International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 4 Issue 07, July-2015

Implementation of Modbus Protocol on Distributed Supervisory Control and Data Acquisition for Industrial Process

Suman Sangolli
Department of Computer Science
BVBCET, Hubli, India

Dr Vishwanath. P. Baligar
Department of Computer Science
BVBCET, Hubli, India

***Abstract* - Industrial process development uses distributed SCADA for controlling and monitoring. SCADA is software oriented process permits to monitor all the data at the field and control the process. This paper concentrates on the description of distributed SCADA, the Modbus protocol, the overview of SCADA at workstation its features and the smart SCADA application features and interfacing with hardware process for remote industrial process.**

Keywords—Modbus, SCADA [4], smart SCADA

I. INTRODUCTION

Most of the industries use Modbus protocol as network protocol for many industrial purposes. The industrial purposes incorporate manufacturing, production, power generation, fabrication, and refining. Modbus protocol [1] is simple, open source, worldwide and easy to use. This protocol is used by SCADA in which it communicates with the devices on same network and submits the outcome to the supervisory computer. Distributed SCADA is a concept i.e. controlling elements are not centralized in location but distributed, in which the SCADA at workstation and smart SCADA are the main roles of distributed controlling elements. Both the SCADA at workstation and smart SCADA are the supervisory computers and use Modbus protocol to communicate with devices over network. SCADA system monitors the acquired data from the field devices and provides remote control for the fields devices. In case of situations of alertness, the system will alert the SCADA at workstation and the necessary control is provided to the device.

The smart SCADA is a remote access and control supervisory computer similar to SCADA on workstation but with critical features. It can be connected to Modbus network remotely and is helpful as stand by for workstation. Distribution of intelligently control, analyze and monitor lets each end to work efficiently and no congestion of data on single master. It enables high availability, high reliability, scalability and multi processing.

II. FEATURES

The distributed SCADA is a software featured system. It is featured to importantly control the monitored information acquired from the field devices. It gives information regarding the status of the devices remotely located. The main feature it is oriented to is the field are monitored and controlled remotely either at workstation or with the help of smart devices (android phone).

A. Smart SCADA features

The Smart SCADA is featured with a user-friendly GUI to connect the Modbus network over TCP/IP using Wi-Fi/internet. It allows the user to acquire real time data from the field devices, the real time data is represented with help of GUI i.e. indicators, meters and sliders. The field devices can be controlled with help of help of GUI buttons. It can feed values in to the devices. The user can switch between the different slaves by providing the IP address of other slaves. It also allows to user to acquire data from all devices connected in field by providing the addresses the devices establish connection with the SCADA system.

B. SCADA at workstation features

The SCADA at workstation provides an entire controlling and real time trending data acquisition. It has an easy and user-friendly user interface. Real time historic trending of all alarms and events. Provides health status. The user can visualize the data in different pdf formats and also control the field devices operations. The visualized pdf formats and other information can be printed. The SCADA at workstation is authenticated with username, password and role of operator i.e. administrator/control operator/others. Based on type of user the features of software are accessible.

III. OVERVIEW OF MODBUS PROTOCOL

Modbus is a resource for communication between many devices over single channel. It was founded by Modicon in 1997. It is a messaging protocol at application layer in the OSI model. It provides client/server communication between the devices deployed on different networks. The Modbus protocol communicates between one master and many slaves. In MODBUS the client role is provided by the Master of the serial bus and the server's role by the

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

slave nodes. The master node that requests explicit information to one of the slaves and processes the responses. A Slave will not transmit data without a request from the master and do not communicate with other slaves. Each slave must be addressed uniquely ranging from 1 to 247.

A. Modbus protocol principle

A master can communicated with many slaves maximum up to 247 slaves. The communication commencement can be made in two ways. Firstly the unicast communication in which a master establishes communication with a single slave. The slave processes the request from master and responds to the master. The other way is broadcast communication in which a master requests all the slaves. The broadcast messages are necessarily the write commands and all the slave need to accept , no respond is given to the master on this request. The address 0 is used to address the salve for broadcast communication.

B. Modbus implemntation

Modbus is presently implemented in the following ways:

i. TCP/IP over Ethernet – Modbus TCP/IP client and servers are connected to TCP/IP network.
ii. Asynchronous serial transmission - The transmission media used are wire (RS-232, RS-422, RS-485), fiber, radio, etc.
iii. Modbus PLUS, a high speed token passing network.

In this paper we concentrated on implementation of Modbus TCP/IP over Ethernet [3]. Unlike the Modbus over serial transmission that used three layer model, the Modbus TCP/IP is featured for five layer internet model as shown in table 1.

