International Journal of Engineering Research-Online
A Peer Reviewed International Journal
Articles available online http://www.ijoer.in

Vol.3., Issue.4., 2015
(July-Aug)

RESEARCH ARTICLE

ISSN: 2321-7758

MONITORING OF CRITICAL INFRASTRUCTURE USING WEB BASED SCADA

VAISHALI N. PAWAR[1], Prof. B. G. GAWALWAD[2]

[1]Student, Dept. of E & TC, SVIT COE, Nasik, India

[2]HOD, Dept. of E & TC, SVIT COE, Nasik, India

pawarvaishali97@gmail.com, balaji_gawalwad@rediffmail.com

ABSTRACT

For decades, the monitoring of CIs has centered on supervisory control and data acquisition (SCADA) systems, where operators can monitor and control the behavior of the system. The reach of the SCADA system has been hampered by the lack of deployment flexibility of the sensors that feed it with monitoring data. Nowadays SCADA systems are used for Home automation, Greenhouse automation, E-agriculture, Power generation and distribution etc. Basically these SCADA applications include Level Monitoring, Light & Climate Control, Security & Surveillance, control and manage spatially separated utility sites and Control of Shutters & Doors and so on. With the arrival of new hardware and software technologies here a system is proposed which can perform the similar SCADA applications at lower cost and lower maintenances. This paper proposes a viable solution for SCADA like applications which include various industrial applications as well as proposes a fine web based solution to access all these acquired data and equipments. Here a remote web based application is used which will allow the user to access the inter-organizational data/equipments in industries via internet, it also overcome the problem of weak encryption used by the SCADA. Arduino Platform is the new technology used for supervisory control purpose. Alarm handling, Access Control, Automation, Logging, Archiving, Report generation, Interfaces to hardware and software etc are some features provided by the application. In future this system using .NET platform may replace the whole SCADA solution.

Keywords- Critical infrastructure monitoring; Arduino platform; Supervisory Control and Data Acquisition (SCADA); Security & Surveillance; Human Machine Interface

INTRODUCTION

SCADA is Supervisory Control and Data Acquisition Systems. SCADA programs are used in industrial process control applications for centralized monitoring and recording of pumps, tank levels, switches, temperatures etc. SCADA systems are also referred to as HMI i.e. Human Machine Interface. SCADA (supervisory control and data acquisition) is a type of Industrial Control Systems (ICS). Often, they are used in critical infrastructures (CIs) where security and safety are vital factors. Due to this, they have to comply with strict regulatory standards [1].

Computers offered flexibility in programming and communicating with field data acquisition units that was previously being done by hard wired equipments [3]. SCADA monitors, controls and alarms the plant and/or regional operating systems

from a centralized location. It includes the communication of information between a SCADA central host computer, many scattered units and/or Programmable Logic Controllers. For example, in a water filtration plant [2], the remote units measure the pressure in pipes and report the readings to the central computer located somewhere in the control tower. In case of any anomaly, the SCADA system would alert the main station of the problem for other details like the severity of the anomaly and measurement values in an organized fashion. The systems may vary from simple, like temperature reporting in a building to complex like monitoring the traffic on many traffic lights.

I. ELEMENTS OF SCADA SYSTEM

1.1 SCADA Master Station Computer Systems

It is the repository of the real-time or near real-time reported data collected from the remote terminal units connected to it. It is generally standard computer hardware equipment and very few SCADA system suppliers have ventured out to make their own computer equipment [3].

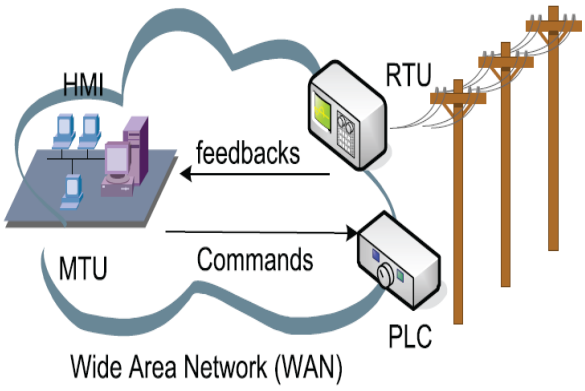

Figure 1: Elements of SCADA System

1.2 Human-Machine Interface

This is the eye candy part on the host station. The values that have been stored in the host computers are presented to the human operator in an understandable and comprehensible form using HMIs. These may provide trending, diagnostic or management information and detailed schematics and animations representing the current states of the machines under its control. Pictorial representation being more understandable to humans is the preferred form in SCADA HMIs [4].

