ERJ
Engineering Research Journal
Faculty of Engineering
Menoufiya University

A PROPOSED NEW SCADA SYSTEM FOR REMOTE PV APPLICATIONS

Yousry Atia[1], Mohamed Zahran[1] , and Ahmed Abulmagd[2]

1. Electronics Research Institute, PV Dept., NRC Building, 33 El-Tahrir St., Dokki, 12311-Giza, Egypt (Tel. +202-33310512, Fax. +202-33351631)

2. Minya University

mbazahran_2007@yahoo.com, yousry_atia@yahoo.com, abulmagd@gmail.com

Abstract

- The remote photovoltaic systems needs a simple, cheap, remote monitoring and remotely controlling devices. Supervisory Control and Data Acquisition (SCADA) systems based on PLCs are very complex, lower reliability, and require developed electronic industry in order to produce sophisticated PLC's. Information Technology (IT) industry requires simpler, cheap, no restrictions on software or hardware SCADA systems. This paper presents a new generation of Remote Terminal Unit (RTU) for SCADA system based on microcontroller for customer side distribution automation system. We have a common trend of attempting to lower SCADA costs on RTU side. Our paper goals are to go to deep in lowering the cost of RTU unit, freeing the software so lowering system cost, and to expand the open source technology culture away from the restricted one of the large companies. A proposed microcontroller-based SCADA system with an open source software and graphical user interface is introduced in this paper. Our proposed system cost lowering efforts depending on the simple remote terminal unit (RTU), and an open source software. The system is modular where the main terminal unit (MTU) with a human machine interface (HMI) can access many RTUs that can plug and play. The proposed system is designed, implemented and gave excellent results in collecting data, transmitting, monitoring, and applying system control as well. One of the most appropriate applications for SCADA is the remote area photovoltaic standalone systems. The SCADA is used as monitoring, control and management system.

تحتاج نظم الخلايا الشمسية البعيدة عن العمران إلى وسائل نقل وعرض بياناتها وكذلك التحكم فيها عن بعد بحيث تكون هذه الوسائل بسيطة ورخيصة وبعيدة عن التعقيد. ولوحظ في نظام سكادا (Supervisory Control and Data Acquisition , SCADA) المستعمل حاليا و المبني على أساس الحاكم المنطقي المبرمج (Programmable Logic Controller, PLC) أنه معقد جدا وغالي الثمن وله درجة إعتمادية منخفضة ويحتاج إلى صناعة الكترونية وتكنولوجيا متقدمة تحتكرها الشركات الكبيرة . ومما يساعد نظم تكنولوجيا المعلومات على الانتشار والتوطين عدة أسباب نذكر منها أن يكون النظام المستعمل بسيط ، رخيص الثمن ، ولا يوجد قيود على المكونات الرئيسية ولا على البرمجيات المستخدمة معها. وتقدم هذه المقالة نظام مقترح يصلح كنواة لجيل جديد من نظم سكادا لا تعتمد على الحاكم المنطقي المبرمج كوحدة طرفية أساسية انما استبدلناها بحاكم دقيق (Microcontroller). وقد اتخذنا اتجاها في هذه المقالة لتخفيض ثمن نظام سكادا الكلي بطريقتين وهما العمل على تخفيض ثمن الوحدات الطرفيه (RTU) بجعلها تعتمد على ميكروكونترولر رخيص الثمن والعمل على نشر ثقافة البرمجيات المفتوحة المصدر (Open Source Software) والتي يمكن لأي مستخدم تطويرها ونشرها بدون الحاجة لترخيص أو شراء هذه البرمجيات. وقد تم بناء نموذج معملي مكون من وحدتين طرفيتين تعتمد على النظام المقترح استخدمنا فيها الميكروكونترولر. و قد استخدمنا برمجيات مفتوحة المصدر بعد تطويرها لبرمجة النظام وبناء نظام تفاعلي مع المستخدم (HMI) لسهولة البرمجة والتعامل مع الوحدات الطرفية والرئيسية. وقد حرصنا في التصميم أن يكون النظام المقترح نمطي وقابل للتكرار والامتداد بحيث أن الوحدة الاساسية والنظام التفاعلي تستطيع التعامل مع عدد كبير من الوحدات الطرفية والتي تعمل بنظام التوصيل والتشغيل (Plug &Play) . وتم تصميم وتنفيذ واختبار النظام المقترح بنجاح على نظام معزول محاكي لنظم الطاقة الشمسية، وقد أعطى نتائج ممتازة في تجميع وعرض البيانات المختلفة من الوحدات الطرفية وكذلك ارسال اشارات التحكم بنجاح عبر النظام للتحكم في الآلات المتصلة بتلك الوحدات.