Layers	OSI Functions	Modbus Functions
5,6,7	Application	Modbus Application protocol
4	Transport	Transmission Control Protocol
3	Network	Internet Protocol
2	Data Link	IEEE 802.3
1	Physical	IEEE 802.3

The messaging protocol involves a request sent by client to initiate a transaction and indication is sent by server to client on confirmation of the request. A response is sent by server to client for the requested information and client sends an acknowledgement to server on receiving the response. The messages between client and server uses the Modbus TCP/IP Application Data Unit (ADU) as shown in figure 1.

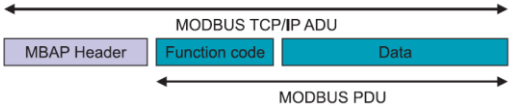

Figure.1- Modbus TCP/IP Application Data Unit

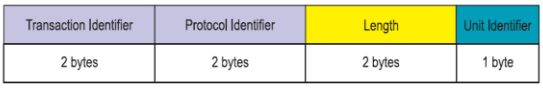

Figure.2 – MBAP header

The Modbus ADU is made up of a MBAP header as shown in figure 2, the function codes are used to address which operation to be performed and the data. The header consists of transaction identifier is provided by the client to keep track of the requests, the protocol identifier specifies the protocol used and supports multiple protocols, Modbus protocol identifier is 0 and the length in header length of remaining inclusive of PDU fields. The unit identifier gives the modbus slave identifier. The function code characterizes the type of message and it is one byte in size. They range from 0 to 255, when a slave responds it mentions the function code requested by master. When an error is detected, the highest bit of the function code is turned on. The commonly used function codes are 01 for read coil status, 02 for read input status, 03 for read holding register, 04 for read input register and 05 for write single coil.

IV. ARCHITECTURE

The system is built on distributed concept basis. This distributed structure enables great efficiency, remoteness and reliability for the industrial processes. The architecture segregates the system into well defined subsystems to attain flexibility to move for functionality from one hardware entity to other. Each subsystem has its own independent design. The SCADA system has both hardware and components. The hardware SCADA system acquires data and feeds the SCADA software system. The software SCADA system processes the data and presents it in the appropriate manner. The hardware and software SCADA systems communicate with the assist of the network using Modbus protocol.

The software SCADA system comprises two control access points, the SCADA at workstation and the smart SCADA. It consists of database to load the monitored data for future access. The devices at field and Modbus RTU form a part of hardware SCADA architecture. The software SCADA analysis the data fed from the hardware units and processes it. The SCADA system is configured in client server architecture. The SCADA at workstation/Smart SCADA is the client and the supervisory system forms the server which acts as a communicator between client and the field devices.

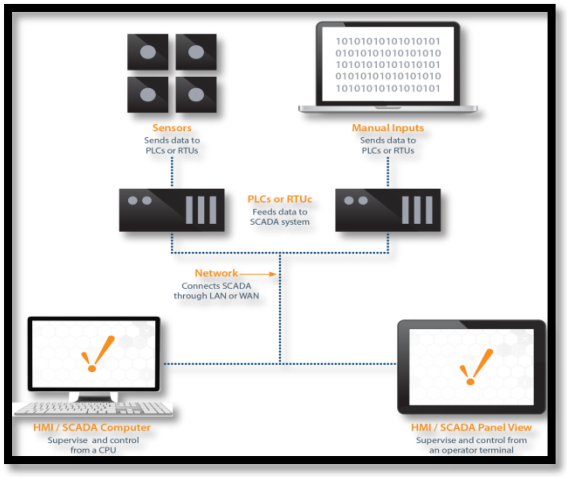

Figure.3- Overview of SCADA architecture

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

The supervisory system includes distributed software applications, disaster recovery sites and multiple servers. The Modbus RTU to Modbus TCP converter interfaces with devices and the supervisory systems. In the above figure.3 the HMI/SCADA computers and the panel view, form the SCADA software components and the RTU and sensors form the SCADA hardware devices. The PLC/RTU's are the Modbus RTU implementing the Modbus protocol. The Modbus Master sends a request to read data from the Modbus slaves. The field device Sensors sense the change in data and transmit it to the PLC/RTU's. The SCADA computer/HMI allows the clients to appropriately supervisory discussions observing the processed data and control the system. The communication network is established through LAN or WAN in SCADA systems and protocol that SCADA works on is Modbus to receive the data at workstations through Modbus TCP/IP [2]. It can utilize wide range of both wired and wireless medium for remoteness, connection availability at remote sites, polling and data rates.