1.3 Remote Terminal Units (RTUs)

An RTU is a normally a transducer or a sensor which allows the electrical circuitry to interface with the process instrumentation and control equipment. The physical parameter like pressure, temperature etc. are measured through a change in electrical property of some component in the transducer which is indicative of the physical change [5].

1.4 Programmable Logic Controllers

The use of microprocessors on RTUs has helped RTUs become smarter with increased functionality. PLCs have been built around the philosophy of automation. Reprogrammability being the biggest asset, PLC based RTUs can be debugged and fixed on the field itself along with adding new features like support for multiple polling, exception reporting, time-tagging etc. This also enables them to execute simple logical processes without involving the master station. Vendors using different type of communication and coding on these equipment has led to standardization of protocols and languages for RTUs too, for example standardized control programming language, IEC 61131-3. These languages require very less training and are based on intuitive approach unlike procedural languages like C and FORTRAN.

1.5 SCADA Communication

The conveying of data from an RTU to the master station and commands from the host to the RTU need to be done over a communication system. Also, since a SCADA system might not be localized to just a single plant, the vastness of the network also has to be catered to along with speed, accuracy, security and performance being among other important issues [6]. Before the computer networking solutions were made available, most systems for communication were voice communication based. SCADA communication systems were also built using the same infrastructure and had the same bandwidth limitations. But, with the corporate now wanting to include the SCADA information network into their core networks over security concerns, SCADA systems have also embraced LANs and WANs for seamless integration with everyday office computer networks [7]. This has an advantage for the

corporate users that they would not need a separate parallel network for SCADA systems.

II. GENERATIONS OF SCADA

2.1 First generation: "Monolithic"

In the first generation, computing was done by mainframe computers. Networks did not exist at the time SCADA was developed. Thus SCADA systems were independent systems with no connectivity to other systems. Wide Area Networks were later designed by RTU vendors to communicate with the RTU [1]. The communication protocols used were often proprietary at that time. The first-generation SCADA system was redundant since a back-up mainframe system was connected at the bus level and was used in the event of failure of the primary mainframe system.

2.2 Second generation: "Distributed"

The processing was distributed across multiple stations which were connected through a LAN and they shared information in real time. Each station was responsible for a particular task thus making the size and cost of each station less than the one used in First Generation. The network protocols used were still mostly proprietary, which led to significant security problems for any SCADA system that received attention from a hacker [1].

2.3 Third generation: "Networked"

These are the current generation SCADA systems which use open system architecture rather than a vendor-controlled proprietary environment. The SCADA system utilizes open standards and protocols, thus distributing functionality across a WAN rather than a LAN. It is easier to connect third party peripheral devices like printers, disk drives, and tape drives due to the use of open architecture. WAN protocols such as Internet Protocol (IP) are used for communication between the master station and communications equipment [1]. Due to the usage of standard protocols and the fact that many networked SCADA systems are accessible from the Internet; the systems are potentially vulnerable to remote cyber-attacks. On the other hand, the usage of standard protocols and security techniques means that standard security improvements are applicable to the SCADA systems [7], assuming they receive timely maintenance and updates.

III. PROPOSED WORK

3.1 Problem Definition

SCADA is the important research areas now a day in embedded systems. Web based development of SCADA is more important so that anyone can keep the watch on real time picture of industrial environment. SCADA/PLCs are available in market of various types but no web interface is available till date. PLCs used cause comparatively more cost.

Its problem definition can be provided as follows:

1. Collection of accurate real time environmental parameters.
2. Continuous monitoring on these parameters.
3. Defining the set points.
4. Depending on these set points fire a supervisory control action.
5. Alarm handling.

3.2 Scope

There are various types of applications are available for this supervisory control and monitoring system. SCADA is Supervisory control and Data Acquisition system. Nowadays SCADA systems are used for Home automation, Greenhouse automation, E-agriculture, Power generation and distribution [8] etc. Basically these SCADA applications include Level Monitoring, Light and Climate Control, Security and Surveillance, control and manage spatially separated utility sites and Control of Shutters and Doors and so on. With the arrival of new hardware and software technologies here a system is proposed which can perform the similar SCADA applications at lower cost and lower maintenances.

This paper proposes a viable solution for SCADA like applications which include Temperature level monitoring, Voltage level monitoring and Displacement control. This system can not only perform these industrial applications but also proposes a fine web based solution to access all these acquired data and equipments. Here a remote based application is used which will allow the user to access inter organizational data/equipments in industries via internet, it also overcome the problem of weak encryption used by the SCADA. Arduino

Platform is the new technology used for supervisory control purpose. Alarm handling, Access Control, Automation, Logging, Archiving, Report generation, Interfaces to hardware and software etc are some features provided by the application. In future this system using .NET platform may replace the whole SCADA solution.