Keywords*: SCADA, Microcontroller, Ethernet, MODBUS, Remote Monitoring, MTU, RTU, HMI, GUI,*

1. INTRODUCTION

Many literatures have reported that, as it is the case in many other information technology (IT) branches, the SCADA area lacks of formal academic theory coverage. This is why the definition of SCADA systems varies in different literatures [1]. For examples the definition of SCADA in some literatures as industrial measurement and control system consisting of a central host or master (usually called a client, master station, master terminal unit or MTU); one or more field data gathering and control units or remotes (usually called servers, remote stations, remote terminal units, or RTU stations); and a collection of standard and/or custom software used to monitor and control remotely located field data elements.

Other definition of SCADA systems are often described in definition as SCADA are a type of industrial control system used to collect data and exercise control from a remote location. In the pipeline industry, SCADA systems are used to collect data from pipeline sensors in real time and display these data to humans (controllers) who monitor the data from remote operate pipeline control equipment, such as valves and pumps. Also SCADA is a process control system that enables a site operator to monitor and control processes that are distributed among various remote sites.

In all the installed SCADA systems, the data servers communicate with devices in the field through programmable logic controllers (PLCs). PLCs are connected to the data servers either directly or via networks or fieldbuses that are proprietary (e.g. Siemens H1), or non-proprietary (e.g. Profibus). Data servers are connected to each other and to client stations via an Ethernet LAN.

From the SCADA systems development and production point of view, those systems are very complex, costly, have restricted software for his components, and requires developed electronic industry in order to produce sophisticated PLC's and to be competitive with different SCADA producers worldwide. This is quite unrealistic goal for domestic IT and electronic industry. Given the complexity of such a systems, especially highly sophisticated PLC's, they are witnessing common trend of attempting to lower SCADA costs on RTU side [1, 2].

In order to overcome this growing RTU side cost problem, they suggest PC-to-PC SCADA (PtP) concept [1]. PtP SCADA concept differs from standard SCADA concept in two main elements:

- RTU in PtP SCADA concept is obligatory PC based station unlike mostly PLC based RTU stations in standard SCADA concept, and
- RTU in PtP SCADA concept could contain elements of artificial intelligence, unlike PLC's which could not have any AI elements.

Our paper describe a new more lower cost SCADA system based on microcontrollers that making the servers (RTU) more cost effective. Using a lower cost microcontroller unit with Ethernet controller as a server for data acquisition and plant control tools will make a revolution in the SCADA world in the near future. In the proposed SCADA system not only RTUs cost are the benefits, but also the open source software that will contribute in propagation of the IT and make more cost effective systems.

1.1. SCADA SOLUTIONS FOR RENEWABLE ENERGY REMOTE AREA SYSTEMS

In recent years, because of highly global industry development, the fossil energy depletes rapidly and oil price rises quickly, so looking for alternative sources of energy becomes widespread throughout the world. The alternative sources of energy essentially means that coal, oil, natural gas and nuclear energies are excluded, which may include wind, solar, geothermal, water temperature difference, waves, tides, biomass energy and fuel cells. Among the many renewable energy resources, wind and solar are most attractive. The solar energy is clean, neither polluting the environment nor costing fuel price, and is directly obtainable. Therefore the researches and applications of solar energy become popular among governments, research institutes, organizations and even the general public’s [16]. In this paper, the design and implementation of a SCADA as a management and control of remote area standalone photovoltaic solar systems are studied.

2. SCADA IMPORTANCE AND ADVANTAGES

Telemetry is automatic transmission and measurement of data from remote sources by wire or radio or other means. A properly designed SCADA system saves time and money by eliminating the need for service personnel to visit each site for inspection, data collection/logging or make adjustments. Real-time monitoring, system modifications, troubleshooting, increased equipment life, automatic report generating SCADA is the combination of telemetry and data acquisition. SCADA system is responsible of collecting of the information, transferring it to the central site, carrying out any necessary analysis and control and then displaying that information on the operator screens. The required control actions are then passed back to the process [3].