V. Implementation of Smart SCADA

The Smart SCADA is divided into subsystems for independent and flexibility of the sub system enhancements. It provides an interface for the analog IO's and digital IO's. The five subsystems that it comprises of are the connection, input status, holding register, input register and coil status. The client initiates the request for connection to the server. The connection is open for a server i.e. for each request from the client a connection is not established. A client is open to only one server at a time. Many transactions can be carried out on one connection simultaneously. Once the connection between client and server is established the client demand for request of data, this data is encapsulated into Modbus request format and sent. At the server request is received and examines if the Transaction Identifier doesn't refer to any MODBUS pending transaction, the response must be discarded. If the Transaction Identifier refers to a MODBUS pending transaction, the response must be parsed in order to send a MODBUS Confirmation to the client.

The connection subsystem establishes connection to Modbus TCP/IP by providing the valid Modbus IP address, slave identifier and 502 port. It pre-fetches the previously established connection parameters and a provision to establish new connection with other slaves. Configuration settings need to be applied by the client requesting for the connection with respect to the slave devices. The input status subsystem is an interface for the digital inputs. It communicates with the server and provides the requested information to clients. It is a read-only system. The slave devices used for digital inputs are fire detectors and switch. The holding register sub system is an interface for the analog outputs. It is read and write system which accepts input from the client requests and gives output to client on request. The coil status sub system is an interface for the digital outputs using which can read and write requests to the salve.

VI. Application of Smart SCADA to harware devices

The Smart SCADA is used as a controlling and monitoring application of the systems. It serves as data gainer and presents the processed data it to the client. The processed data is the conversion of fetched data into required conversed data for further usage. The Smart SCADA is interfaced with two screens one for configuration setting needed to setup connection with the server and the other screen for the study of data from the devices and controlling of the data into the devices. The SCADA products that were evaluated decompose the process in atomic parameters (e.g. a power supply current, its maximum value, its on/off status, etc.) to which a Tag-name (e.g. input status, coil status, holding register and input register) is associated. The Tag-names used to link graphical objects to devices can be edited as required.
The Smart SCADA application is deployed on an android phone and is interfaced with to the devices to communicate over Modbus TCP/IP network. The potentiometer is used to study the analog input voltages supplied to the system. The indicator lights are used for digital outputs which indicate the states of lights 0 for off and 1 for on and Smart SCADA provides two buttons for the indicator lights to turn on/off. An off state of indicator turns the button to yellow and green on state. The switch is used to provide the digital inputs to the system and indication of switch on state indicates a green symbol and off state indicates a red symbol on the Smart SCADA application. The multi-meter is used to study the voltage across the analog output. A slider indicator depicts the multi-meter value. The application is tested using modbus simulator (ModSim32). The Smart SCADA application is implemented for Modbus master and the simulator is the Modbus slave. The master connects to the slave over TCP/IP using slave id 1 and IP address of the slave device as shown in figure 4.

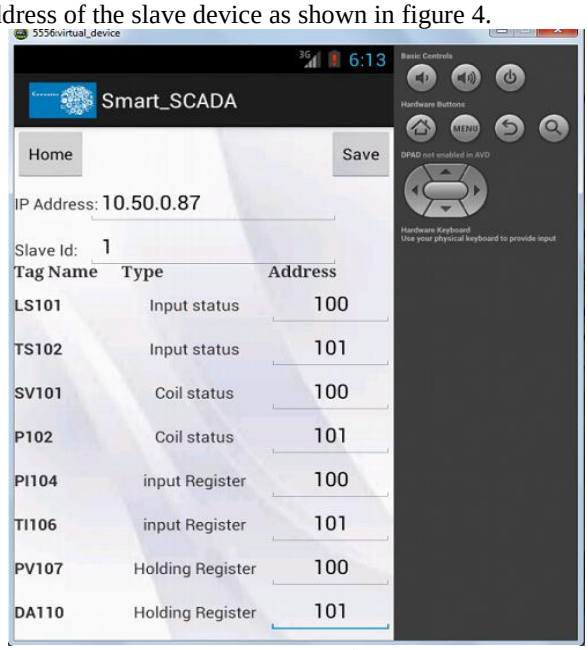

Figure.4 – Smart SCADA configuration settings

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

In figure 5, the Smart SCADA depicts the devices status values and provides the client to control the slave slaves. It shows the initial state of the application where all the parameters of the tag names are set to zero. And the connection is established with the slave of IP address 10.50.0.87 and id 1 i.e. modsim32 simulator.

