPR4x0) can function as a base station when it is connected to a standard PC interface or gateway board, such as the MIB510CA serial interface board. The MIB510CA provides an RS-232 serial interface for both programming and data communications. Crossbow also offers a stand-alone gateway solution, the MIB600CA for TCP/IP-based Ethernet networks.

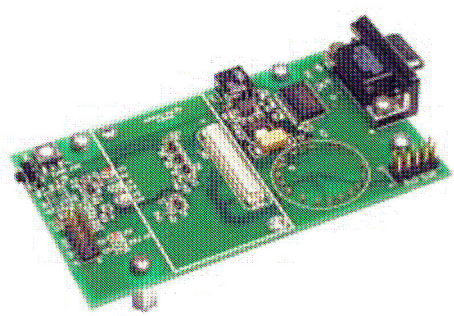


Figure 4. Cross-bow sensors base station

The MICA2 & MICA2DOT sensors operate on a small, open source, energy efficient software operating system called TINY-OS which was developed by UC Berkeley which supports large scale, self-configuring sensor networks.

2.1.2. Mobile Sentry Robots

The wireless sensor network test-bed also consist of mobile sentry robots manufactured by Cybermotion Inc. and donated by J.C. Penny. These robots are of first generation cyberguard SR2/ESP class robots which carries sensor technology by K2A+ mobile platform.

Figure 5. Cyberguard SR2/ESP mobile sentry robot

These robots include a high speed camera, a sensor package and an advanced subsystem comprising of ultrasonic intrusion detector, optimal flame detector, dual passive infrared, microwave intrusion radar, smoke sensor, gas sensor, temperature sensor, humidity sensor, ambient light sensor, video level measurement system, video transmitter, voice channel, auxiliary gas sensor, oxygen sensor, time lapse VTR, inventory tag reader, infrared illuminator, optical pyrometer, zoom lens. With its wide range of standard and optional features, the SR2/ESP can perform an enormous variety of missions from environmental monitoring to inventory control.

2.2. Discrete Event Controller

The decision making system of the wireless sensor network works on a novel discrete event controller. The discrete event controller (DEC) is a rule based matrix controller that has inner loops for non-shared resources and outer decision making loops for shared resources. As the controller is based on matrices; it provides efficient mathematical framework for discrete event analysis and design.

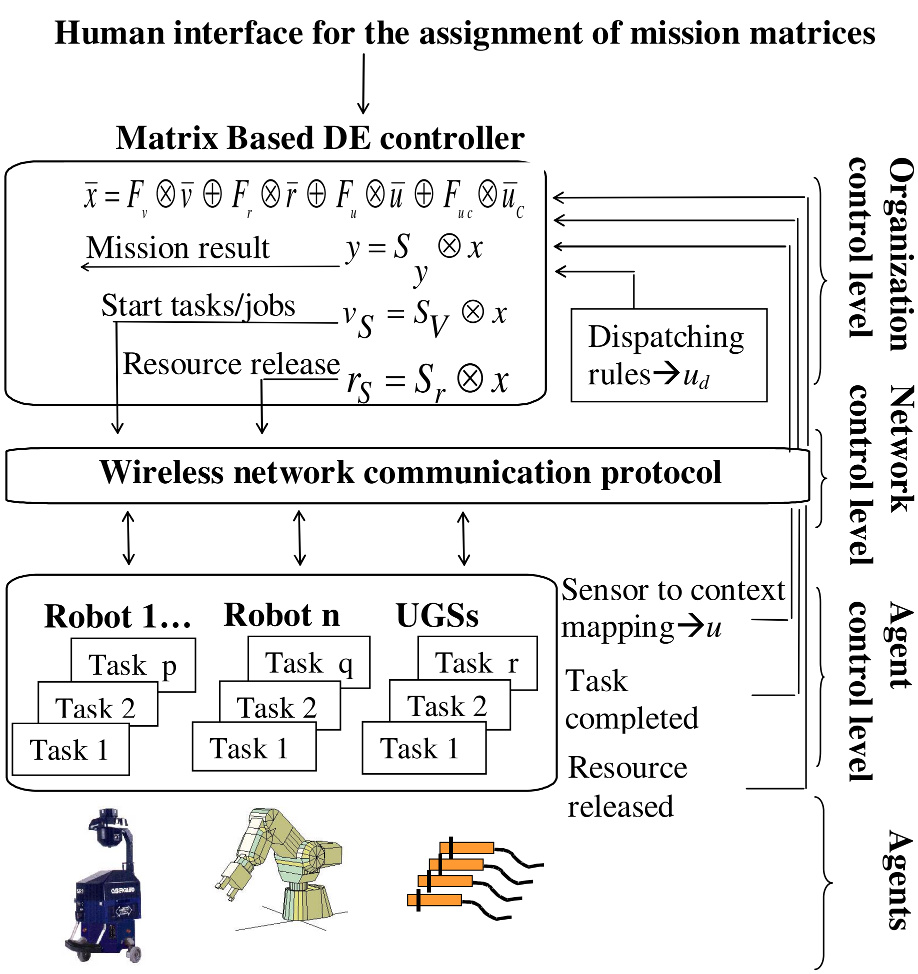


Figure 6. Human Interface for assignment of the mission matrices

Let v be the vector of tasks used in the system, r the vector of resources used in the system (agents), u the vector of inputs (occurrence of sensor detection events, node failures, etc.) and y the vector of outputs or results of the mission performed by the agents (e.g. false fire alarm, network coverage recovered etc.). The DE controller is illustrated in fig. 1 and described by the following equations:

$$\overline{x} = F_v \overline{v} + F_r \overline{r} + F_u \overline{u} + F_{ud} \overline{u}_d \ , \qquad \textit{matrix controller state}$$

$$v_s = S_v x \ , \qquad \textit{operation or job start}$$

$$r_s = S_r x \qquad \textit{resource release}$$

$$y = S_y x \qquad \textit{job completion}$$

where:

x is the task or state logical vector,

F_v is the task sequencing matrix,

F_r is the resource requirements matrix,

F_u is the input matrix,

F_{ud} is the conflict resolution matrix

U_d is the conflict resolution vector.

S_v is the task start matrix

S_r is the resource release matrix

S_u is the output matrix

A complex system can be sub-divide into rule based tasks for job sequencing, resource allocation and conflict resolution using the matrix based discrete event controller (DEC). The complete system architecture would be as follows:

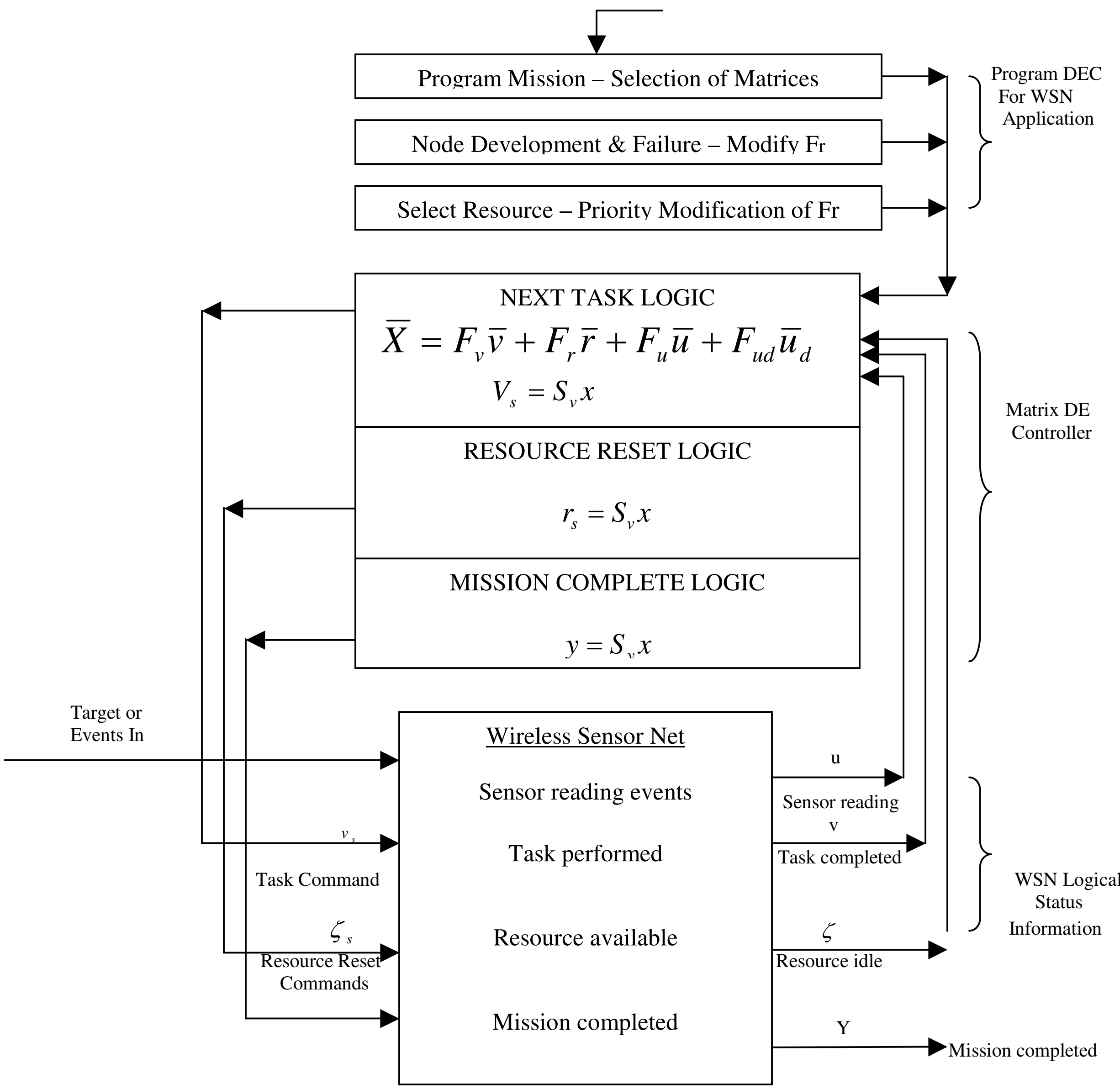


Figure 7. Discrete Event Controller Architecture

2.3. Machine Planning

Machine planning is a field of artificial intelligence. Basically through machine planning we aim to design an agent that does not require a human to provide step by step direction of the agent's actions. Instead, the agent uses its own knowledge about a particular domain to resolve the problems it encounters. It can also be referred as an intelligent control system which can adapt to the environment. Intelligent control systems are typically able to perform one or more of the following functions to achieve autonomous behavior:

- planning actions at different levels of detail
- emulation of human expert behavior
- learning from past experiences
- integrating sensor information
- identifying changes that threaten the system behavior such as failures, and reacting appropriately

This identifies the areas of Planning and Expert Systems, Fuzzy Systems, Neural Networks, Machine Learning, Multi-sensor Integration, Failure Diagnosis, and Reconfigurable Control, to mention but a few, as existing research areas that are related and important to Intelligent Control. While these techniques provide several key approaches to Intelligent Control, for complex systems they are often interconnected to operate within an architecture which is hierarchical and often distributed.

CHAPTER 3

DATA-LOGGING

3.1. Introduction to data-logging

"It is a capital mistake to theorize before one has data. Insensibly one begins to twist facts to suit theories, instead of theories to suit facts".

[Sir Arthur Conan Doyle (1859 - 1930)]

In its most basic form, data logging is the measurement and recording of parameters over a period of time. The parameter can be anything such as; temperature, light, pressure, voltage, current, resistance, power or any of a wide range of other parameter. But what ever it may be, it is called 'data' as it contains information. And data-logging is nothing but storing this information for future usages and reference. For example: weather profile of a region is monitored and stored over a long period of time. Later this information is used to predict the future weather in that region. Similarly ocean temperature and water salinity is stored and analyzed to predict the flow of ocean currents.

Real-world data logging applications are typically more involved than just acquiring and recording signals, typically involving some combination of online analysis, offline analysis, display, report generation, and data sharing. Moreover, many data logging applications are beginning to require the acquisition and storage of other types of data.

With the advancement of modern computers, new PC based data-logging systems has been evolved. These systems combine the acquisition and storage capabilities of stand-alone data-loggers with the archiving, analysis, reporting, and display capabilities of modern PCs. Today, PC-based logging systems provide the widest range of measurement types, analysis capabilities, and reporting tools.

