Microcontroller based Automation system using Industry standard SCADA

Mayur Avhad Vinit Divekar Harshad Golatkar Sanket Joshi
Department of Electronics,
Vidyalankar Institute of technology
Wadala(E), Mumbai

***Abstract*—SCADA (Supervisory Control and Data Acquisition) is a type of industrial control system. In typical industrial automation applications, PLCs are used to communicate between the plant and SCADA. In this paper we are using a low cost microcontroller to understand the various control techniques and capabilities of SCADA and implement them in real time. A simple temperature control scheme is developed using ATMEGA2560 microcontroller and industry standard SCADA software Vijeo Citect v7.2. We make use of MODBUS ASCII protocol which is an industry standard communication protocol to set up a temperature control system which collects real time temperature data, processes the data and performs the desired control action. Major contribution of the paper is in developing MODBUS library for communication between microcontroller and SCADA and implementing control algorithms such as ON/OFF and proportional controller on data obtained in real time. Thus this paper presents a low cost automation scheme which can be easily extended for more complicated control schemes including wireless control. The project has been developed as first step towards a project on pipeline leakage detection scheme using wireless sensor nodes.**

Keywords–SCADA, Atmega2560, MODBUS

I. INTRODUCTION

SCADA systems historically distinguish themselves from other ICS systems by controlling large scale processes that can include multiple sites, and large distances. With the advent of wireless technology microcontroller based SCADA systems are being studied in literature. In [1], authors give an overview of SCADA system with LabVIEW interface to monitor and measure temperature parameter with serial RS232 communication. In [2], authors have developed integrated wireless SCADA system using the mobile telephone network with GPRS technique for monitoring & accessing the performance of remotely situated device parameter such as temperature, pressure, humidity on real time basis. In [3], the authors focus on the development of an automated setup for the measurement of temperature of any industrial environment, for example boilers, refrigerator etc. They have developed hardware to withstand industrial environment using PIC18F452microcontroller.

Wireless sensor networks are commonly being used with SCADA to monitor and control many complex and critical processes in real time [4]. In [5], authors have proposed a Wireless Sensor Network Communication Model for Automation of Electric Power Distribution. While in [6], authors review the various standards for communication in wireless sensor networks that are being deployed in the control of critical and complex processes and also analyze their security. Authors have identified a set of threats and potential attacks in their routing protocols, and suggested recommendations and countermeasures to help Industry protect its infrastructures. In [7], authors present architecture for mobile sensor network which is appropriate for SCADA system aiming to make it mobile to enhance the conventional system that SCADA is currently having. This is also in connection with the aim to level up the SCADA components from fixed to mobile.

With the increasing need of wireless SCADA system, one major and critical application could be in the detection of leakages in the underground water pipelines. In such a system, wireless sensor network can be used to effectively monitor the data related to pressure and flow at various locations in the pipelines. In [8], authors discuss the issues and challenges in the use of wireless sensor networks in the protection and monitoring of the critical and essential infrastructures of pipelines carrying oil, gas, water, and other important resources. The paper presents an architectural model that can be used to provide this monitoring and control functions. The model includes an overview of networking and routing protocols that can be used to provide the necessary communications. In addition, the paper provides discussions and recommendations concerning network reliability and the use of different wireless sensor technologies and protocols. In [9], authors have presented study of TriopusNet a mobile wireless sensor network system for autonomous sensor deployment in pipeline monitoring. To show the benefits and feasibility of TriopusNet, they have prototyped and tested the system in a real pipeline test bed. In [10], authors discuss how wireless sensor networks can increase the spatial and temporal resolution of operational data from pipeline infrastructures and thus address the challenge of near real-time monitoring and eventually control.

Using this earlier work as reference, we have created our own system for monitoring data in real time and use it for implementing control. Unlike systems mentioned above which are made to withstand industrial conditions, our setup for temperature measurement and control is at an experimental level. Hence, the hardware specifications are not very rigorous. We have used ATMEGA 2560 microcontroller. We have

978-1-4799-2275-8/13/$31.00 ©2013 IEEE

followed the similar algorithm for temperature set point & continuous temperature measurement given in [3] although; we have not used any simulation tool. We are using Schneider electric's Vijeo Citect v7.2, SCADA software to perform control and automation.