The prevalence of SCADA systems has grown to the point that national infrastructure today depends, to a large degree, on SCADA systems. SCADA system capabilities have evolved from that of simply replacing lights and push buttons to handling very complex process control and critical safety shutdown systems. The intelligence of SCADA systems has advanced to the point where their automated operation requires fewer operators: less human supervision than previous control system methodologies. [4]

The main advantages of using SCADA are [1]:

- Real-time monitoring,

- System modifications,
- Troubleshooting,
- Increased equipment life,
- Automatic report generating, etc.

Moreover, SCADA systems provide industry with many additional secondary advantages. Good SCADA systems today not only control processes but are also used for measuring, forecasting, billing, analyzing and planning.

3. SCADA HARDWARE ARCHITECTURE

Standard SCADA system block diagram is shown on **Figure 1**.

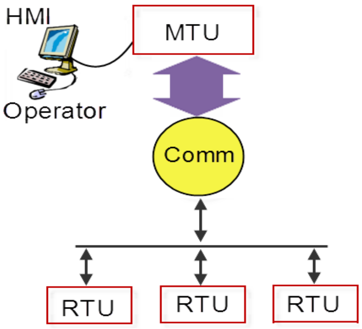


Figure 1, Block diagram of standard SCADA system.

Diagram shows three main components, Master Terminal Unit (MTU), Remote Terminal Unit (RTU) with a communication means between them, and Human Machine Interface software (HMI) [3, 4].

3.1. MTU

A central host or master (also called a master station, master terminal unit, or MTU). The master station itself often consists of two separate subsystems:

- A workstation which processes all the information and where the system operator can monitor and control the system.
- A communication controller, which takes over the burden of real-time, low-level communication processing and network management, relieving the workstation for other tasks.

3.2. RTU,

One or more field data gathering and control units or remotes (also called remote stations, remote terminal units, or RTU's). An RTU provides intelligent I/O collection and processing, such as reading inputs from switches, sensors, and transmitters, then arranging the representative data into a format that the SCADA system can understand. RTU also converts output values provided by the SCADA system from their digital form into that which can be understood by field-controllable devices such as discrete (relay) and analog outputs (current or voltage).

3.3. HUMAN MACHINE INTERFACE (HMI)

HMA is software used to monitor and control the whole system. Contemporary SCADA systems have mostly open loop control characteristics and utilize long distance communication with relatively low data rates. HMI is the means by which the user (operator) interacts with the SCADA system. Simply put, the HMI provides a clear and easy-to understand computer representation of what is, in fact, being controlled or monitored by the SCADA system.

The HMI allows the operator to view virtually all system alerts, warnings, and functions as well as change set points, analyze, archive or present data trends. A good and effective SCADA HMI design makes the interaction with the SCADA through the HMI seem natural to the operator, or in other words, clear and easy to understand, with no need for explanation.

3.4. COMMUNICATION BETWEEN MTU & RTU

The communication between MTU and RTU of the system may be wired or wireless, via local LAN or via the internet.

- The first method for private data transmission includes wire lines or buried cable and modems, and is usually limited to low bandwidth. When it makes sense to string or bury your own communication cable between sites, consider the staffing requirements necessary to support the technical/maintenance aspects of the system.
- The second method to consider is wireless transmission and includes Spread Spectrum, Microwave or VHF/UHF radios. VHF/UHF radio (good for up to 30 miles) is an electromagnetic transmission with frequencies of 175MHz-450MGz-900MHz received by special antennas.
- If you need a 24-hour permanent connection for analog data transmission between two or more locations, the Private Leased Line (PLL) should be considered. The Digital Data Service (DDS) with the Digital Subscriber Lines (DSL) and Integrated Service Digital Network (ISDN) should be considered for high speed/low error rate applications.

4. TYPICAL LAYERS OF SCADA

One distinguishes two basic layers in a SCADA system: the "client layer" which caters for the human machine interaction and the "data server layer" which handles most of the process data control activities. **Figure 2** shows the typical SCADA system layers [5].

The data servers communicate with devices in the field through process controllers. Process controllers, e.g. PLCs, are connected to the data servers either directly or via networks or fieldbuses that are proprietary (e.g. Siemens H1), or non-proprietary (e.g. Profibus). Data servers are connected to each other and to client stations via an Ethernet LAN. The data servers and client stations are NT platforms but for many products the client stations may also be Windows machines.