3.2 Functional Structure of Data-logging

A PC based data-logging system mainly comprises of five main units. The functional block diagram is given as follows:

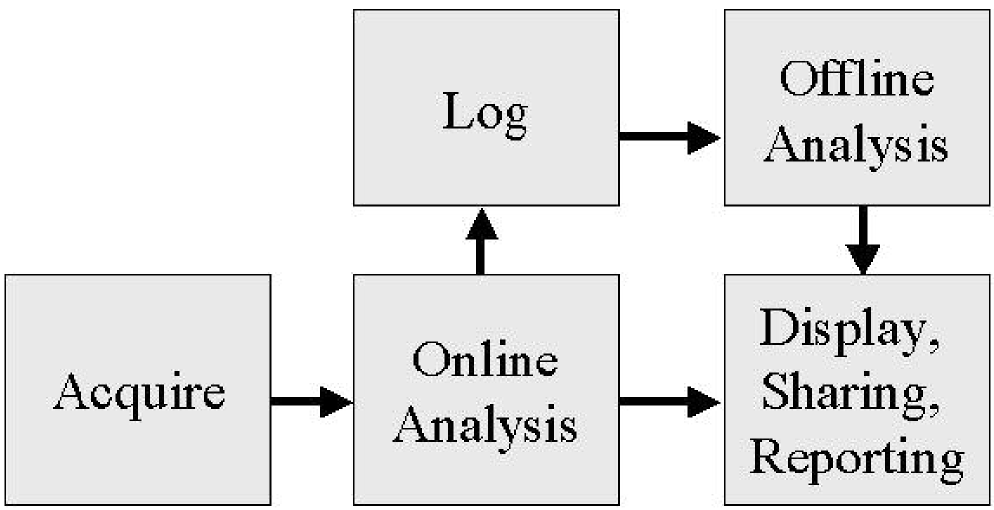


Figure 8. Functional Block Diagram of Data-Logging

The five major blocks of PC based data logging system are,

- Data Acquisition
- Online Analysis
- Logging & storage
- Offline Analysis
- Display, Sharing & Report generation

3.2.1. Acquisition

In a PC based system, data acquisition is accomplished by the measurement hardware which can be,

- sensors
- signal connectivity
- signal conditioning
- analog to digital converters

Sensors

Sensors are devices which converts physical parameters into electrical signals.

Signal Connectivity

Signal connectivity describes the component of the measurement hardware with which sensors are connected to the logging system.

Signal Conditioning

Signal conditioning is defined as the process of passing the signal through some kind of preparation before being digitized. There are many types of conditioning such as;

- Amplification and Attenuation
- Sampling
- Multiplexing
- Filtering
- Linearization

Analog to Digital Conversion

After the sensor converts the physical parameter into electrical signals, it is required to convert this analog signal into digital values which can be processed by a computer.

3.2.2. Online Analysis

In PC based systems special software are used to accomplish online analysis. A few schemes of online analysis can be stated as follows:

Channel Scaling

Channel scaling is defined as the conversion of raw binary values captured by acquisition unit into properly scaled measurements with appropriate measurement units.

Alarming & Event Management

Alarming and event management means monitoring a channel and providing notification if limits are exceeded. Alarming can also include an automated response to certain events.

Feedback Control System

Feedback generally means taking the output and feeding it back with the input. Thus the actual value is compared with the desired value and by minimizing the error the system can be approached to desired state.

3.2.3. Logging & Storage

The logging or storage block in essential in every data-logging system. Software is of critical importance in PC-based data logging systems, because well-written logging software determines how data is stored, how quickly data can be written to disk, and how efficiently disk space is used. Logging software also gives you data management capabilities, such as changing data formats, archiving data, and access to databases. There are three general formats commonly used for storage in data logging systems –

- ASCII text files
- binary files
- databases

ASCII text files

ASCII text files are the most common and flexible form of data storage. Text files for data logging applications are typically made up of a header section and columns of data. The header section gives information such as channel names, engineering units, test equipment, and user comments. The first data column is usually the timestamp of each sample, and it is followed by another column for each channel being logged.

Advantages:

i. Easy to open and import into any software package
ii. Easily transferable between operating systems

Disadvantages:

i. Inefficient Usage of disk space

ii. Need of additional process units for read and write to file

iii. Can be used for only slow data acquisition models

iv. Cannot be used when the data size is large

Binary Files

Binary files are the most efficient method of data storage. With binary files, the raw bytes that the computer is using to store data in memory are written directly to the file. This data takes up considerably less space than the same information written in ASCII text format, and it requires much less processor overhead than formatting into text.

Advantages:

i. Less processor overhead

ii. Less processor space

iii. Appropriate for faster data acquisition models

iv. Improved stream to disk speed

Disadvantages:

i. The binary files must be translated into another format before the can be shared between different applications

Databases

Databases are typically binary files that provide a structured format for inserting and retrieving data. They are optimized for efficiently handling large amounts of data and for searching through information in the database without loading everything into memory. Databases also are often designed for easy backup and archiving of data and multiple-user access. They usually have software methods to make it easy to import data into different software packages for analysis and report generation. In many ways, databases are the ideal storage format for PC-based data logging systems.

Advantages:

i. Capable of handling large amount of data
ii. More structured and efficient way of storing data
iii. Optimized data management

Disadvantages:

i. Increased complexity
ii. Difficult to implement if starting from scratch

Many different storage media types are used for data logging. Stand-alone data-loggers can use onboard nonvolatile memory, floppy disks, PCMCIA memory cards, tapes, or a variety of other options. PC-based data logging systems usually rely on the internal PC hard drive, which is now possible because of the trend toward more reliable and higher capacity hard drives. The 40 GB (and larger) hard drives that are readily available today make hard drives one of the most economical storage devices.

3.2.4. Offline Analysis

Offline analysis is performing mathematical functions on data after it has been acquired in order to extract important information. Types of offline analysis can include computing basic statistics of measured parameters, as well as more advanced functions such as the frequency content of signals and order analysis. Offline analysis can be integrated with the rest of the data logging application, or it can occur separately through stand-alone analysis software packages. Often, offline analysis is combined with the report generation, historical display, and data sharing functions.

3.2.5. Display, Sharing & Report Generation

The data-logging application requires a display to view the measurements that are being recorded. The display can be of two types,

- Historical Data- Data that has been previously acquired
- Live Data- Data that are currently being acquired

Data viewing utility should provide a user interface with general customization features.

Also the data-logging application requires some capability for reporting the data. Report generation can be integrated into PC-based data logging applications for increased efficiency. The logging application can be set up to periodically generate specified reports and distribute them.

For data that has been logged to be useful, it must be available to the right people. Logging application should have the capability to publish data over a network in order to propagate the information to the concerned people.

In widely distributed data logging applications, each logging node can publish its measurements to the network, and a main computer can serve as the central collection facility. The central computer retrieves the measurements from each node, combines them for further analysis, logs the results for permanent archiving, and periodically generates reports analyzing the data.

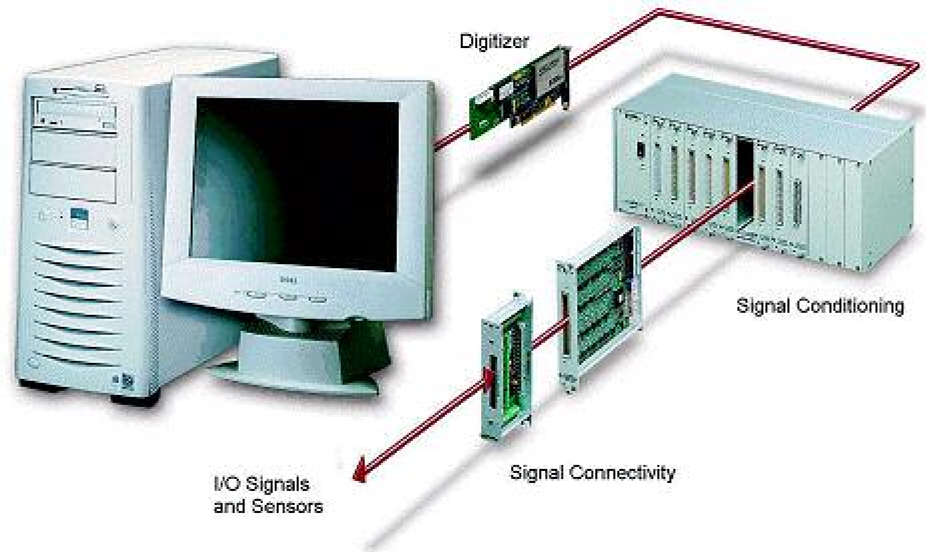


Figure 9. PC Based data acquisition

3.3. Dead Band

A *dead-band* is a filter that eliminates noise from data. Any changes in value from a data point are compared to the previous value. Only if the difference between the new value and the previous value exceeds the dead-band does the new value replace the old.

Dead-band allows defining what constitutes a significant change. The DSC ignores an operation if the change in data is not considered significant. By increasing the dead-band size, you can reduce the strain on the DSC, though this might compromise data resolution.

There are three types of dead-band by definition. They are:

- Update Dead-Band
- Log Dead-Band
- Alarm Dead-Band

3.3.1. Update Dead Band

By setting an update dead-band, any new value acquired by the DSC Engine is compared to the existing value. The new value replaces the existing value only when the difference between the new value and the existing value exceeds the update dead-band.

For example, for a data point with a range of values of 0 to 100, set the update dead-band to 1%. The existing value in the Tag Engine is 12.3. If the Tag Engine reports a new value of 13, the Tag Engine does not update because the change in value did not exceed the low dead-band. If the Tag Engine reports a new value of 11, it updates because the difference is greater than the low dead-band.

3.3.2. Log Dead Band

By setting a log dead-band value, the new value is compared to the old value. The new value is logged if it exceeds the log dead-band value.

For example, for a data point with a range of values of 0 to 100, set the log dead-band value to 2%. The last value logged was 12.3. When the Tag Engine updates to 11, the updated value is not logged because it is smaller than the 2% dead-band. The value in the Tag Engine must be greater than 14.3 or less than 10.3 for the data point to be logged.

3.3.3. Alarm Dead Band

By setting an alarm dead-band value, the new value is compared to the old value. The alarm is triggered when the value falls outside the range of the dead-band and is cleared when the alarm value reaches the inside range of the dead-band.

For example, for a data point with a range of values of 0 to 100, set a LO condition alarm at a value of 12 with a dead-band of 1.5%. The alarm condition is not triggered until the Tag Engine value drops to 12 or below. The alarm stays active until the Tag Engine value rises to 13.5 or greater.

3.4. Data-logging in wireless sensor networks

The wireless sensor network comprises of a set of sensors which are distributed on cross-bow motes and mobile robots. These sensors read the environment parameters such as temperature, light, sound, vibration, magnetism etc. and send them to the central supervisory control unit. So there arises a need for efficient management of this continuous flow of data. The data management scheme of the wireless sensor network has to take care of the following issues:

- The incoming data from the sensors need to be stored for reference. (Data-Logging)
- Due to the constraint of limited storage space only the essential and significant data need to be extracted from the raw incoming data. (Dead-Band)
- As different sensors send electrical value for different parameters, these electrical signals need to be properly sorted and attached with corresponding measurement units. (Channel Scaling)
- Also as the cross-bow mote has five different sensors but it use only one transmitting unit, hence the incoming data readings need to be properly time-spaced and sampled. (Sampling)
- Also the data acquisition unit uses technique to ensure accuracy of data. (Averaging)

CHAPTER 4

SUPERVISORY CONTROL

4.1. Introduction to supervisory control

"All science is concerned with the relationship of cause and effect. Each scientific discovery increases man's ability to predict the consequences of his actions and thus his ability to control future events". *[Lawrence J. Peters]*

Basically a control system is defined as a system for controlling the operation of another system. A control system is always needed because some other systems which are uncontrollable in themselves need to be controlled. A supervisory control system is a system that constantly keeps watch on the process going on and handles the situation according to its importance. When designing a system, it is important to first understand what needs to be accomplished. Moreover if the system is to deal with data, then it becomes highly essential to manage the data flow efficiently and intelligently. The basic features that a supervisory control system associated with a sensory data acquisition system, has to take care of are;

1. Continuously monitor the flow of raw data
2. Extract only important or significant data from the raw data
3. Store the extracted data and export it when necessary
4. In the process identify critical data and raise alarms if necessary
5. Protect the data from unauthorized access

4.2. Characteristics of supervisory control

A supervisory control unit for a data management has the following characteristics;

- Connect
- Monitor
- Process
- Store
- Trend
- Threat Identify
- Protect
- Export

4.2.1. Connecting to the data source

The data may be generated from a number of different sources which may be operating individually and isolated from each other. The parameters that the sensors read may be completely different and the method of data acquisition also might be different. So the supervisory control system must have the capability to connect to the varied sources of data.