The main objective of this project is to understand the basic working of the industry standard SCADA software and develop an interface between a microcontroller and SCADA to form a low cost system. The various capabilities of SCADA are verified using a standalone control system on Vijeo Citect v7.2. To demonstrate supervisory control, we have selected a temperature control system and the interface protocol selected is MODBUS ASCII.

The MODBUS protocol supports master-slave control which is very essential in supervisory control from SCADA that acts as Master to Microcontroller which acts as slave. The microcontroller chosen is Atmega2560 which is an 8 bit AVR microcontroller having high performance and low power. The Atmega2560 board has provision for analog and digital I/O ports. The system comprises of temperature as analog input to Vijeo Citect v7.2 through temperature sensor. For a particular temperature range appropriate control is achieved from SCADA software to the Atmega2560 board which drives a DC Fan and a DC Motor. "Figure 1" shows block diagram of the system.

II. HARDWARE & SOFTWARE COMPONENTS

A. *SCADA software (VijeoCitect v7.2)*

Vijeo Citect v7.2 is used for operating and monitoring components [14]. It is the new Process Automation system of Schneider Electric. With its powerful visualization capabilities and operational features, it delivers actionable insight faster, enabling operators to respond quickly to process disturbances and thereby increase their effectiveness.

The graphics, controls, configuration data and programming associated with a Vijeo Citect v7.2 installation is configured and implemented through projects. A project acts as a digital representation of production facility, allowing the entire system to be monitored and controlled in real-time.

Vijeo Citect v7.2 is made up of a several configuration tools and a runtime section [14]. Vijeo Citect v7.2's architecture can be divided into three distinct area of functionality:

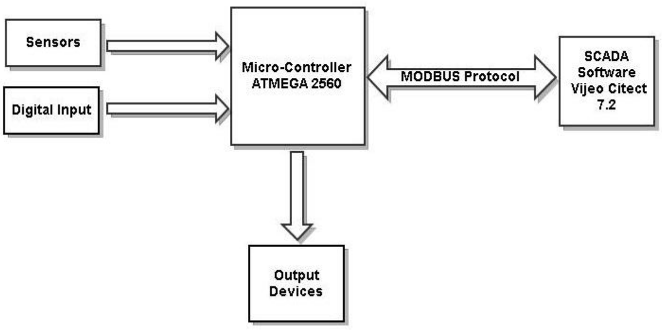

Fig. 1. Block Diagram of the System

- Configuration
- Runtime
- Drivers

Configuration involves the tasks necessary to prepare and build a project, while runtime is the implementation of a project in a live production environment. Drivers enable communication with devices via a number of communication protocols. The driver defines the specific project settings necessary for Vijeo Citect v7.2 to communicate with a particular device.

Vijeo Citect v7.2 can communicate with an array of I/O Devices, including PLCs, loop controllers, and distributed control systems (DCS). Drivers enable communication with devices via a number of communication protocols (including Ethernet, TCP/IP, and Serial). This includes information about:

- Boards
- Ports
- Devices
- Tag addressing

The components which can be incorporated in a project are logically divided across the following categories:

- Graphics components
- Tags
- Alarms
- System components
- Communications components
- I/O Server components
- Cicode / CitectVBA

Using above components of Vijeo Citect v7.2, a simple system can be built which can be used to demonstrate the interfacing between SCADA and microcontroller system & the automation which can be achieved through it. We have developed such a system using all the above components whose graphic layout is shown below in Figure 2.

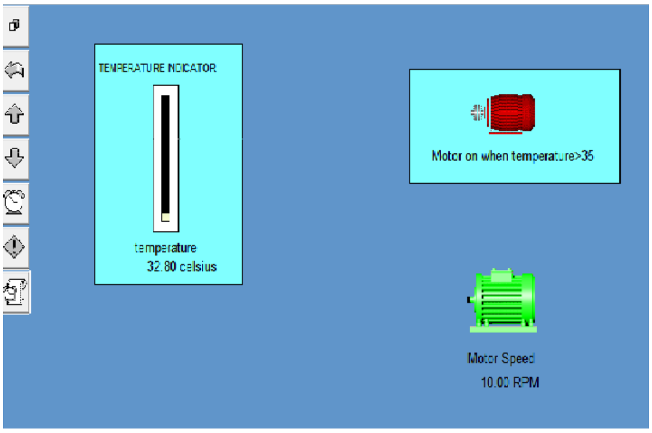

Fig. 2. A Graphic Page on Vijeo Citect v7.2

In order to achieve desired automation, 'cicode' must be written. We have used following cicodes -

For ON-OFF Control:-

if (temperature>35) then

motor_control=1;

else

motor_control=0;

end;

For Proportional Control:-

error=35-temperature;

if temperature <35 then

*motor_speed =5*error;*

else

motor_speed =3(error);*

end;

B. *Atmega 2560*

The Atmega2560 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the Atmega2560 achieves throughputs approaching 1 MIPS (Millions Instructions Per Second) per MHz allowing the system designer to optimize power consumption versus processing speed [13].