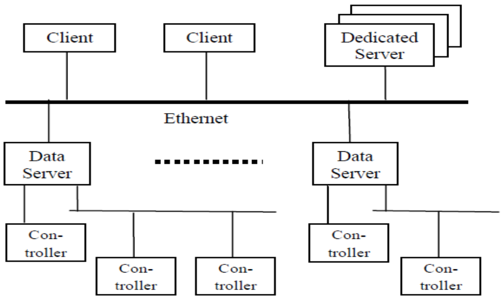


Figure 2, Typical Hardware Architecture

5. COMMON SCADA ARCHITECTURES

Early SCADA systems used proprietary event–driven operating systems and some form of rudimentary/proprietary serial communications typically based on R232/RS422/RS485. Today, SCADA system components utilize purpose built event-driven operating systems, commercial operating systems (for example, Windows/Linux as well as hybrids), and commercial operating systems with real-time extensions. While SCADA systems for critical processes are available with fault-tolerant networking, most have evolved to take advantage of UDP/TCP over Ethernet communications. In a SCADA network Figure 3, field devices are typically connected to the PLCs across an independent network using specialized protocols such as Fieldbus, HART, or MODBUS [15].

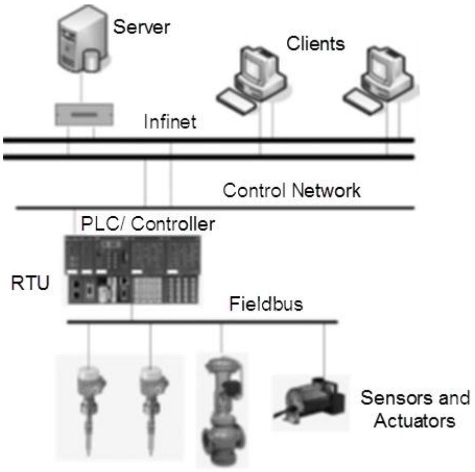


Figure 3, A SCADA Network

The PLCs are on a separate network and Communicate to the system's HMIs, hybrid controllers, and event loggers using protocols such as InfiNET. If the preceding example were the extent of connectivity to the SCADA system, they would be relatively secure. However, it is common today Figure 4 for the corporate IT network to typically connect to the InfiNET network with a bridge to allow for the collection of production data, and SCADA vendors are typically allowed to connect to PLCs or hybrid controllers in order to facilitate vendor support of the SCADA system.

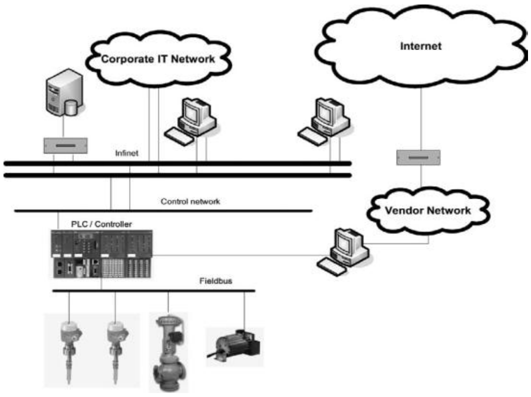


Figure 4, A Typical Network Bridge.

6. PROPOSED SCADA SYSTEM

The proposed SCADA system agrees with the idea presented in [2] since the main disadvantages of PLC- based SCADA system are:

- The system is more expensive, and don't applicable to the small projects,
- It has restricted software and hardware,
- The system is more complicated than the sensor to panel type,
- Different operating skills are required, such as system analysts and programmer,
- With thousands of sensors there is still a lot of wire to deal with,
- The operator can see only as far as the PLC,

In addition to the previous items, PLC SCADA system has complexity of system hardware and dependent modules leads to low reliability.

In [1] the authors close their modification to replace the traditional PLC based RTU to PtP RTU. This modification is very well from the cost point of view while it may some disadvantages from reliability and modularity point of view.

In our proposed system, we have used the Atmel microcontroller with their more advanced capability of computational, multiple I/O, built in A/D, timers, counters, external and internal interrupts. The microcontroller is simply a single chip microcomputer, so it has most of the computer capability in addition; it is portable, modular, compact, industrial level component and many other well-known advantages. In the proposed system, we are not concentrating to modify the RTU only, but also we developed a human machine interface (HMI) software as a graphical user interface (GUI) based on open source software for data communications and protocols. The proposed system has the same function; same ability of PLC SCADA system while it uses commercial components, open source software, by means modification could be applicable any time without the need of any permission, any license from any authority, etc. Simply it is an open system in everything hardware and software, just knowhow of the developer team.