4.2.2. Monitoring the data flow

The next point after connecting to the data source, is to monitor the acquire data. As one cannot predict the nature of incoming data that depends on the environment that keeps on varying with time, it becomes necessary to monitor the data continuously or at some regular intervals.

4.2.3. Processing the raw data

The data acquisition unit keeps on flooding the supervisory control system with raw data continuously. Some of them may be of low significance and some of high importance. Some data may be critical and require urgent attention. It is the responsibility of the supervisory control system to discern in between there widely varied types of data. The supervisory control system also has to take care of constraints such as limited storage space, restricted channel width etc. Because of the storage space for the data is limited hence all the raw data cannot be stored as it would over crowd the storage space with unwanted information. Also it would not be possible to log the data over a longer period of time. So the supervisory control system has to devise some method for efficient processing and storing of data. Only the significant and important portion of the incoming raw data should be logged. A filtering method called "Dead-Band" has been described in the previous chapter for this purpose. Also as there are only a limited number of transmitting channels, it might so happen that raw data from different sources may have been transmitted over the same transmitting channel with time-spacing. So the supervisory control system should also be equipped with some demodulating technique to extract the data. Also the supervisory control system should take care of the accuracy of data acquisition.

4.2.4. Storing the data

The selected data need to be stored in a database for future reference. The supervisory control system handles the storing of data in the database.

4.2.5. Trending the data

The stored data is used to analysis the trends. The supervisory control system should have the ability to prepare and display the data trends. The trend could either be historical or current. One such trend pattern could be as follows:

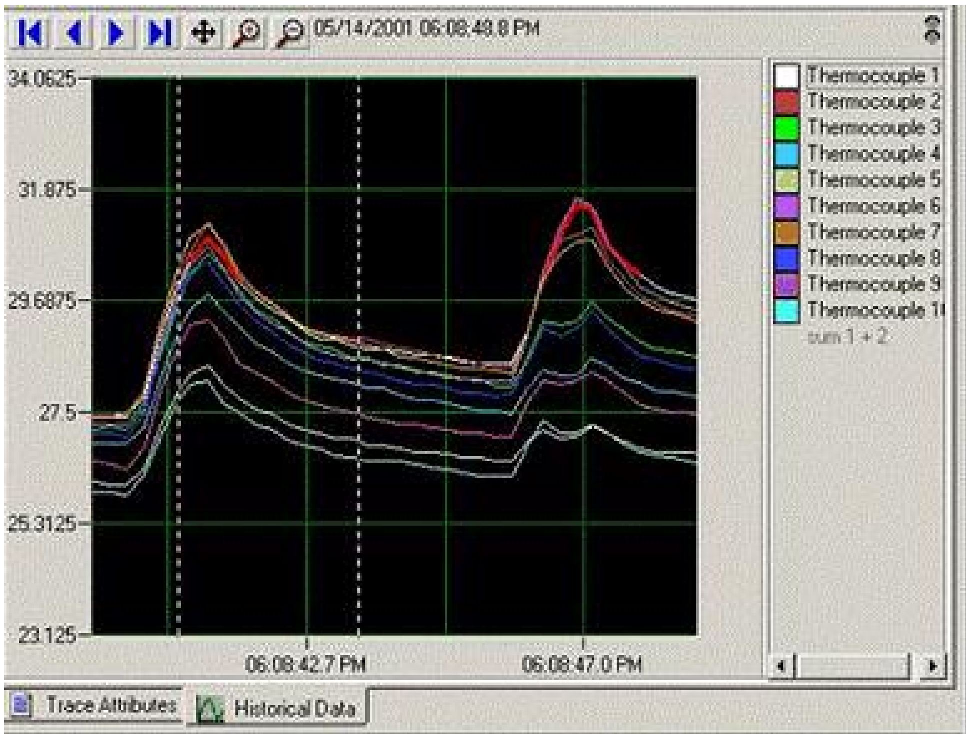


Figure 10. A Data trend Example

Trends are useful in deciding the data flow pattern and predicting the future nature of the environment.

4.2.6. Identifying Critical data

One major responsibility of the supervisory control unit is to identify critical data from the raw data. The situation could be various, such as sudden change in environment parameters, natural hazards and threat attacks or also some user defined situations when the system demands to be attended immediately. Under such circumstances the supervisory control system must raise some alarm to inform the emergency situation. The supervisory control system should also disable the alarming system once it has been acknowledged.

4.2.7. Protecting the data

The data might contain information which can be of secure type or confidential. This confidential information must be protected from unauthorized access. So the supervisory control system should be able to set up a rule for access privilege to data. Setting up different levels of security, the supervisory control system can categorize the access privilege to users; such as read only, write only, read & write, read write & modify etc.

4.2.8. Exporting the Data

Data received from the source would be of no use if it does not reach its proper destination. In today's networked environment, the ability to connect data application to the Web can be very important. The application development utility should provide tools to make publishing results to the network a trouble-free process.

4.3. Supervisory control in wireless sensor networks

The wireless sensor network test-bed has ample scope and need for supervisory control. The sensor data management process involves almost all the features of supervisory control system; such as connecting to various data sources, monitoring and processing the raw data, storing the extracted data, alarm & event management, trending and report generation, data protection, data exportation etc.

The wireless sensor network implements a discrete event controller (DEC) to generate a rule-based task system to automate the smart environment. The supervisory control unit (DSC) plays a vital role in congruence with the DEC to accomplish this goal. The DSC scans the raw data and prepares a useful set of data and stores it in a database. This data then serves the purpose of informing the environment information to the DEC from which the DEC determines the state and resource requirement. When ever a critical situation arises the supervisory control system (DSC) raises an alarm indicating the DEC that there has been an event occurring which needs to be attended. The DEC then, based on the data from the affected sensors as well as other sensors and mobile robots; prepares a series of tasks that need to be carried out in order to deal with the situation. Once the mission has been accomplished successfully the DSC disables the alarming system. Also the event is logged for future reference. The DSC also adds networking feature to the DEC which enhances the capability of the controller. With the networking capability the DEC now can run in a network controlling the smart environment. As the data and task instructions can be published over the web by the DSC, the DEC can now be safely run from a more secure and sophisticated location with more advanced facilities. For instance; if the wireless sensor network is deployed in a dessert or in a ocean or in a war field or in any other such in-convenient places, then the DSC can enhance the performance and security of DEC by isolating it from the hazardous environment and allowing it to run from a safe location such as control room or clean room; thus by saving appreciable amount of money and reducing the risk factor. Last but not the least; data security is one major issue of concern. In no case one can leave the control of the DEC in the hands of unauthorized user. So it becomes highly imperative that the DEC operation be controlled and data be accessed only by authorized users. Here the DSC plays a vital role in setting up security measures. The DSC set different levels of security such as administrator, operator, guest etc. The DSC also categorize the access privilege such as data read only, data write only, data modify only, data read write & modify etc.

There the Data-logging & Supervisory Control (DSC) enhances the performance of the Discrete Event Controller (DEC) and the wireless sensor networks.

CHAPTER 5

IMPLEMENTATION

Contributions of this thesis work were implemented on the wireless sensor network test-bed at ARRI which consist of mobile sentry robots and Berkley cross-bow unattended ground sensors. The following component of the WSN test-bed were developed & implemented;

- ✓ A PC based data acquisition system
- ✓ A data logging system
- ✓ A filtering method called "dead-band"
- ✓ Report generation & historical data trending
- ✓ Alarming and event handling
- ✓ A security system
- ✓ Web publication

The major software of wireless sensor network test-bed was built using LabVIEW® and MatLab.

5.1. NI-LabVIEW®

National Instrument's LabVIEW® is a graphical development environment for creating flexible and scalable test, measurement, and control applications rapidly and at minimal cost. With LabVIEW®, engineers and scientists interface with real-world signals, analyze data for meaningful information, and share results and applications. Regardless of experience, LabVIEW® makes development fast and easy for all users.

5.1.1. Key features of LabVIEW®

- Graphical Programming
- Built-in measurement and control function
- Multiple development tools
- Wide array of computing targets

The main programming section of LabVIEW is a VI or Virtual Interface and a corresponding block diagram. Programming for the VI is done using control palette which contains several controls and indicators. Similarly the corresponding block diagram is programmed using the function palette.

5.2. Data-logging and Supervisory Control Unit

The LabVIEW Data-logging and Supervisory Control (DSC) Module adds features and capabilities to LabVIEW to help creating monitoring applications and data logging applications. The DSC Module provides solutions for supervisory control of a wide variety of distributed systems using the flexibility of graphical LabVIEW programming.

The features are:

- Configuration utilities and wizards
- Historical data collection and trending
- Alarm and event reporting and logging
- Security
- Connection to PLC and industrial device networks
- OPC server and client
- Connection to a wide selection of device servers

5.3. Citadel5® Database

The LabVIEW Data-logging and Supervisory Control (DSC) Module uses the National Instruments Citadel historical database. The DSC Module also includes the Citadel ODBC driver that has special commands to perform data transforms, so you can retrieve, manipulate, and analyze historical data automatically from outside the LabVIEW environment.

Data that has been configured to be logged to a Citadel database resides in a set of files in the target directory set by the user for logging. This data can include values from the application as well as alarms and events. It is under the user's control to select which data is logged to what location through tag configuration and alarm and event configuration. Data can be logged to the local computer or to a remote computer on the network, provided the directory to which the data is being logged must be writable from the computer running the Tag Engine.

Access to Citadel data can be accomplished through the Historical Data Viewer, Historical Data VIs, SQL queries, or any other ODBC-compliant application such as Microsoft Query, Microsoft Access, or even Microsoft Excel.

5.4. Functional Architecture of DSC Module

A DSC Module application contains three parts that work together—the graphical interface or HMI application, the Tag Engine, and various device servers. The Tag Engine, with any device servers, runs as a separate application independent of the HMI application.

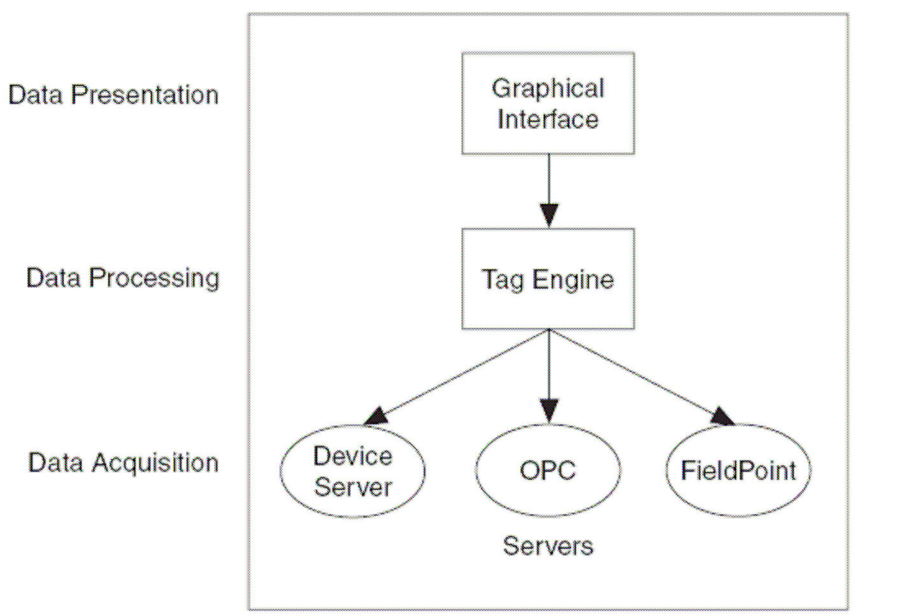


Figure 11. Data-logging & supervisory control module architecture

The data-logging & supervisory control module comprises of three major units. They are:

1. Server
2. Tag
3. Database

5.4.1. Server

In the LabVIEW Data-logging and Supervisory Control (DSC) Module, a *device server* is an application that communicates with and manages input/output devices such as PLCs, remote input/output devices, remote Tag Engines, and Data Acquisition (DAQ) plug-in devices. These servers read selected input items and write to them on demand. The DSC Module can connect to any OPC-compliant server and many third-party device servers and also to VI-based servers.

A server item is a channel, input/output point, or variable in a hardware device. DSC Module applications are connected to these server items with tags. Device servers monitor the values acquired by the hardware and the Tag Engine updates the tags when the server sends new data to the Tag Engine. Servers also update each output when the Human Machine Interface (HMI) application writes that tag value and handle and report communications and device errors. A good device server covers all device- and hardware-specific details, establishing a device-independent input/output layer for the DSC Module. Many device servers include a configuration utility as well as the run-time application that communicates with the Tag engine.