IV. SYSTEM ARCHITECTURE AND METHODOLOGY

The basic concept was to create a system which can perform like supervisory control and data acquisition system for monitoring and controlling the Industrial equipments. This system implements two ways for application remote monitoring and command, the first is based on SMS/call functionalities provided by GPRS network and the second one is accomplished by using a web page interface provided when using a web server. Figure 2 shows Proposed System Architectural View

Following are some functions provided by the system:

- Flexible and open architecture
- Alarm Handling
- Access Control
- Automation
- Logging, Archiving, Report Generation
- Interfaces to hardware and software

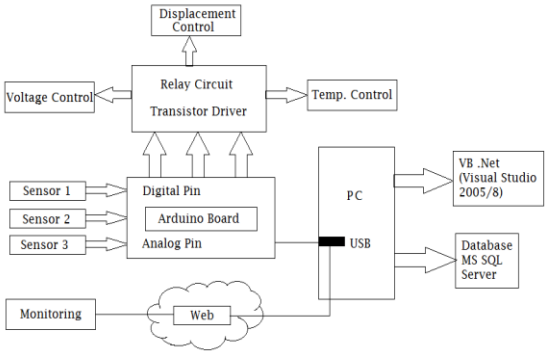

Figure 2: Proposed System Architectural View

4.1 Hardware Specifications

There are various types of sensors available in the market for different purposes. In this application we are using LM35 temperature sensor, Limit Switch for displacement control & under voltage circuitry using opam 741 to control under voltage. Also various types of components are used in this design. Few of them are listed below:

4.1.1 LM53 Temperature Sensor:

The LM35 series are precision integrated-circuit temperature sensors, with an output voltage linearly proportional to the Centigrade temperature. Thus the LM35 has an advantage over linear temperature sensors calibrated in ° Kelvin, as the user is not required to subtract a large constant voltage from the output to obtain convenient Centigrade scaling. The LM35 does not require any external calibration or trimming to provide typical accuracies at room temperature. Low cost is assured by trimming and calibration at the wafer level. The low output impedance, linear output, and precise inherent calibration of the LM35 make interfacing to readout or control circuitry especially easy. The device is used with single power supplies, or with plus and minus supplies. As the LM35 draws only 60 µA from the supply, it has very low self-heating of less than 0.1°C in still air. The LM35 is rated to operate over a −55°C to +150°C temperature range. The LM35 series is available packaged in hermetic TO transistor packages. Figure 3 shows a LM35 Temperature Sensor.

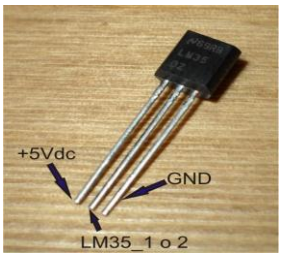

Figure 3: LM35 Temperature Sensor

4.1.2 Opamp 741

The Operational Amplifier is probably the most versatile Integrated Circuit available. It is very cheap especially keeping in mind the fact that it contains several hundred components. The most common Op-Amp is the 741 and it is used in many circuits. Its main purpose is to amplify (increase) a weak signal a little like a Darlington Pair. The OPAMP has two inputs, INVERTING (-) and NON-INVERTING (+), and one output at pin 6. The 741 integrated circuit looks like any other 'chip'. However, it is a general purpose OPAMP. You need only to know basic information about its operation and use. Figure 4 shows an Opamp 741 IC.

Figure 4: Opamp 741 IC

4.1.3 Limit switch

An electrical switch is any device used to interrupt the flow of electrons in a circuit. Switches are essentially binary devices as they are either completely on ("closed") or completely off ("open"). The simplest type of switch is one where two electrical conductors are brought in contact with each other by the motion of an actuating mechanism. In any case, the final output of any switch will be (at least) a pair of wire connection terminals that will either be connected together by the switch's internal contact mechanism ("closed"), or not connected together ("open"). These switches are operated by means of a lever which is clamped to a knurled shaft extending from the operating head. These devices can be easily field converted to clockwise, counter clockwise, or both directions of operation without any loose parts. The head is interlocked with the base unit to resist accidental shearing. Lever type switches can be equipped with a variety of operating levers: roller lever, adjustable roller lever, micrometer adjustment roller lever, rod lever, one-way rod or roller lever and fork. Figure 5 shows a Limit Switch.