Referring to the architecture of SCADA systems, our proposed SCADA system is consists of three main units; Master Terminal Unit (MTU), Remote Terminal Unit(s) (RTUs), and communication protocol software. MTU is a client PC (Laptop) with open source SCADA software. Each RTU is composed of microcontroller Ethernet units (AT328 microcontroller board and Ethernet controller) serves as system server. The communication protocol software is open source SCADA software with multiple communication capability. The system is attached to an emulator board with analog and digital input and output capability as shown in **Figure 5**.

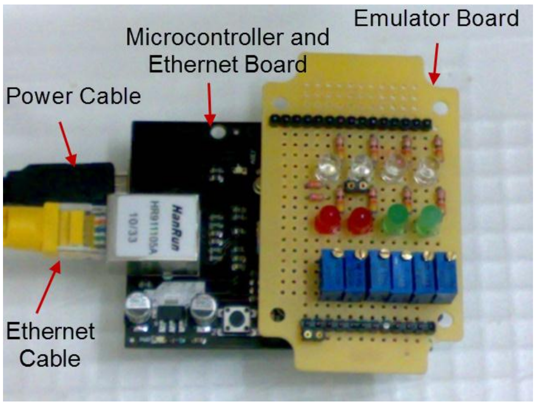


Figure 5, RTU Emulator Board.

6.1. Proposed SCADA MTU and HMI

The MTU is a client PC (Laptop) with open source SCADA software. It should be noted that many RTU's could be handled by MTU, while two RTU's only are implemented in our proposal as an example. The implemented human machine interface (HMI) software is a graphical user interface (GUI) one with an easy way to design, change, depict, and modify any of the system variables. The GUI interface menu is shown in **Figure 6** while an image of the GUI on PC monitor is shown in **Figure 7**. **Figure 8** shows the RTU_1 and RTU_2 list of variables and its declaration values.

In the GUI HMI window, left group shows the status and control signals of RTU_1 while right group shows the status and control signals of RTU_2. The upper four bars show the values of the output analog signals, matching could be found between the white LED's brightness and the high of each bar slider. Slider bar is used to control of the value of each output analog signal.

The six bars in the middle represent the analog input signals, the high of each bar changes according to the change of each analog input signal. Each analog signal value is changing from 0 to 1023 decimal as 10-bit resolution A/D is used. This reading is displaying in a text box under the displayed bar taking a color varying from green to red according the strength of the signal as shown clearly in **Figure 6**. The check boxes down shows the status of two digital output signals that controls the operation of green LED's. Each LED is illuminating according the related signal. The right two signals of each RTU block shows the status of digital input signal.

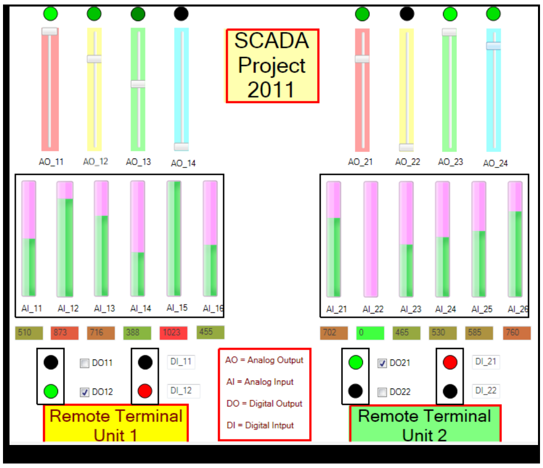


Figure 6, MTU GUI Menu.