The DSC module supports various types of servers such as OPC servers, DDE servers and VI based servers.

Servers need to be registered and then launched for operation of DSC applications.

5.4.2. Tag

In the LabVIEW Data-logging and Supervisory Control (DSC) Module, a tag is used to create and maintain a connection to a real-world input/output point.

A memory tag can be used for data held by an application that is needed to use or track.

A network tag is remotely connected to any type of tag on another Tag Engine.

The tasks performed by tags depend on how the tag attributes are configured. Tag attributes include how tag data is scaled, if and how a tag is logged to a historical database, and alarm levels and priorities for tag data.

By configuring tag attributes, the following tasks can be accomplished;

- Organize tags into logical groups for convenience and efficiency
- Configure the tag data type
- Set initialization values
- Set separate dead-bands for logging or updating data
- Attach units of measurement to data
- Attach alarm message to a tag whose values enter the set alarm ranges
- Set alarm dead-bands separate from the logging & update dead-bands

After the creation tags and configuration of their attributes, the information is saved in a configuration (.scf) file. Any DSC Module utility that needs tag information uses the .scf file. These utilities include the Tag Engine, Tag Monitor, and HMI Wizard, which generally access the .scf file to find a list of active tags and other configuration information.

The .scf file does not contain any information about the VIs in the HMI application and does not need to be specific to any single application. Multiple applications can run concurrently using the same .scf file.

5.4.3. Database

The LabVIEW Data-logging and Supervisory Control (DSC) Module uses the National Instruments Citadel historical database. The DSC Module also includes the Citadel ODBC driver that has special commands to perform data transforms, so the user can retrieve, manipulate, and analyze historical data automatically from outside the LabVIEW environment.

Data that has been configured to be logged to a Citadel database resides in a set of files in the target directory you set for logging. This data can include values from the application as well as alarms and events. The user can control which data is logged to what location through tag configuration and alarm and event configuration. The user can also log data to the local computer or to a remote computer on the network, but the desired log directory must be writable from the computer running the Tag Engine.

In addition to these major and basic blocks, the data-logging and supervisory control unit also has some other important block of significance. They are;

- Human Machine Interface (HMI)
- Alarms and Events
- Security

5.4.4. Human Machine Interface (HMI)

A Human Machine Interface (HMI) is the interface through which a user interacts with the LabVIEW system and with the outside environment that LabVIEW monitors and controls. In LabVIEW, VIs are used as the HMI for an application. The LabVIEW Data-logging and Supervisory Control (DSC) Module installs custom controls, indicators, VIs and functions, and wizards to make HMI creation easier.

5.4.5. Alarms and Events

An *event* is something that happens within the DSC Module application. Events can be divided into two groups: *tag events* that pertain to individual tags, and *system events* that pertain to the overall DSC Module system. An example of a tag event is a change of alarm state for a tag. Examples of system events include a user logging on, the Tag Engine starting up, or historical logging being turned on.

In the DSC Module, an *alarm* is a specific kind of event related to the value of a tag. An event can be virtually any instantaneous activity such as clicking a mouse button.

An alarm typically has the following characteristics:

• Denotes an abnormal condition

• Occurs under certain, specific conditions

• Must be acknowledged by the user or configured for automatic acknowledgment

5.4.6. Security

User and group accounts are set to implement security in an application. Then access is configured and restricted to the application-specific Human Machine Interface (HMI). A system with permission-based security is a system in which users are allowed various degrees of access to tools or data depending on the permission attached to them.

5.5. Implementation of DSC to WSN

The data-logging and supervisory control unit of the wireless sensor network uses a VI-based server as communication channel. The operation of a VI-based server is pretty simple. The server supplies data points from several input items to the tag engine as these points are read. The tag engine also can send the values for output items. The server automatically timestamps values as they are acquired from items. A Typical VI-based server registration program is shown below,

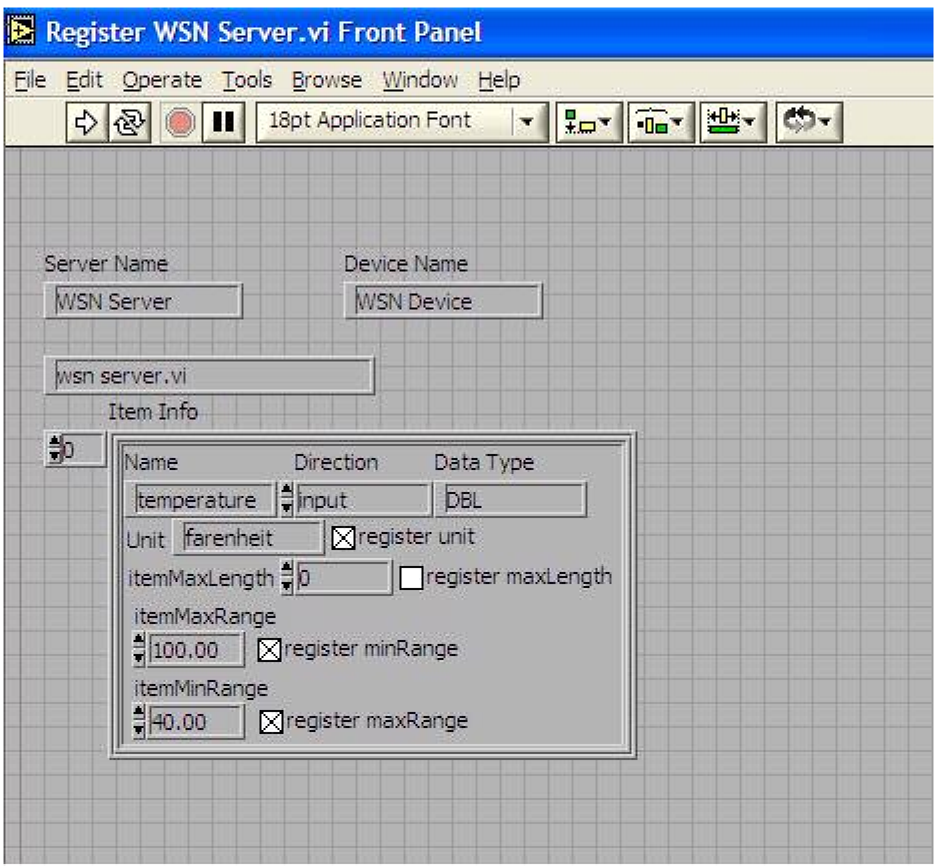


Figure 12. Front Panel of Registering theWSN Server

The corresponding block diagram for the server registration process is shown below,

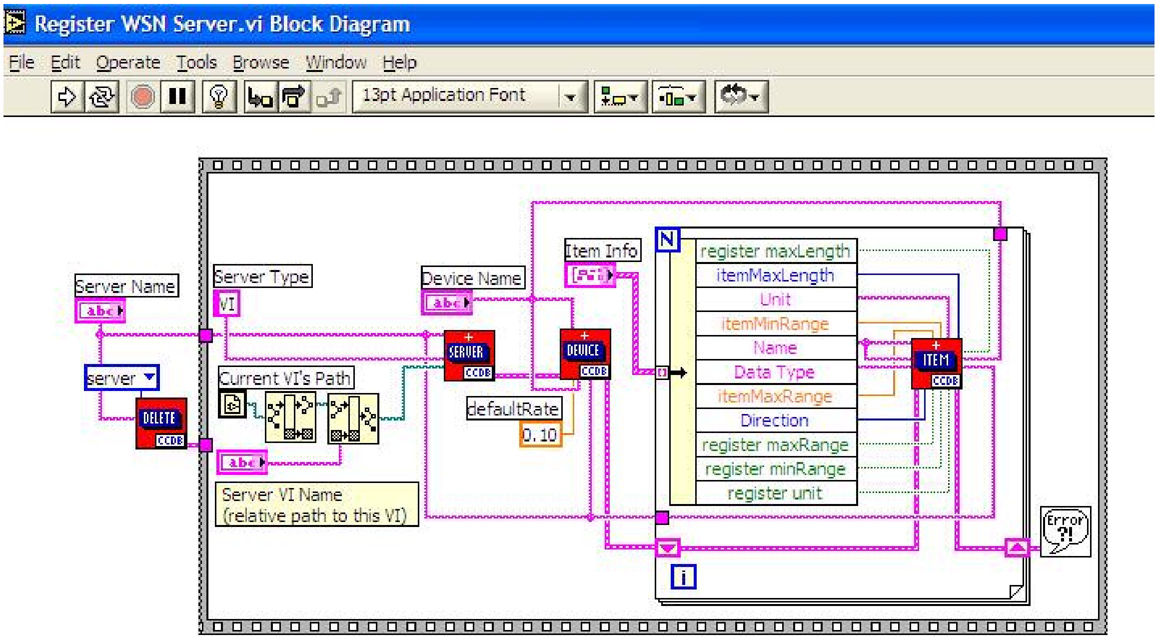


Figure 13. Block Diagram of Registering the WSN Server

After the creation and registration of the VI-based server, a tag has been created to maintain connection with the I/O points. LabVIEW DSC module has tag configuration wizard which easily creates tag to connect to the I/O points and add items. A tag can be created in several ways,

- by the tag configuration wizard
- manually in the tag configuration editor
- in the HMI wizard

We have used the tag configuration editor to create the required tag manually. The brief configuration process is shown as follows,

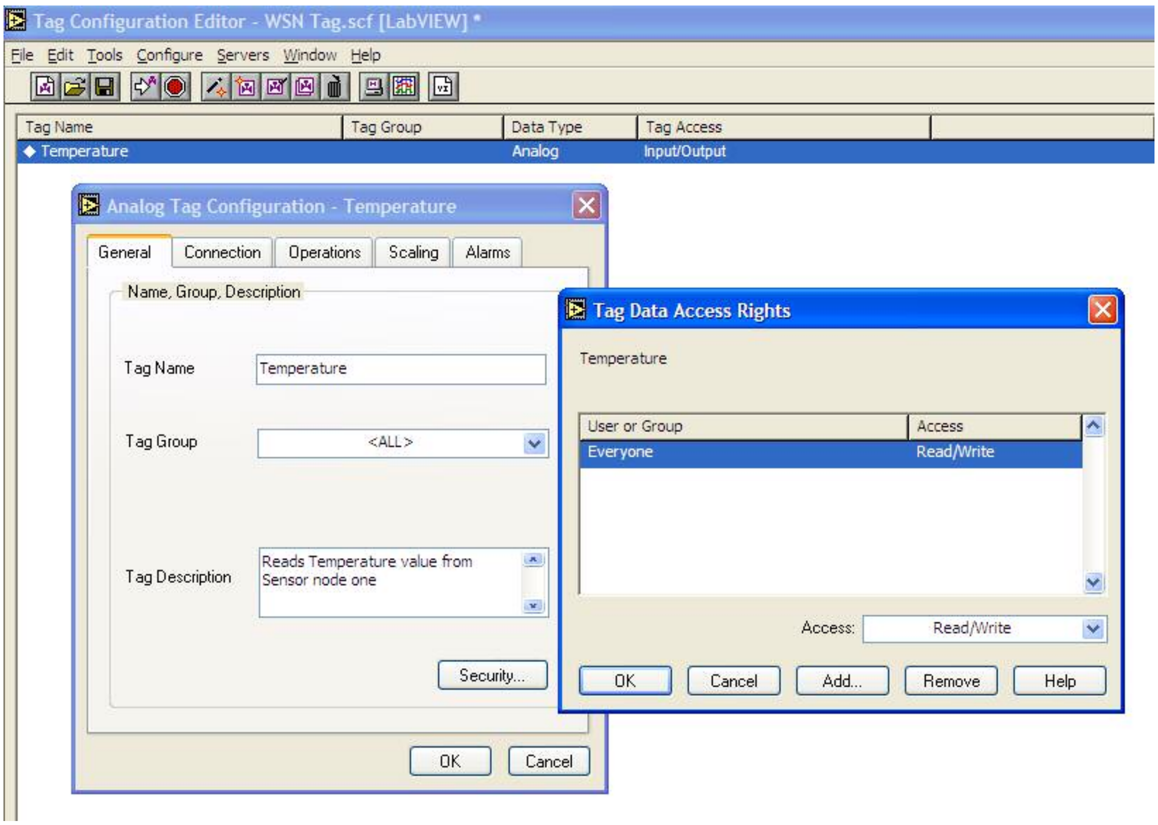


Figure 14. Configuring tag using Tag Configuration Wizard

In this configuration process attributes for the tag were set. These mainly include scaling, connections, alarms and dead-bands. These attributes are as follows;

- General—Attributes such as tag name, group, and description.
- Connection—Attributes that describe where the Tag Engine sends or receives values for the tag and how to access that data. These tags have access rights of input, output, or input/output. Memory tags are not connected to a real-world input/output point.
- Operations—Attributes that describe additional functionality that the Tag Engine performs on a tag or its values.