The architecture is more code efficient than conventional CISC microcontrollers. By combining an 8-bit RISC CPU with In-System Self- Programmable Flash on a monolithic chip, the Atmel Atmega2560 is a powerful microcontroller that provides a highly flexible and cost effective solution to many embedded control applications. It has 256K Bytes of In-System Self-Programmable Flash Memory.

C. *MODBUS Protocol*

Modbus protocol supports Master-Slave Technique which is widely used in many control processes such as SCADA systems [15]. This protocol defines a message structure that controllers will recognize and use, regardless of the type of networks over which they communicate. It describes the process a controller uses to request access to another device, how it will respond to requests from the other devices, and how errors will be detected and reported. It establishes a common format for the layout and contents of message fields [16].

During communications on a Modbus network, the protocol determines how each controller will know its device address, recognize a message addressed to it, determine the kind of action to be taken, and extract any data or other information contained in the message. If a reply is required, the controller will construct the reply message and send it using Modbus protocol [15].

1. *Transactions on Modbus Network*

Controllers communicate using a master–slave technique, in which only one device (the master) can initiate transactions (called 'queries').The other devices (the slaves) respond by supplying the requested data to the master, or by taking the action requested in the query. Typical master devices include host processors and programming panels. Typical slaves include programmable controllers.

The master can address individual slaves, or can initiate a broadcast message to all slaves. Slaves return a message (called a 'response') to queries that are addressed to them individually (Refer Figure 3).Responses are not returned to broadcast queries from the master. The Modbus protocol establishes the format for the master's query and slave response.

2. *Memory*

MODBUS protocol contains mainly two types of functions A) Data and control functions Diagnostic functions. In all, there are 24 data & control functions and 17 diagnostic functions.

Memory considerations are important while programming a microcontroller. The microcontroller Atmega 2560 has program memory of 256 kb which is comparatively large. The actual memory required for our system was 8456 bytes i.e. only 3.2% of total memory available in microcontroller. Thus, more functions can be added to the program as per requirement of application, without worrying about memory constraints.

D. *Peripherals*

Figure 4 shows actual hardware of the project. Hardware contains various peripherals which are interfaced on PCB's. We have designed three different PCB's-

1) Controller Board
2) Peripheral Board
3) Power Supply Board

The Atmega2560 board consists of microcontroller. It contains IC'S that enable communication such as MAX232 which is dual driver/receiver for serial communication and FT232 for USB communication. It also includes 6 pin ISP connector for programming via AVRISP mkII.