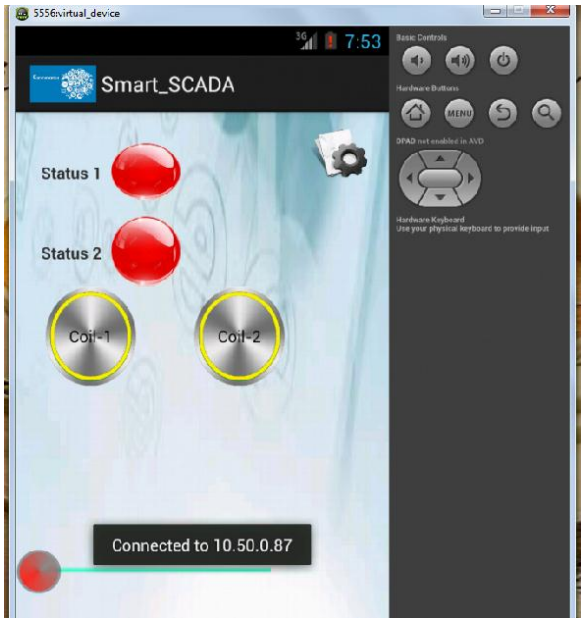

Figure.5 – Smart SCADA

ModSim32 is a testing device for confirmation of correct protocol operation in new systems. It is a designed to operate as Modbus slave, simulating data which may be accessed from a Modbus master application. ModSim32 can support direct serial connections to the master application, or operate as a network server to supply data to multiple modbus/TCP-compatible client applications. It is simulates data as shown below in the figure 6, which is connected to the Smart SCADA master application.

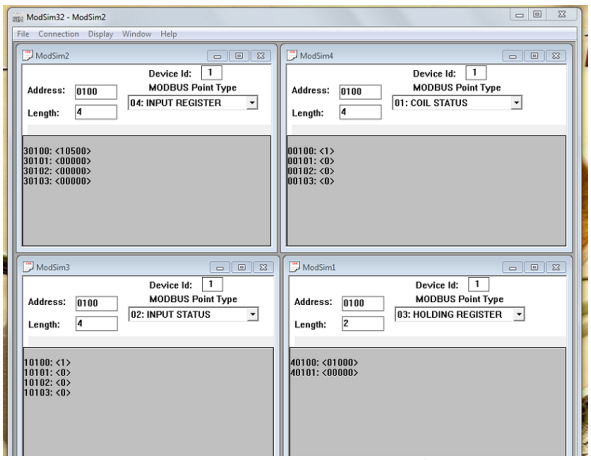

Figure.6 – ModSim32 (Slave interface)

In figure.7, the Smart SCADA application status1 and status2 read the input status values of the slave and turn it on if simulator value is 1 and the coil 1 and coil2 are to control the coil status, as coil1 is on the simulator value is 1. And the slider indicates the holding register value.

Figure.7 – Smart SCADA monitoring and controlling the Modsim32 slave.

VII. Conculsion

The distributed SCADA is designed to control and acquire data from the field devices. The software SCADA at workstation has fully fledged functionalities with animated meter to indicate the values (power, voltage, current) and controlling unit to set the set point for closed looped systems. The software Smart SCADA is equipped to monitor and control only certain critical field devices. In future this Smart SCADA is looking forward to be designed and deployed over iOS, so that it can support wide range of users and enhance compatibility. The functionality is to be increased to four parameters per tag names for additional device access. The greater fast access of data is to be featured.

VIII. Acknowledgment

Thankful to Sunlux Technologies, Bangalore , Ram V. Kerur (Managing Director & CEO), Lakshmi Prakash (Asst. Manager-Software Development& Mentor) and Abhinav A. Kalamdani (VP-Projects & Prodoucts) who inspired and guided throughout the project and also thankful to Dr Vishwanath Baligar (Pg Co-coordinator Computer Science and Engineering department, BVBCET – Hubli & guide) for guidance and devoting their precious time.

IX. References

[1] Modbus, I. D. A. "Modbus application protocol specification v1. 1b." *North Grafton, Massachusetts (www. modbus. org/specs. php)* December 28, 2006.
[2] Modbus, I. "Modbus messaging on tcp/ip implementation guide v1. 0b." *North Grafton, Massachusetts (www. modbus. org/specs. php)* October 24, 2006.
[3] George Thomas, "Introduction to modbus serial and modbus TCP", Contemporary control systems.Inc Vol. 9 Issue 5, September-October 2008.
[4] Supervisory Control and Data Acquisition (SCADA) System for Power System Applications [LITD 10: Power System Control and Associated Communications], IS 15953 (October 2011)

(This work is licensed under a Creative Commons Attribution 4.0 International License.)