- Scaling—Attributes that describe which scaling function is applied to a tag value.
- Alarms—Attributes that describe abnormal process conditions for a given tag.

After successful creation and configuration of the tag, it has been saved in a .scf file.

The created tag can be monitored through National Instrument® tag monitor.

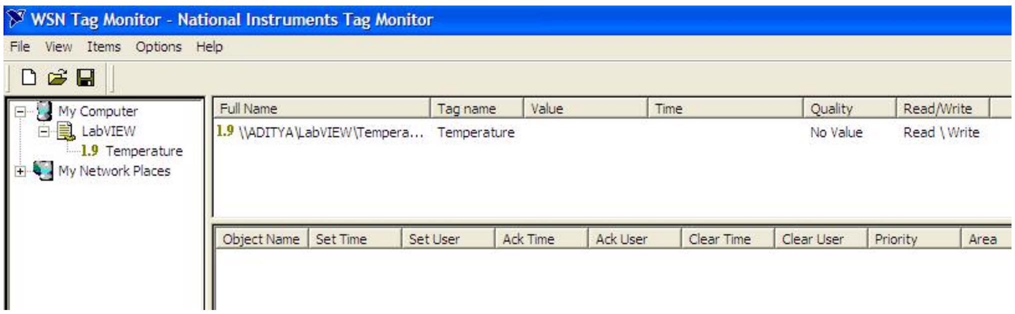


Figure 15. Tag Monitoring using NI tag monitor

Tags can be accessed by other computers over a network. A DSC Module server is a computer that allows tags configured in the current .scf file to be accessed by other computers connected to it. A client is a computer that gets its data through tags from one or more DSC Module servers. A DSC Module server also can act as a client and get its data from other DSC Module server computers. A .scf file for a DSC Module client can contain network tags from multiple DSC Module servers, as well as other types of servers.

<u>Security</u>

Security in the application has been implemented by setting up of user and group accounts. LabVIEW DSC module has a User Account Manager which creates and edits the properties of groups, user accounts. It assigns users to one or more groups and manages security accounts for applications.

The User Account Manager for WSN is shown as follows:

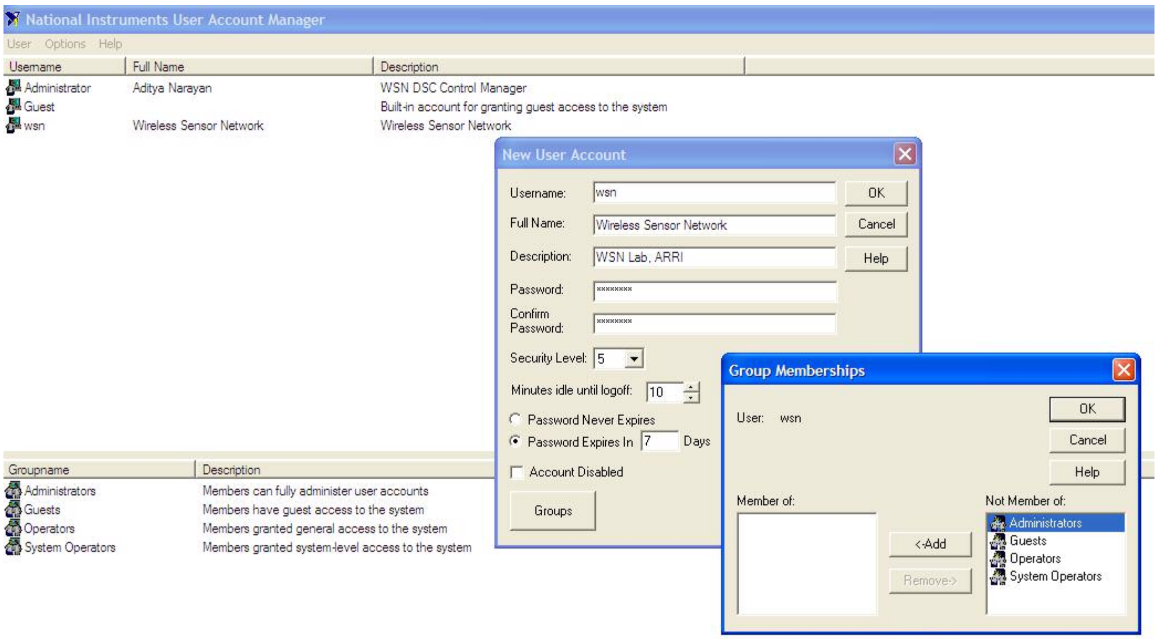


Figure 16. User Account Manager

The user privileges are defined as follows:

Administrator: Can fully administer user accounts and access to all

Guest: Can only view the reports

Operators: Can only access to general operations

System Operator: Can access system-level operations

The WSN DEC log-in screen is built implementing the DSC User Account Manager to set-up user privileges.

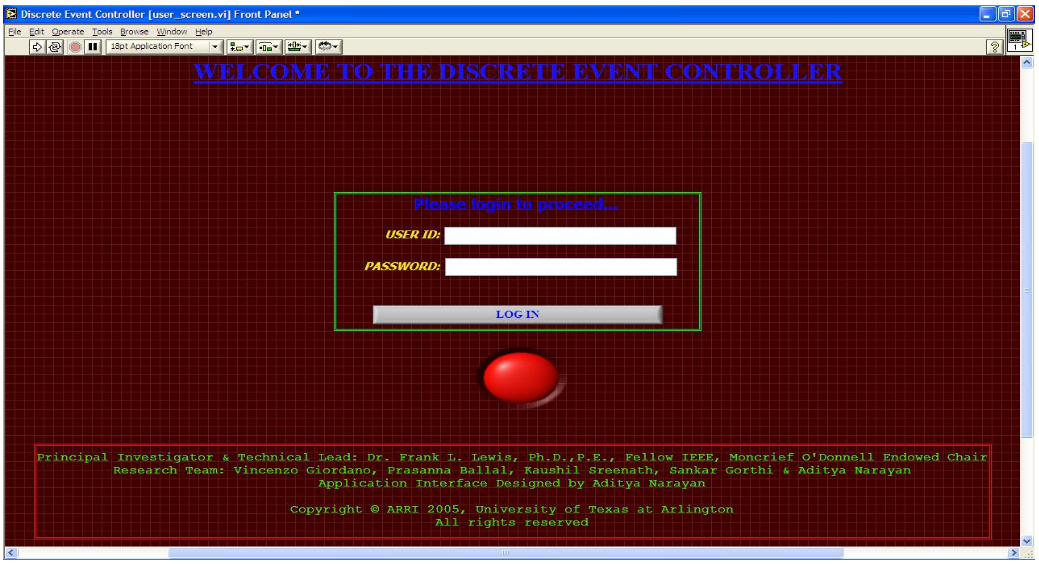


Figure 17. WSN User Log-in Interface

After successful log-in the user is directed to the main console.

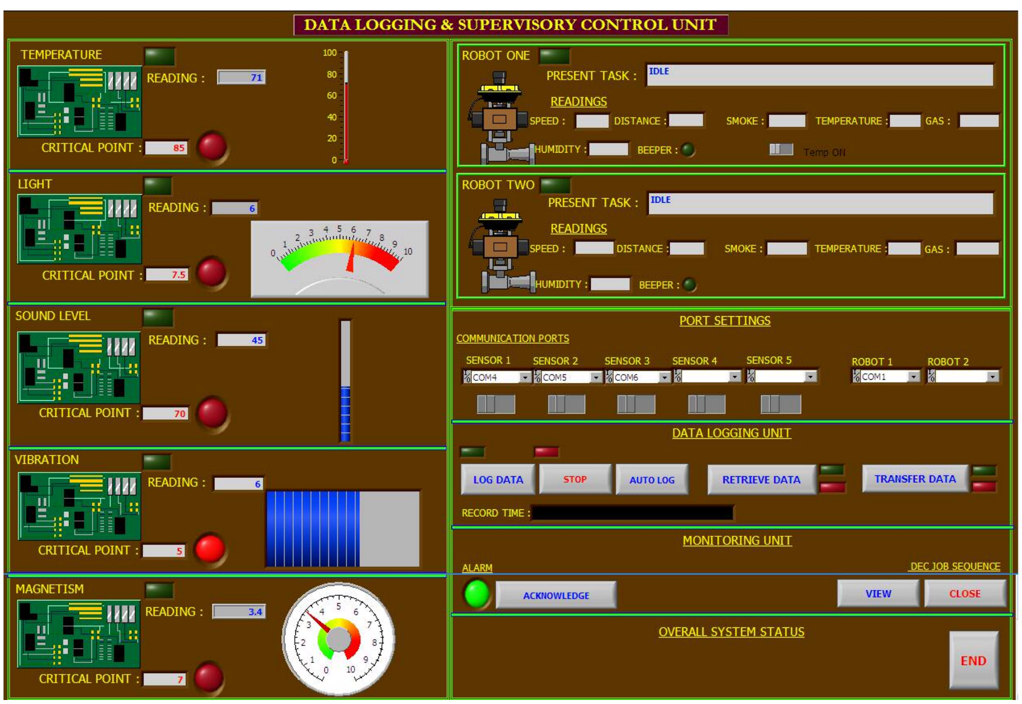


Figure 18. Front Panel of Supervisory Control Console

The corresponding block diagram for the supervisory control console is as follows:

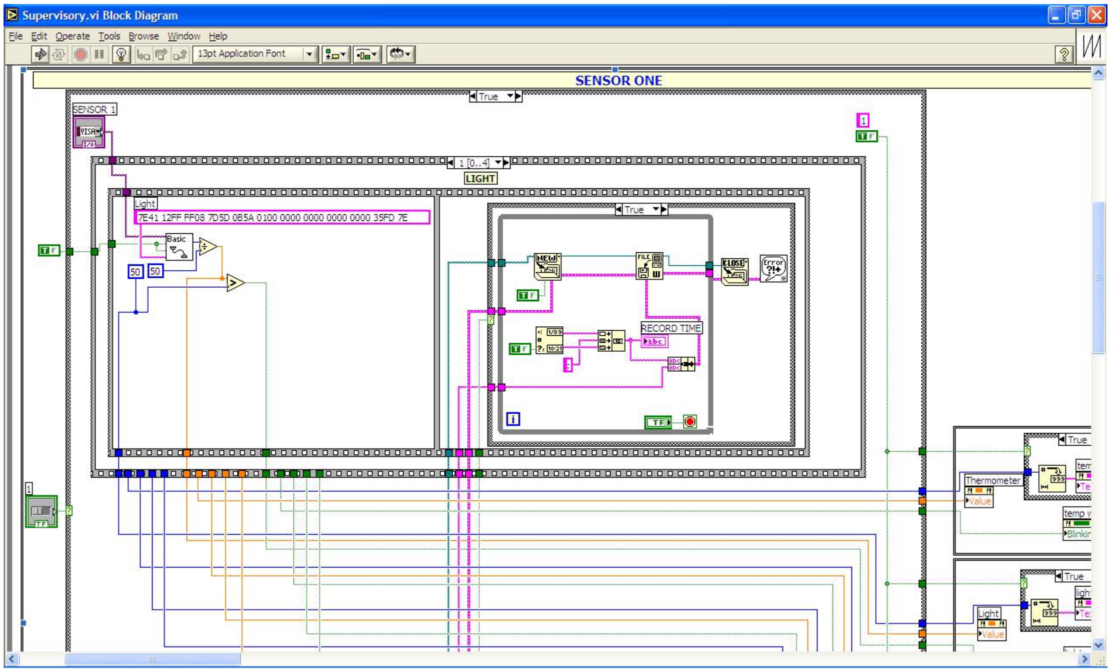


Figure 19. Partial Block Diagram of Supervisory Control Console

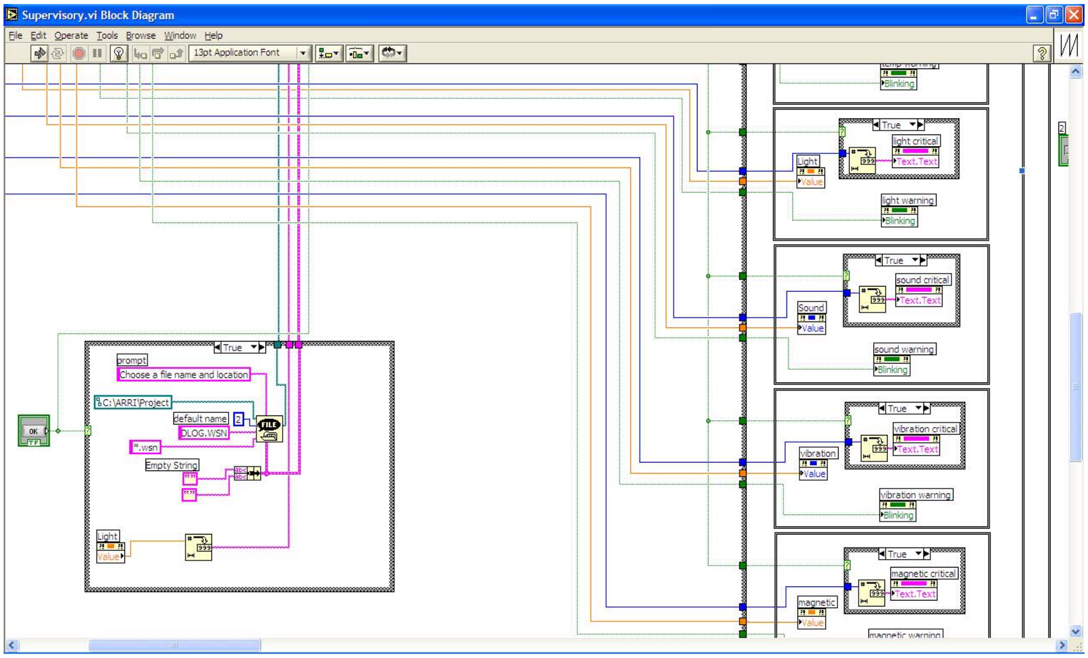


Figure 20. Partial Block Diagram of Supervisory Control Console

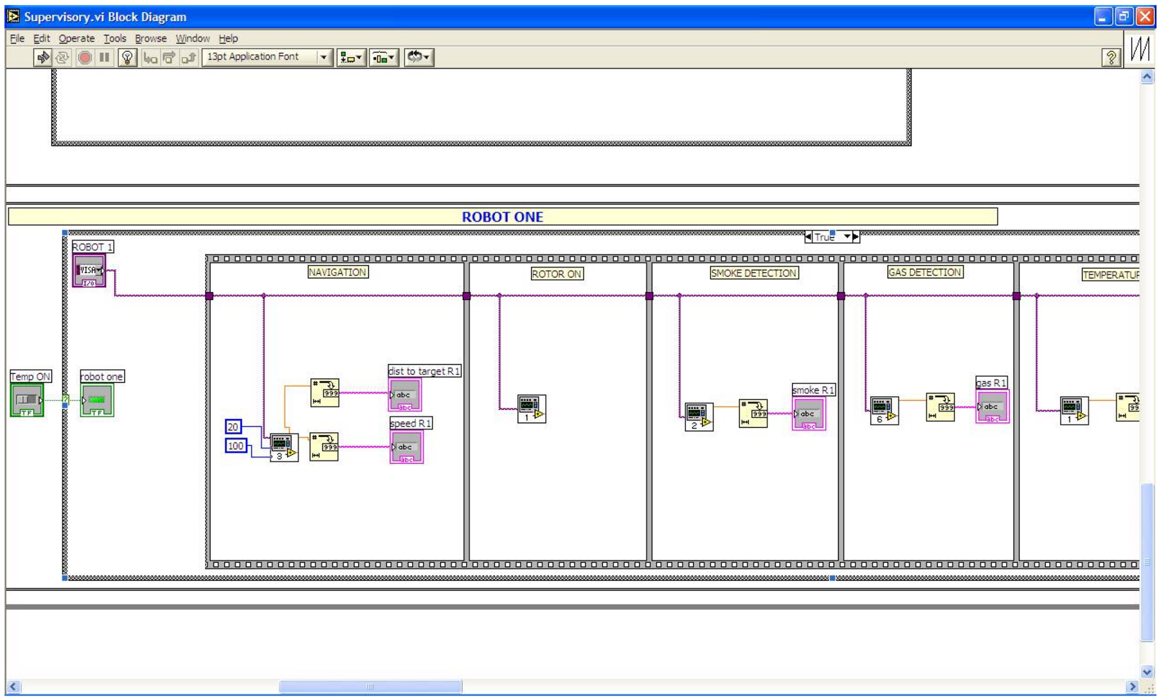


Figure 21. Partial Block Diagram of Supervisory Control Console

The block diagram of the Supervisory control Console contains several subVIs to accomplish different tasks. A few of these are given below;

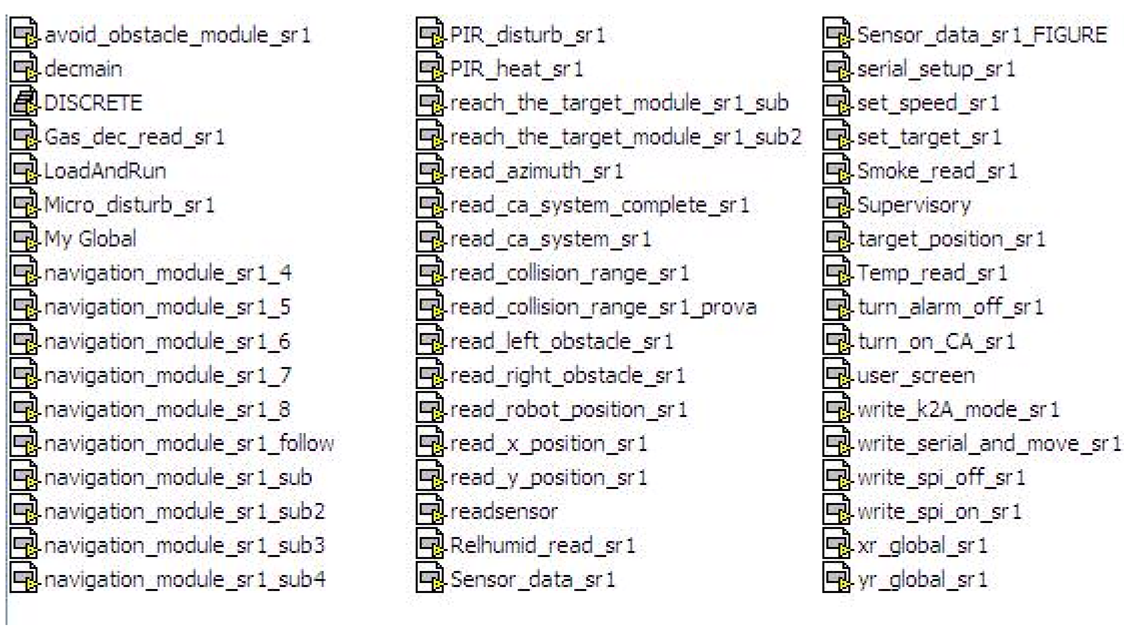


Figure 22. Partial List of subVis

Data-log Retrieval

The WSN DSC module uses the Citadel 5 database to store data. The stored data can be used to produce historical trending and report generation. The data can be retrieved from the Citadel 5 database using the historical data viewer in the LabVIEW® Measurement and Automation Explorer (MAX®). No additional programming is required for this retrieval. The historical data viewer allows saving multiple views of traces and settings. Views can also be created for alarms and events.

A simple trace for the temperature data measured by a cross bow mote UGS is shown as follows:

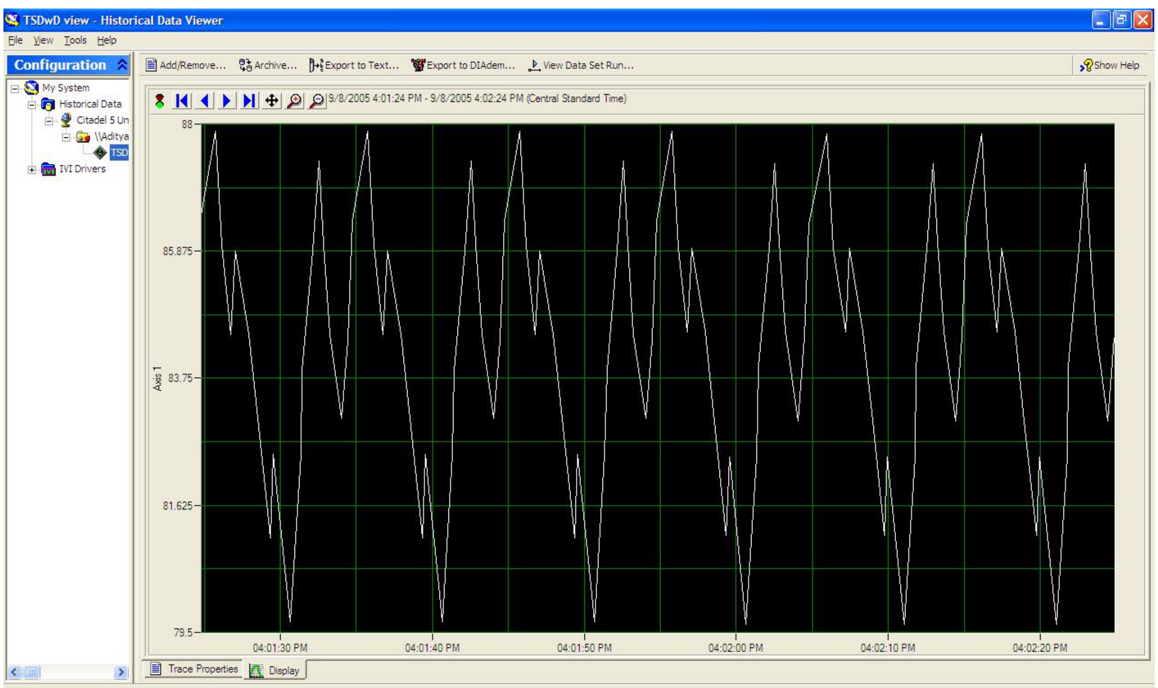


Figure 23. Historical Data Viewer

WSN on WEB

Another important application of WSN DSC module is the web publishing of the control console. The DSC console can be accessed and controlled from the web. So the WSN test-bed can be monitored from the web. The data from the historical database can be exported to html file which then can be published on the web. A demo of this feature is shown

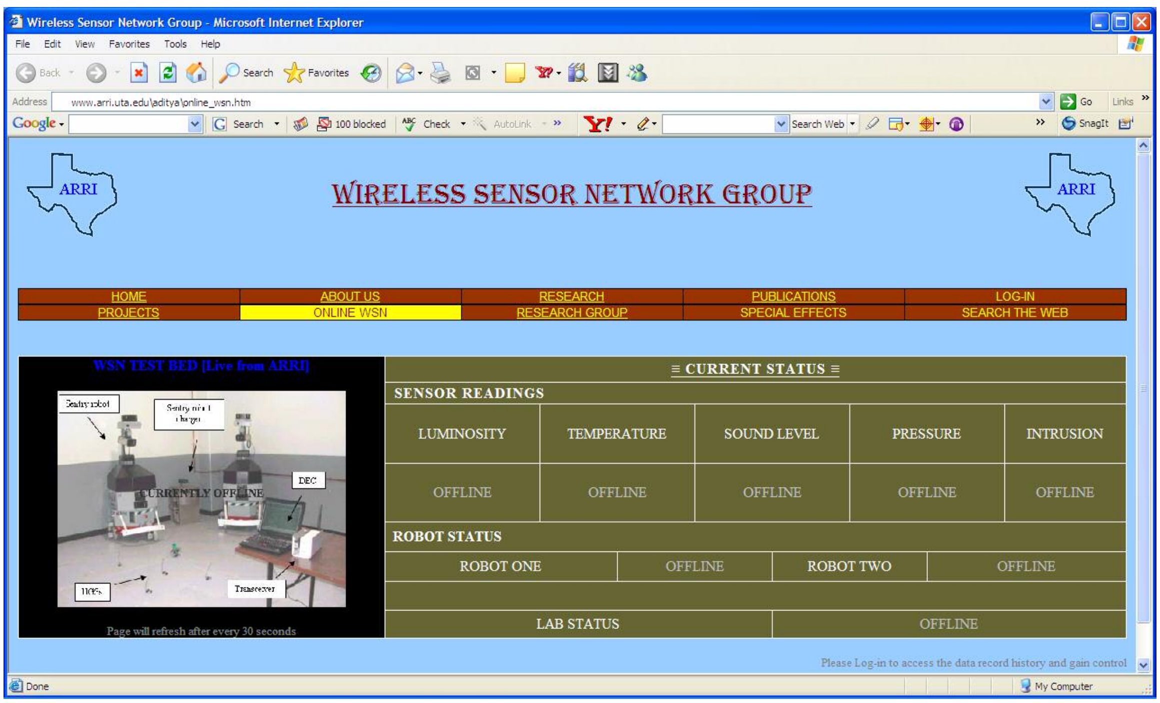


Figure 24. WSN on the Web

With proper authorization the operator can log-in to the Supervisory Control Console and monitor the test-bed over the web. Also with sufficient privilege the operator can control the environment using the discrete event controller (DEC).