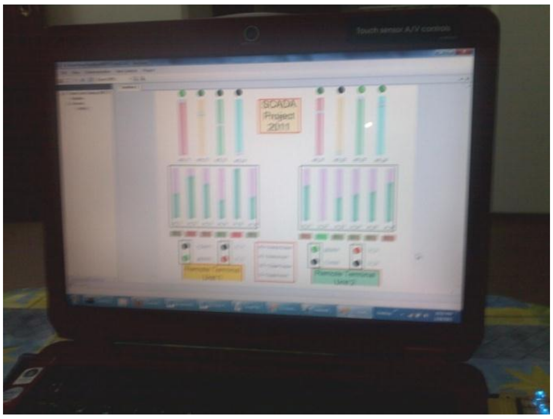
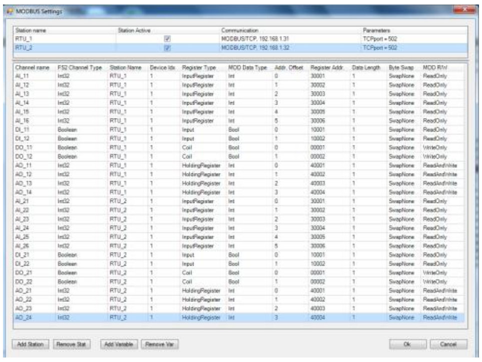


Figure 7, Image of the implemented GUI menu



Station name	Station Active	Communication	Parameters
RTU_1	☑	MODBUS/TCP, 192.168.1.31	TCPport = 502
RTU_2	☑	MODBUS/TCP, 192.168.1.32	TCPport = 502

Channel name	FS2 Channel Type	Station Name	Device Idx	Register Type	MOD Data Type	Addr. Offset	Register Addr.	Data Length	Byte Swap	MOD R/W
AI_11	Int32	RTU_1	1	InputRegister	Int	0	30001	1	SwapNone	ReadOnly
AI_12	Int32	RTU_1	1	InputRegister	Int	1	30002	1	SwapNone	ReadOnly
AI_13	Int32	RTU_1	1	InputRegister	Int	2	30003	1	SwapNone	ReadOnly
AI_14	Int32	RTU_1	1	InputRegister	Int	3	30004	1	SwapNone	ReadOnly
AI_15	Int32	RTU_1	1	InputRegister	Int	4	30005	1	SwapNone	ReadOnly
AI_16	Int32	RTU_1	1	InputRegister	Int	5	30006	1	SwapNone	ReadOnly
DI_11	Boolean	RTU_1	1	Input	Bool	0	10001	1	SwapNone	ReadOnly
DI_12	Boolean	RTU_1	1	Input	Bool	1	10002	1	SwapNone	ReadOnly
DO_11	Boolean	RTU_1	1	Coil	Bool	0	00001	1	SwapNone	WriteOnly
DO_12	Boolean	RTU_1	1	Coil	Bool	1	00002	1	SwapNone	WriteOnly
AO_11	Int32	RTU_1	1	HoldingRegister	Int	0	40001	1	SwapNone	ReadAndWrite
AO_12	Int32	RTU_1	1	HoldingRegister	Int	1	40002	1	SwapNone	ReadAndWrite
AO_13	Int32	RTU_1	1	HoldingRegister	Int	2	40003	1	SwapNone	ReadAndWrite
AO_14	Int32	RTU_1	1	HoldingRegister	Int	3	40004	1	SwapNone	ReadAndWrite
AI_21	Int32	RTU_2	1	InputRegister	Int	0	30001	1	SwapNone	ReadOnly
AI_22	Int32	RTU_2	1	InputRegister	Int	1	30002	1	SwapNone	ReadOnly
AI_23	Int32	RTU_2	1	InputRegister	Int	2	30003	1	SwapNone	ReadOnly
AI_24	Int32	RTU_2	1	InputRegister	Int	3	30004	1	SwapNone	ReadOnly
AI_25	Int32	RTU_2	1	InputRegister	Int	4	30005	1	SwapNone	ReadOnly
AI_26	Int32	RTU_2	1	InputRegister	Int	5	30006	1	SwapNone	ReadOnly
DI_21	Boolean	RTU_2	1	Input	Bool	0	10001	1	SwapNone	ReadOnly
DI_22	Boolean	RTU_2	1	Input	Bool	1	10002	1	SwapNone	ReadOnly
DO_21	Boolean	RTU_2	1	Coil	Bool	0	00001	1	SwapNone	WriteOnly
DO_22	Boolean	RTU_2	1	Coil	Bool	1	00002	1	SwapNone	WriteOnly
AO_21	Int32	RTU_2	1	HoldingRegister	Int	0	40001	1	SwapNone	ReadAndWrite
AO_22	Int32	RTU_2	1	HoldingRegister	Int	1	40002	1	SwapNone	ReadAndWrite
AO_23	Int32	RTU_2	1	HoldingRegister	Int	2	40003	1	SwapNone	ReadAndWrite
AO_24	Int32	RTU_2	1	HoldingRegister	Int	3	40004	1	SwapNone	ReadAndWrite

Figure 8, HMI SW for RTU_1 and RTU_2 variables.