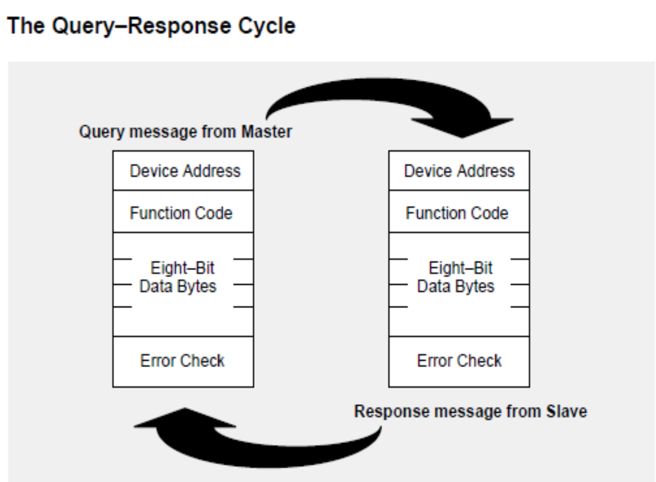

Fig. 3. Query-Response cycle in MODBUS network

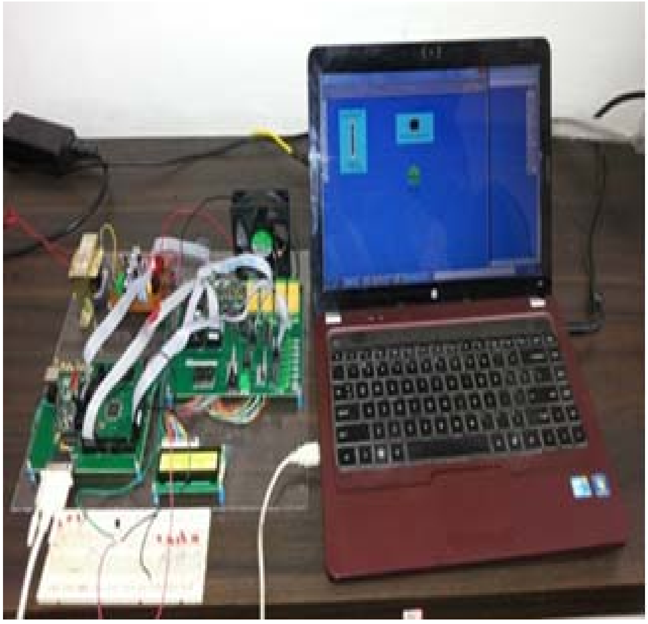

Fig. 4. Hardware Setup

Peripheral board contains many hardware components that can be used for digital input, digital output, analog input, analog output. It contains switches, relays, LED's, an LCD display, Buzzer etc. It also contains IC's such as-

1) High Current Driver ULN2003 (Relay driver).
2) Motor Controller Driver IC L298 which is a DC motor driver that can drive 2 DC motors with up to 4.5V to 46V and up to 2A current rating each.
3) DAC 0808 (8-Bit D/A Converter) is an 8-bit monolithic digital to analog converter.

Power supply board includes various components such as transformers, voltage driver IC's etc. that deliver power to various hardware components of the project.

III. COMMUNICATION BETWEEN SCADA AND MICROCONTROLLER

Before proceeding towards microcontroller programming, it is important to understand how the communication between SCADA software & microcontroller takes place. To establish communication, SCADA software will send a handshaking signal to microcontroller in MODBUS ASCII format.

Communication starts when microcontroller responds to this signal. The query format for the handshaking signal is shown in Table 1. Microcontroller should echo back the received frame as response.

Once handshaking is achieved, further communication takes place. According to the system developed on SCADA, further queries will be sent whose format will be similar to the one given above. The microcontroller should respond to these queries in MODBUS ASCII format.

The MODBUS functions used in this paper are listed in Table 2.

TABLE I. QUERY FORMAT

START	**ADDRESS Slave ID**	**FUNCTION Code**	**DATA**				**LRC CHECK**
			Starting Address		*No. of Points*		
			Hi	***Low***	***Hi***	***Low***	
:	01	02	00	00	00	10	ED

TABLE II. MODBUS FUNCTIONS

Function Code	**Name**
02	Read Input Status
03	Read Holding Register
04	Read Input Registers
05	Force Single Coil
06	Preset Single Register

IV. PROGRAMMING THE MICROCONTROLLER

The ATmega2560AVR is supported with a full suite of program and system. The programming is done on AVRStudio which is an integrated development platform (IDP) created by Atmel[13], for developing and debugging Atmel AVR microcontroller based applications.

The microcontroller is programmed so that it receives the MODBUS query frame, defragments it into its components (function code, address etc.), performs the necessary actions, then it forms a response frame to the corresponding MODBUS query. It then notifies the SCADA software about the action taken by sending this response frame. All these actions must be performed in an orderly manner and within a short interval of time. Inherent speed of processing of microcontroller (Up to 16 MIPS Throughput at 16 MHz) ensures that. Also, we have used hardware interrupt & timer interrupt mechanism to ensure faster processing and real-time data acquisition from peripheral devices.

A simple algorithm is followed to make programming easier.

A. *Algorithm*

1) Initialize all the Ports, Registers & Counters which are being used. Initialize UART for communication. Wait for a character to be received on receive buffer of UART, generate an interrupt upon reception.(The query frame will be received one character at a time)
2) Inside ISR of UART, Copy the received characters into an array. Also calculate slave id, function code etc.
3) Convert the received ASCII array into a HEX array for processing.(For MODBUS ASCII protocol, received data will be in ASCII format & microcontroller does not understand ASCII)
4) Calculate LRC of the received Query. Compare the calculated LRC with the received Query's LRC. If it does not match with received LRC, do nothing and proceed to next step.(Important to detect errors during communication)

5) Otherwise do all the processing for the MODBUS function code received.