Figure 5: Limit Switch

4.1.4 Arduino Hardware

Arduino hardware is Open source prototyping platform. Open-source hardware shares much of the principles and approach of free and open-source software. In particular, we believe that people should be able to study our hardware to understand how it works, make changes to it, and share those changes. To facilitate this, all of the original design files (Eagle CAD) for the Arduino hardware is released. These files are licensed under a Creative Commons Attribution Share-Alike license, which allows for both personal and commercial derivative works, as long as they credit Arduino and release their designs under the same license. Arduino can sense the environment by receiving the input from variety of sensors and can affect the surrounding by controlling the light, fan, motors or other actuators. It is having digital, analog and USB pins. It is based on ATMega8 Microcontroller IC. Figure 3.2.6 shows Arduino board of ATMega8 Microcontroller IC.

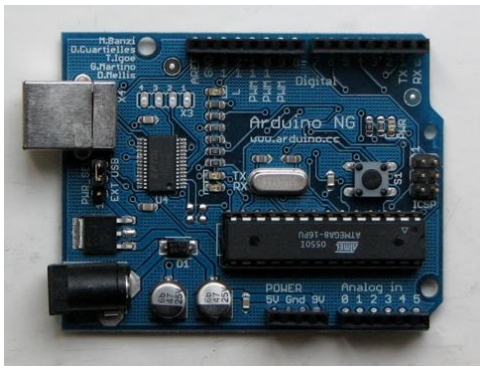

Figure 6: Arduino Board of ATMega8 Microcontroller IC

4.2 Circuit Schematic

Figure 7 shows circuit schematic for various sensors of proposed system.

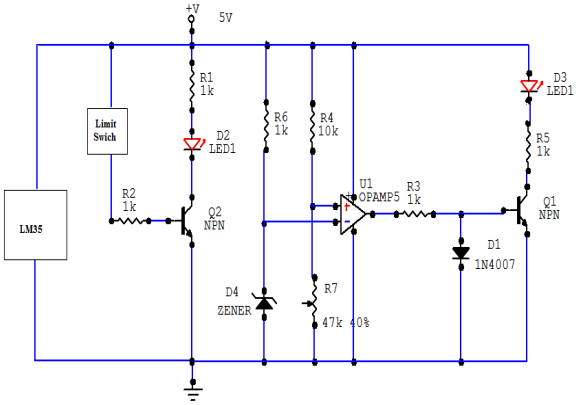

Figure 7: Circuit Diagram of Proposed System

V. SOFTWARE REQUIREMENT SPECIFICATION

5.1 Product Overview

This system will be an extended version of the PLC/SCADA. This system will be developed as web-based SCADA solution for environmental

parameter monitoring. Also our system is going to produce a quality results at low cost.

5.2 Software Details

5.2.1 .Net Interface

The .NET Framework is the infrastructure for the new Microsoft .NET Platform. The .NET Framework is a common environment for building, deploying, and running Web applications and Web Services. The .NET Framework contains a common language runtime and common class libraries like ADO .NET, ASP .NET and Windows Forms to provide advanced standard services that can be integrated into a variety of computer systems. The .NET Framework provides a feature rich application environment, simplified development and easy integration between a numbers of different development languages. The .NET Framework is language neutral. Currently it supports C++, C, Visual Basic, and Script (The Microsoft version of JavaScript). Microsoft's Visual Studio .NET is a common development environment for the new .NET Framework. We are using VB.Net for developing web based interface for web monitoring.

5.2.2 Arduino Software

The Arduino software is also open source. The source code for the Java environment is released under the GPL and the C/C++ microcontroller libraries are under the LGPL. The Arduino language is merely a set of C/C++ functions that can be called from your code. Your sketch undergoes minor changes (e.g. automatic generation of function prototypes) and then is passed directly to a C/C++ compiler (avr-g++). All standard C and C++ constructs supported avr g++ should work in Arduino.

VI. TEST PLAN

6.1 Software to be tested

The software to be tested is the Web page clustering system which is used for document clustering. So the system to be tested is a clustering algorithm which will cluster the web pages.

6.2 Testing strategy

Each module is tested separately. Components for unit testing are:

- Reads all the sensor values and insert these values to the specified locations in the database.
- Define the set points.
- Compare the sensed values with the defined set point values
- If any value exceeds the set point fire a supervisory control action on it.
- After a control action when the sensor values fall down to normal state reset the action.

The criteria selected for identifying the unit test module is identity modules that have core functionality implemented and its execution is independent of other modules.