6.1.1. COMMUNICATION PROTOCOLS

A protocol is a set of rules that governs how digital messages are exchanged. In the early years of SCADA systems, few if any communications standards existed, hence individual SCADA equipment vendors each created their own exclusive proprietary protocols. It has been estimated that at one point there were between 150 to 200 different proprietary SCADA protocols in use. The high number of protocols in use along with their proprietary nature actually afforded a degree of security through “security by obscurity.” As the SCADA industry matured and vendors began to adopt open standards, the total number of SCADA protocols commonly in use was reduced to a small number of popular protocols that were being promoted by industry professional organizations. In modern protocol design, protocols are “layered”. Layering is a design principle which divides the protocol design into a number of smaller parts, each of which accomplishes a particular sub-task, and interacts with the other parts of the protocol only in a small number of well-defined ways.

For example, one layer might describe how to encode text (with ASCII, say), while another describes how to inquire for messages (with the Internet’s simple mail transfer protocol, for example), while another may detect and retry errors (with the Internet’s transmission control protocol), another handles addressing (say with IP, the Internet Protocol), another handles the encapsulation of that data into a stream of bits (for example, with the point-to-point protocol), and another handles the electrical encoding of the bits, (with a V.42 modem, for example).

Table 1, Some Communication Protocols

MODBUS	Ethernet/IP
PROFIBUS	ControlNet
InfiNET	HART
UCA	Fieldbus
Distributed Network Protocol (DNP)	Utility Comm. Architecture (UCA)
Inter-Control Center Comm. Protocol (ICCP)	Telecontrol Application Service Element (TASE)

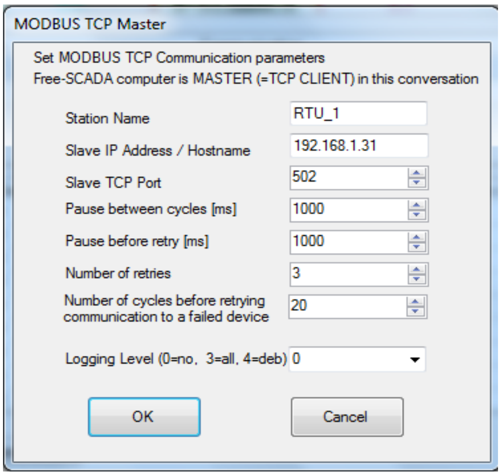
6.1.2. MODBUS PROTOCOL

Modbus is an open standard Master/Slave application protocol that can be used on several different physical Layers. Modbus is an application-layer messaging protocol. It provides client/server communication between devices connected on different types of buses or networks. Modbus-TCP means that the Modbus protocol is used on top of Ethernet-TCP/IP. Modbus-TCP is an open Industrial Ethernet network.

MODBUS is one of the most popular protocols used in the industrial world. Supporting traditional serial protocols of RS232/485/422 and Ethernet protocols allow industrial devices, such as, programmable Logic Controls (PLCs), microprocessors, microcontrollers, HMIs and meters, to use MODBUS as their communication mode [2, 3, 5, 6, 7, 8, 9]. The MODBUS standard is flexible and easy to implement. Not only intelligent devices like micro controllers, PLCs etc. can communicate over MODBUS, but also many intelligent sensors are equipped with a MODBUS interface to send their data to host systems. While MODBUS is mainly used on wired serial communication lines, there are also extensions to the standard for wireless communications and Transmission Control Protocol / Internet Protocol (TCP/IP) networks. The MODBUS specification includes two possible transmission modes, ASCII and RTU. MODBUS RTU mode is the most common implementation, using binary coding and Cyclic Redundancy Check (CRC) error-checking. Communication on a MODBUS network is always initiated (started) by a “Master” with a “query” to a “slave (RTU)”. The slave, who is constantly monitoring the network for queries, will recognize only queries addressed to it and will respond either by performing an action or by returning a “response”. Only master can initiate a transaction (query). Modbus/TCP is a Modbus/RTU message transmitted with a TCP/IP wrapper and sent over a network instead of serial lines.