6) Create an array of ASCII characters formed after collecting appropriate data from peripherals & calculating LRC for the response string.

7) Transmit this array one character at a time.

V. RESULTS

All the control actions in the system are governed by SCADA software which continously communicates with microcontroller via MODBUS protocol .

Figure 5 shows graphical representation of real time changes taking place in temperature. Initially, the temperature is constant. A temperature disturbance is introduced by turning on external heat source, due to which temperature starts rising.

A limit on temperature is set at 35 degree celcius, above which controller starts taking corrective action. Thus as seen from Fig 5, when it crosses the threshold, the controller will drive the fan to turn on at point x.

When temperature falls below the set point value at point y, the fan turns off.Thus, simple regulatory control is achieved usingon-off control of fan.

In this paper, proportional control has also been achieved & it is demonstrated by using DC motor whose speed varies according to the error between actual temperature &set point value.

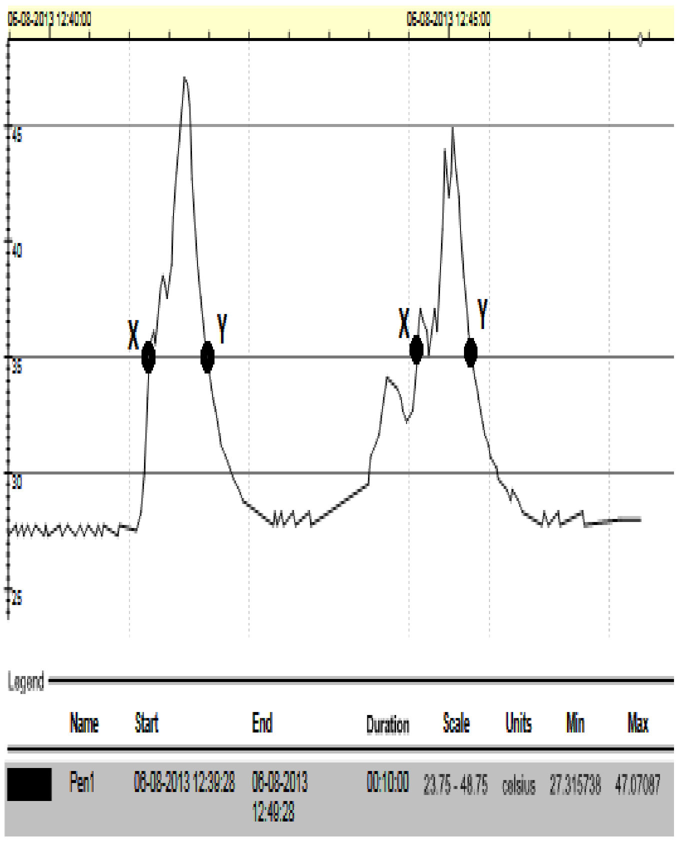

Fig. 5. Real Time Temperature Data on SCADA

VI. CONCLUSION

This paper presents a microcontroller based low cost automation system using Industry standard SCADA software(Vijeo Citect v7.2) and microcontroller (Atmega2560). To facilitate communication between the two, we have used MODBUS protocol, which is industry standard protocol. To demonstrate the functions of the system a temperature control system has been set up. This system senses the real time temperature data, compares it with the set point and takes a corrective action based on ON/OFF or proportional control law to achieve the desired set point. Results obtained reveal that the microcontroller has been successfully programmed to receive a Modbus frame, process it, and give an appropriate response. We have managed to reply to a Modbus query in a proper format using microcontroller.

The microcontroller used in this project, Atmega2560 is one of the most powerful microcontrollers around. Thus, we have successfully integrated microcontroller, SCADA and MODBUS protocol to build a general purpose system which can be easily modified to suit our needs.

VII. FUTURE SCOPE

The system presented in this paper integrates microcontroller based control system with industry standard SCADA. Scalability is the biggest advantage of the system described in this paper. However, To verify the feasibility of the idea at development stage a general purpose system which supports limited number of analog and digital I/O's has been built up. This can be easily extended to a variety of analog and digital inputs. By making minor changes in hardware & software we can easily expand the I/O capability which could satisfy the requirement of a small plant.

Real time temperature control is implemented with simple control laws as ON/OFF and proportional control. This can be further extended to more complex control including multivariable control.