This enhances the performance of the DEC by broadening the spectrum of operation and adding more versatility to the control system.

CHAPTER 6

CONCLUSION

6.1. Conclusion

In this thesis, the addition of Data-logging & Supervisory Control (DSC) module to the Wireless Sensor Networks (WSN) has been discussed. It is observed from the discussion that the DSC adds more strength and flexibility to the Wireless Sensor Networks by complimenting the Discrete Event Controller (DEC) with many supervisor level controls. The data-logging unit of DSC is a major advantage in managing the raw data acquired from the sensors. Data filtering concepts such as 'dead-band' plays a vital role in effective logging of sensible data. Alarms & events feature supplement to quicker data identification. Report generation and Web exportation schemes of DSC enhance data publishing techniques. Lastly and most importantly the security measures incorporated by the DSC unit is very important in data protection. Especially the setting up of different levels of security to different group of user adds more flexibility to the data security.

Programming through the DSC module might be a bit more complicated such as, new manual server creation and registration, tag configuration, server and tag monitoring and also the operation of the tag engine. The Citadel database used by the DSC module is constrained to a maximum size of 2GB which might create some difficult in managing large amount of data. The efficiency of the database in storing or managing large object files such as video or audio could be a point of concern. The programming language is mostly graphical. So unlike other text based programming languages, it could be a concern running these programs on an uninterrupted basis on relatively slow computers. Above all the programming demands a high skill-set of the programmer.

But in the end, the LabVIEW Datalogging and Supervisory Control module (DSC) is an effective toolbox in industrial process control and automation. PC-based data logging systems provide the most flexibility, customization, and integration to the control systems.

6.2. Future Scopes

There are still some issues in the data-logging and supervisory control area which yet to be explored. It becomes essential to make the supervisory control unit more and more intelligent so that it can adapt to the fast changing environment. There is always a scope of advancements in the data acquisition and data identification schemes. Scalability is a big challenge. Since a smart environment may consist of enormous tiny smart devices that communicate with each other via wireless or wired links, to provide information security protection (e.g., authentication, data encryption, access control, etc.) is not a trivial task, especially when the resources of such tiny devices are limited. A further research on threat modeling, formal security verification, and automatic incidence detection and containment in a fully automated environment is always welcomed.

APPENDIX

A. IMPORTANT LABVIEW® BLOCKS OF DSC

A.1. Controls Palette

The DSC Module installs the DSC Module controls palette and the following sub-palettes.

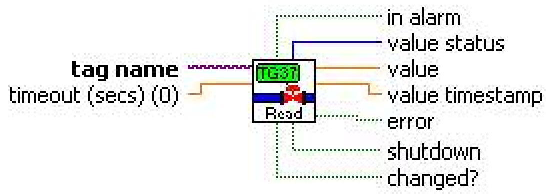
1. Alarms and Events—A set of display, format, query, and summary tools for handling alarms and events.
2. DSC Module Server Data Types—Various data type definitions which can be used to write VI-based servers for the DSC Module.
3. Pipes, Pumps, & Valves—Various Boolean pipe, pump, and valve objects.
4. Trends—Historical Trend XY graph, Teal-Time Trend waveform chart, and the NI Hyper-Trend control.
5. Vessels—Several tanks, a hopper, and a bin. All objects on this sub-palette are numeric.
6. Tag—The tag control to select a tag from those configured in the active configuration (.scf) file.

A.2 Functions Palette

The DSC Module installs the DSC Module VIs palette and the following Sub-palettes.

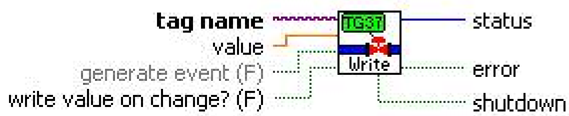
1. Tags—These VIs and functions are used to read the recent value of a tag, write a new value to a tag get data for a real-time trend.
2. Alarms & Events—These VIs are used to acknowledge alarms, display alarm summary or event history information, query logged alarm and event data, or obtain alarm summary status.
3. Tag Attributes—These VIs are used to read and write tag configuration parameters programmatically.
4. Historical Data—These VIs are used to read and write historical data, resample trace data, compute statistical data for a historical trend, convert historical trend data to a spreadsheet, or perform some other database administrative task.
5. Historical Data Data Set Logger—These VIs are used to retrieve information or values for sets of logged data.
6. System—These VIs are used to launch or shut down the Tag Engine, to enable or disable event logging, historical datalogging, or printing, or to manipulate the DSC Module environment.
7. System Security—These VIs are used to implement security in applications and to access security information.
8. DSC Module Server Development—These VIs are used to develop VI-based device servers.

IMPORTANT DSC BLOCKS

<table>
<tr><td>


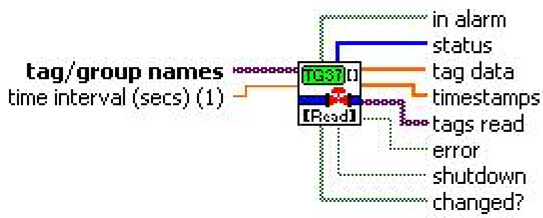
</td><td>

<u>Read Tag</u>

Reads the latest value of a tag from the Tag Engine.
</td></tr>
<tr><td>


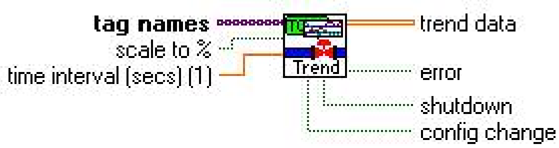
</td><td>

<u>Write Tag</u>

Updates the Tag Engine with a new value for memory, output, and input/output tags.
</td></tr>
<tr><td>


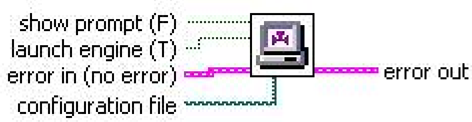
</td><td>

<u>Read Multiple Tags</u>

Reads the latest value for the tags in tag/group names.
</td></tr>
<tr><td>


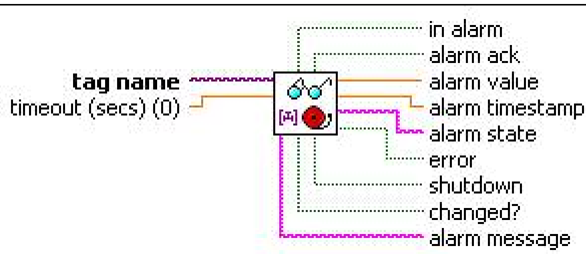
</td><td>

<u>Trend Tags</u>

Sets data for a real–time trend chart.
</td></tr>
<tr><td>


</td><td>

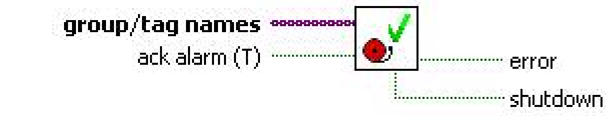
<u>Start Tag Monitor</u>

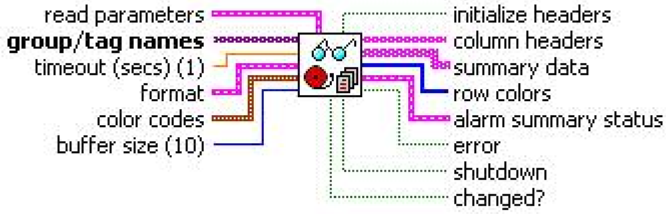
Starts the Tag Monitor.
</td></tr>
<tr><td>


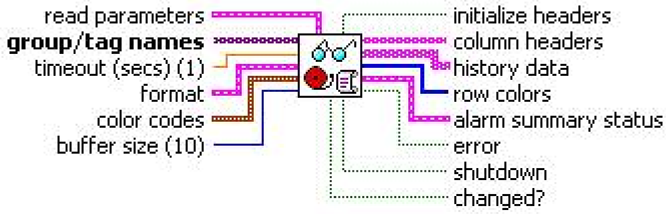
</td><td>

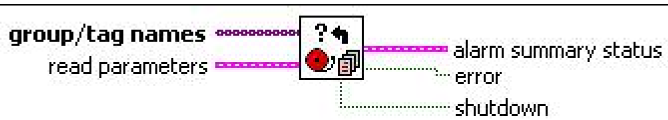
<u>Read Tag Alarm</u>

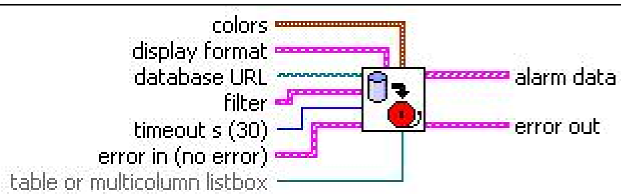
Read detailed alarm status from the Tag Engine. This Indicates if a tag is in alarm, which alarm state it is in, when the alarm occurred, at which value it occurred, and whether it has been acknowledged.
</td></tr>
</table>

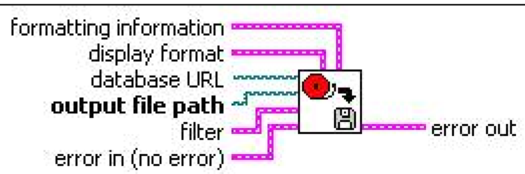
<table>
<tr><td>
</td><td><u>Acknowledge Alarm</u>

Acknowledge alarms from an application.</td></tr>
<tr><td>
</td><td><u>Read Alarm Summary</u>

Display current alarm information for a set of tags or tag groups within a given alarm priority range.</td></tr>
<tr><td>
</td><td><u>Read Event History</u>

Display all the alarms and events that have occurred since the Tag Engine started for a set of tags or tag groups within a given alarm priority range.</td></tr>
<tr><td>
</td><td><u>Get Alarm Summary Status</u>

Check the status of alarms in the system.</td></tr>
<tr><td>
</td><td><u>Alarm & Event Query</u>

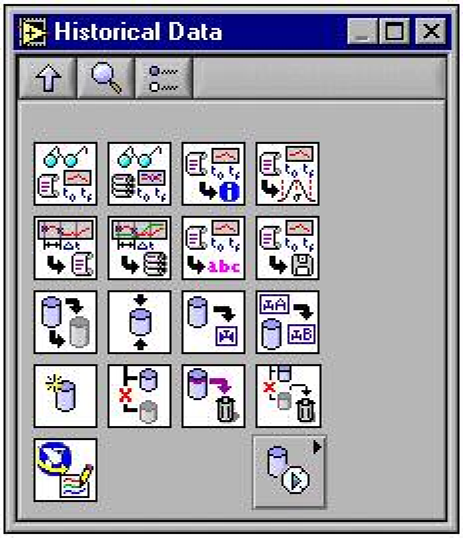
Extract data from a historical alarm and event database.</td></tr>
<tr><td>
</td><td><u>Alarm & Event Query To Spreadsheet File</u>

Export data from a historical alarm and event database to a delimited text or .html file.</td></tr>
</table>

Tag Attributes VIs and Function

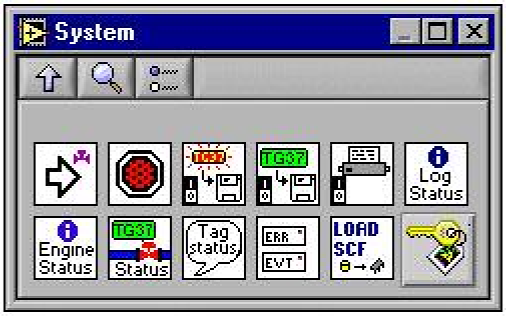
- Get Analog Tag Alarm Limit
- Get Tag IO Connection Info
- Get BitArray Tag Alarm Setting
- Get Tag List
- Get Discrete Tag Alarm Setting
- Get Tag Logging Info
- Get Group List
- Get Tag Range and Units
- Get Tag Alarm Enabled
- Set Multiple Tag Attributes
- Get Tag Attribute
- Set Tag Attribute
- Get Tag Bad Status Alarm Info
- Tag Attribute Ring
- Get Tag Description Group