6.2.1 Integration testing

Integration testing deals with integration of different module and testing it to check if all the modules work accordingly. Integration testing takes as it is input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers its output for system testing. The modules which are united tested and combine testing is performed to see if the correct information is passed between the modules as per the algorithm.

6.2.2 Validation testing

The process of evaluating software during or at the end of the development process to determine whether it satisfies specified business requirements. The requirements of the scratch removal system for video in painting are to detect the scratch from video then remove scratches. And finally rebuild the video. Finally do analysis of video in painting algorithm for scratch removal.

6.2.3 GUI Testing

GUI testing is the process of

(a) Testing a products graphical user interface to ensure it meets its specification

(b) Surety of the navigation between icons/buttons with source code.

6.2.4 High-order testing

The System tests will focus on the behaviour of our system. Various modules will be tested all together to confirm the overall functionality of the system. Overall, the system tests will test the integrated system and verify that it meets the requirements defined in the requirements document.

VII. **DATA TABLES AND DISCUSSIONS**

The results produced with our implementation are very promising. First of all we have to read the sensor values accurately. Three sensors we are using-Under voltage sensor, Temperature sensor, Displacement sensor. All the three are producing the results but reading these values together and placing them in proper location in database in the computer is most challenging job. By heating the temperature sensor with solder gun, we checked whether the value is varying or not. In the same way we checked whether the value is varying or not for another two sensors. And the values are changing in the proper textboxes only.

After that we do a microcontroller programming in Arduino's ATMEGA8 microcontroller. Set points are defined as per requirement and the sensor values are compared with proper set points. A central computer system on field is continuously fed with the sensor values in the database. It should be clear that for web monitoring this system is connected to the internet with any of technology. On web side administrator is provided with user admin authentication i.e. user-id and password. He can monitor any time on field using any device may be laptop or mobile phone connected to internet. Administrator will get to know about sensor values and control action (if any) continuously.

VIII. **CONCLUSION**

The Web-based SCADA is the area of research nowadays. Here we developed a small SCADA like simple automated application which senses the real time environmental parameters and comparing the values with set point it fires a control action automatically.

On the field side all the sensors lies which continuously monitors on real time environment. All these sensor values are given to the computer system with the new technology Arduino! As it is having analog as well as digital input and output pins feeding the input of sensor values from analog pins is much easier. It is also having a USB port. So communication with the computer system is done easily.

A control action can be fired any time automatically if the value exceeded set point with the help of program done on microcontroller ATMEGA8 of Arduino. Set points are compared with the sensor values in this program. A central computer system on field is continuously fed with the sensor values in the database. It should be clear that for web monitoring this system is connected to the internet with any of technology. On web side administrator is provided with user admin authentication i.e. user-id and password. He can monitor any time on field using any device may be laptop or mobile phone connected to internet. Administrator will get to know about sensor values and control action (if any) continuously.

IX. **REFERENCES**

[1]. P. J. May and C. Koski, "Addressing public risks: Extreme events and critical infrastructures," Rev. Policy Res., vol. 30,pp.139-159,2013.

[2]. S. Peter and G. Weber, "Monitoring drinking water pipelines," Eurescom message, issue 1/2011, Mar. 2011.

[3]. M. Merabti, M. Kennedy, and W. Hurst, "Critical infrastructure protection: A 21st century challenge," in Proc. Int. Conf. Commun. Inf. Technol. (ICCIT), Mar. 2011, pp. 1–6.

[4]. C. Alcaraz, G. Fernandez, and F. Carvajal, "Security aspects of SCADA and DCS environments," in Critical Infrastructure Protection. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 120–149.

[5]. António M. Grilo, Jaime Chen, Manuel Díaz, Daniel Garrido, and Augusto Casaca, "An Integrated WSAN and SCADA System for Monitoring a Critical Infrastructure", IEEE Transactions on Industrial Informatics, Vol. 10, No. 3, pp. 1755-1764, August 2014.

[6]. S. A. Boyer, "SCADA Supervisory Control and Data Acquisition", Englewood, CO, USA: ISA—International Society of Automation, Jun. 15, 2009. ISBN: 978-1-936007-09-7.

[7]. B. Galloway and G. P. Hancke, "Introduction to industrial control networks," IEEE Commun. Surveys Tuts., vol. 15, no. 2, pp. 860–880, May 2013.

[8]. A. Grilo et al., "A wireless sensor and actuator network for improving the electrical power grid dependability," in Proc. 8th Conf. Next Gener. Internet, 2012, pp. 71–78.