In the current system, MODBUS library is built with certain data and control MODBUS functions for limited number of I/O's provided in the hardware (Refer Table I). This can be easily extended to include Diagnostic functions to implement entire MODBUS protocol. Depending on application, more functions can be easily incorporated in the system by programming the microcontroller to understand the function specific query & to give its appropriate response. Also, using powerful SCADA software provides an inherent flexibility as it is built to handle large scale data & processes.

The interfacing between microcontroller and SCADA is done via serial cables which can lower the performance in large scale application. This can be remedied by using Zigbee modules for wireless communication. Such modules have plug & play capability which will simply replace the cables.

It can be extended for variety of applications in process and control industries. The future extension plan for the project is to develop a Wireless SCADA system which could be used to monitor leakages in water pipelines.

ACKNOWLEDGMENT

We are thankful to Prof.Siddharth Dattagupta (IIT Bombay) for providing us this research opportunity. The authors are deeply indebted to Dr.Anjali Deshpande, Department of Electronics, Vidyalankar Institute of Technology, Mumbai for helping us prepare the project and for her remarks and suggestions throughout the development of the project. We take this opportunity to thank Mr.Anand Paralkar, Embedded systems expert in Vidyalankar Institute of Technology, for his guidance on technical aspects throughout this project. In addition, we would like to thank Mr. C.S. Limaye, Directorate Technocrats of 'SuperTech Instrumentation" for providing us with SCADA software, Vijeo Citect v7.2.

REFERENCES

[1] Neeraj Khera, Sumit Balguvhar, "Design of microcontroller based wireless SCADA system for real time data", UACEE International Journal of Advances in Electronics Engineering, volume 2:issue 1 ISSN 2278 - 215X (Online).

[2] Dr.Aditya Goel, Ravi Shankar Mishra, "Remote data acquisition using wireless-SCADA system", International Journal of Engineering (IJE), volume (3) : issue (1).

[3] Haider Ali, Ahmed Ali, Riaz Ul Hassnain Syed, Ajmal Khan & Ihsanullah Khan, "SCADA implementation of industrial temperature automation", IJCSNS International Journal of Computer Science and Network Security, vol.11:no.8, August 2011.

[4] Siti Rohainiza BintiChe Mood, "Development of SCADA system based on wireless sensor network", Thesis, Universitiy Teknikal, Malaysia Melaka, May 2009.

[5] M. Muthukumar, N. Sureshkumar, and MA.Bhadri Narayan, "A wireless sensor network communication model for automation of electric power distribution", the Technology Interface Journal/Spring 2010, Murugan, Nedunchezhian, Matheswaran(selected paper from the Proceedings of the IAJC-IJME 2008 Conference), volume 10:no. 3 issn#1523-9926,[Online],Available:

http://technologyinterface.nmsu.edu/Spring10/

[6] Cristina Alcaraz, and Javier Lopez, "A security analysis for wireless sensor mesh networks in highly critical systems", Dept. Computer Science, University of Malaga (Spain).

[7] Gelogo, YE, And Kim, TH, "Wireless sensor network architecture for SCADA system: review", information, 14, (10) pp. 3343-3350, issn 1343-4500 (2011).

[8] Imad Jawhar, Nader Mohamed, and Khaled Shuaib, "A framework for pipeline infrastructure monitoring using wireless sensor networks", College of Information Technology, United Arab Emirates University, Al Ain, UAE.

[9] Ted Tsung-Te Lai, Wei-Ju Chen, Kuei-Han Li, Polly Huang, and Hao-Hua Chu, "Triopusnet: automating wireless sensor network deployment and replacement in pipeline monitoring", National Taiwan University, Taipei, Taiwan.

[10] Ivan Stoianov, Lama Nachman, and Sam Madden, "PIPENET: A wireless sensor network for pipeline monitoring".

[11] Microcontroller-based experimental setup and experiments for SCADA education:-

http://ieeexplore.ieee.org/

[12] Atmega2560:

http://www.nexrobotics.com/products/microcontrollerdevelopement-boards/atmega2560-development-board.html

[13] Atmega2560 datasheet:-
http://www.atmel.in/Images/doc2549.pdf

[14] General information regarding Vijeo Citect v7.2:-
http://www.schneider-electric.com/products

[15] MODBUS reference:-
http://www.modbus.org/

[16] MODBUS reference:-

http://www.simplymodbus.ca/