Historical Data VIs



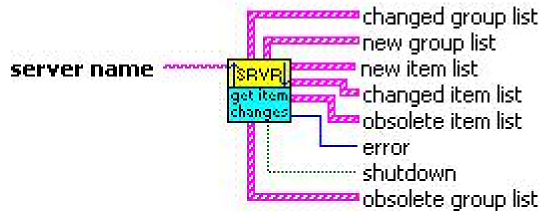
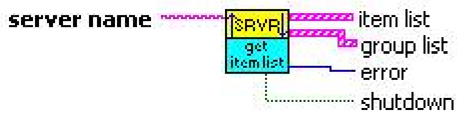
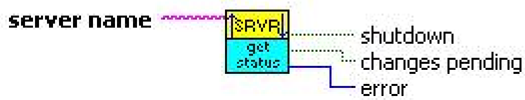
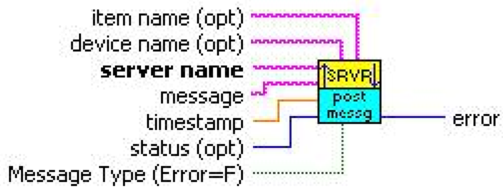
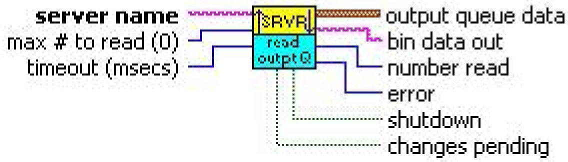
- Archive Traces
- Get Trace Info
- Call HDV
- Get Trace List
- Compact Database
- Get Trace Statistics
- Create Database
- Read Trace
- Decimate Trace
- Read Traces
- Decimate Traces
- Rename Historical Data
- Delete Traces
- Write Traces to Spreadsheet
- Delete Database
- Write Traces to Spreadsheet File
- Detach Database

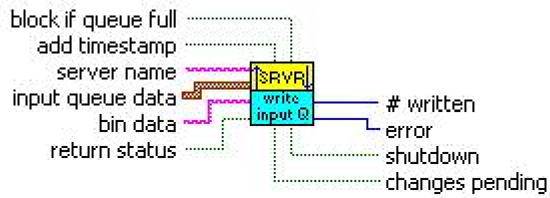
System Vis

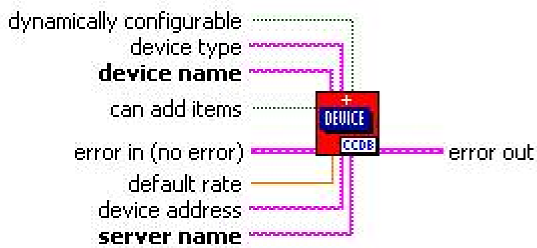


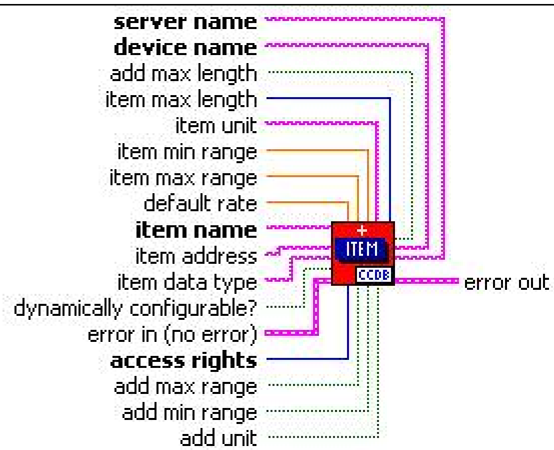
- Enable Event Logging
- Get Logging Status
- Enable Historical Data Logging
- Get Tag Status Info
- Enable Printing
- Load SCF
- Engine Launch
- Post System Error or Event
- Engine Shutdown
- Tag Status Handler
- Get Engine Status

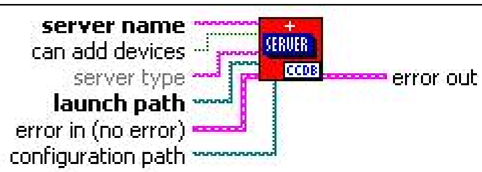
DSC Module Server Interface VIs

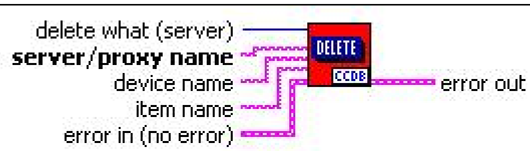
	SRVR Get Item Changes Returns a list of item changes.
	SRVR Get Item List Returns lists of items, item characteristics, and item refnums that the Tag Engine requests from a specific server.
	SRVR Get Status Polls the Tag Engine for the current server status.
	SRVR Post Message Writes error messages from the server to the Tag Engine where the messages can be logged and displayed to the user.
	SRVR Read Output Queue Receives new output values for output or input/output items from the Tag Engine.

<table>
<tr>
<td>
</td>
<td><u>SRVR Write Input Queue</u>

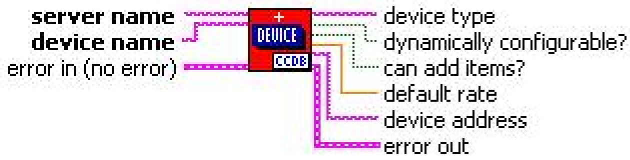
Writes input item and input/output item data to the Tag Engine.</td>
</tr>
<tr>
<td>
</td>
<td><u>SVRG Add Device Row</u>

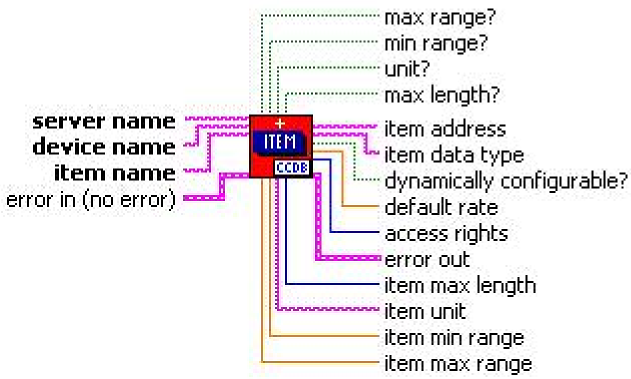
Registers a device for your server with the DSC Module.</td>
</tr>
<tr>
<td>
</td>
<td><u>SVRG Add Item Row</u>

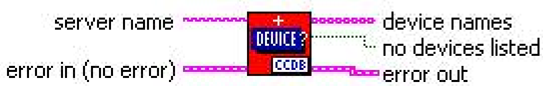
Registers an item for your server device with the DSC Module.</td>
</tr>
<tr>
<td>
</td>
<td><u>SVRG Add Server Row</u>

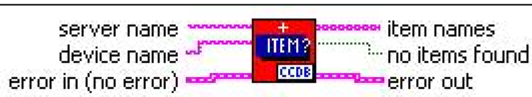
Registers your server for use with the DSC Module.</td>
</tr>
<tr>
<td>
</td>
<td><u>SVRG Delete Row</u>

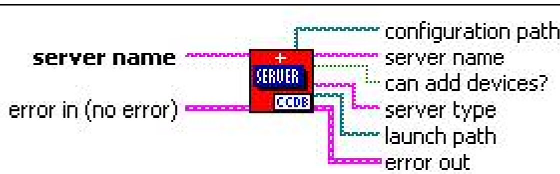
Deletes a specific row from the Server, Device, or Item tables.</td>
</tr>
</table>

<table>
<tr><td>
</td><td><u>SVRG Get Device Row</u>

Returns the information registered for the device name corresponding to server name.</td></tr>
<tr><td>
</td><td><u>SVRG Get Item Row</u>

Returns information registered for item name corresponding to the server name and device name.</td></tr>
<tr><td>
</td><td><u>SVRG Get Server Devices</u>

Returns a list of devices registered for the server name.</td></tr>
<tr><td>
</td><td><u>SVRG Get Server Items</u>

Returns a list of items registered for the server name and device name.</td></tr>
<tr><td>
</td><td><u>SVRG Get Server Row</u>

Returns the information registered for the server name from the Server table.</td></tr>
</table>

REFERENCES

[1] Akyldiz I., Su W., Sankarasubramaniam Y, Cayirci E., “A survey on sensor networks”, IEEE Communications magazine, August 2002

[2] Bronson, R., and Naadimuthu, G., *Operations Research,* 2 ed., Schaum’s Outline, McGraw-Hill, New York, 1997.

[3] Butler Z., Rus D., “Event-based motion control for mobile-sensor network”, IEEE Transactions. on Pervasive Computing, vol.2 issue 4, October-December 2003

[4] Christensen T., Noergaard M., Madsen C., Hoover A., “Sensor networked mobile robotics”, Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, June 2000

[5] Harris B., Lewis F., Cook D., “Machine planning for manufacturing: dynamic resource allocation and on-line supervisory control”, Journal of Intelligent Manufacturing, pp. 413-430, vol. 9, 1998

[6] Harris, B., Cook, D., and Lewis, F.L., “Automatically generating plans for manufacturing,” J. Intelligent Systems, vol. 10, no. 3, pp. 279-319, 2000.

[7] Howard, A.; Mataric, M.J.; Sukhatme, G.S.; "An incremental deployment algorithm for mobile robot teams", Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and System, vol.30, Pages:2849 – 2854, October 2002

[8] Vincenzo Giordano, Frank Lewis, Biagio Turchiano, Prasanna Ballal, Jing Bing Zhang "Supervisory control of mobile sensor networks: discussion and implementation"

[9] John Krumm Steve Shafer Andy Wilson Ubiquitous Computing Group

Microsoft Research Microsoft Corporation "How a Smart Environment Can Use Perception"

[10] Jun Wang, Yaling Yang and William Yurcik; "Secure Smart Environments: Security Requirements, Challenges and Experiences in Pervasive Computing"

[11] Philip Robinson and Michael Beigl. Trust Context Spaces: An Infrastructure for Pervasive Security. In First International Conference on Security in Pervasive Computing, 2003.

[12] Kusiak A. "Intelligent scheduling of automated machining systems." In Intelligent design and Manufacturing. A. Kusiak (ed.) Wiley, New York (1992)

[13] Lee J., Hashimoto H, "Controlling mobile robots in distributed intelligent sensor network", IEEE Transactions on Industrial Electronics, vol.50, no.5, October 2003

[14] Lewis F., "Wireless sensor networks", Smart environments: technologies, protocols, and applications, ed. D. J. Cook and S. K. Das, John Wiley, New York, 2004.

[15] McMickell M., Goodwine B., Montestruque L., “MICAbot: a robotic platform for large-scale distributed robotics”, Proceedings of the International conference of robotics and automation, September 2003

[16] Mireles J., Lewis F., “Intelligent material handling: development and

implementation of a matrix-based discrete event controller IEEE Transactions on Industrial Electronics, , vol. 48 , Issue: 6 , Dec. 2001

[17] Saridis G., “Intelligent robotic control”, IEEE Transactions on Robotics & Automation, vol. 28, no.5, May 1983

[18] Sibley G., Rahimi M., Sukhatme G., “Robomote: a tiny mobile platform for large-scale ad-hoc sensor networks”, Proceedings of the International conference of robotics and automation, May 2002

[19] Tacconi D., Lewis F., “A new matrix model for discrete event systems: application to simulation”, IEEE Control System Magazine

[20] Tilak S., Abu-Ghazaleh N., Heinzelman W., "A taxonomy of wireless microsensor network models," ACM Mobile Computing and Communications Review, Vol. 6, No. 2, 2002

[21] National Instruments, “LabVIEW DSC Tutorials and Examples”

[22] National Instruments “LabVIEW Datalogging and Supervisory Control Module Release Notes”

[23] National Instruments "LabVIEW Datalogging and Supervisory Control Module Run-Time System Release Notes"

[24] National Instruments "LabVIEW Datalogging and Supervisory Control Module Developer Manual"

[25] National Instruments "LabVIEW Datalogging and Supervisory Control Module Run-Time System Manual"

[26] National Instruments "VI-Based Server Development Toolkit Reference Manual"

BIOGRAPHICAL INFORMATION

Aditya Narayan received his Bachelor of Engineering degree in Electronics & Telecommunication Engineering from Utkal University, India in 2002. He then joined in Masters Program in Electrical Engineering at University of Texas at Arlington in fall 2004. Motivated by the interest in the field of system controls and robotics, he joined the Automation & Robotics Research Institute (ARRI) as a research assistant in 2004-05 under the able supervision of Dr. Frank L. Lewis. During his tenure he worked on the wireless sensor network test-bed. This thesis is mainly based on his research study and implementation work done on this wireless senor network test-bed.