6.1.3. MODBUS/TCP PROTOCOL SETTING MENUS

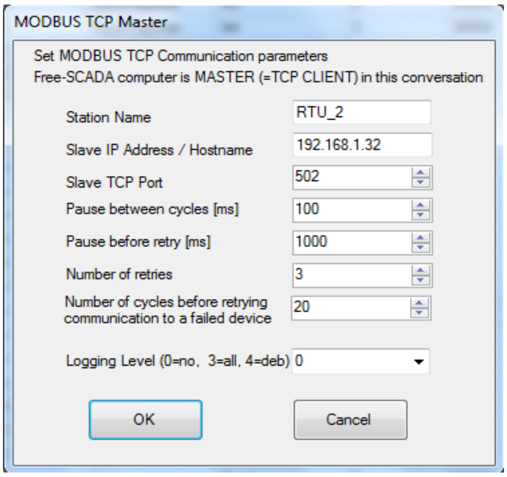
In the proposed system the MODBUS protocol is applied. **Figure 9** and Figure 10 shows the setting of MODBUS/TCP for both RTU1 and RTU2; setting and initialization.


MODBUS TCP Master

Set MODBUS TCP Communication parameters
Free-SCADA computer is MASTER (=TCP CLIENT) in this conversation

Station Name: RTU_1
Slave IP Address / Hostname: 192.168.1.31
Slave TCP Port: 502
Pause between cycles [ms]: 1000
Pause before retry [ms]: 1000
Number of retries: 3
Number of cycles before retrying communication to a failed device: 20
Logging Level (0=no, 3=all, 4=deb): 0

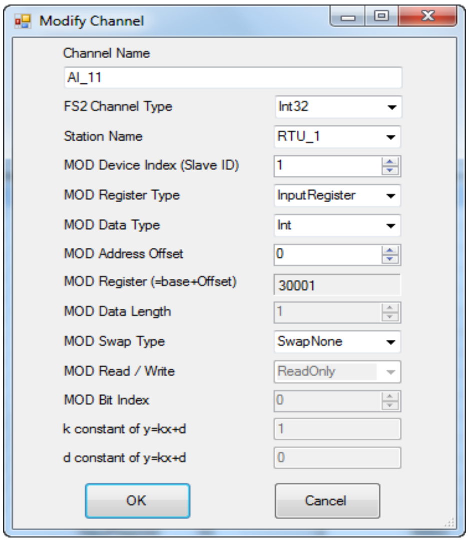
Figure 9, RTU_1, IP, Port Assignments.


MODBUS TCP Master

Set MODBUS TCP Communication parameters
Free-SCADA computer is MASTER (=TCP CLIENT) in this conversation

Station Name: RTU_2
Slave IP Address / Hostname: 192.168.1.32
Slave TCP Port: 502
Pause between cycles [ms]: 100
Pause before retry [ms]: 1000
Number of retries: 3
Number of cycles before retrying communication to a failed device: 20
Logging Level (0=no, 3=all, 4=deb): 0

Figure 10, RTU_2, IP, Port Assignments

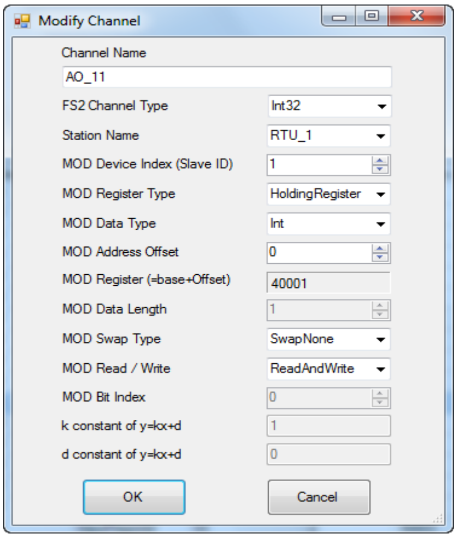

Modify Channel

Channel Name: AI_11
FS2 Channel Type: Int32
Station Name: RTU_1
MOD Device Index (Slave ID): 1
MOD Register Type: InputRegister
MOD Data Type: Int
MOD Address Offset: 0
MOD Register (=base+Offset): 30001
MOD Data Length: 1
MOD Swap Type: SwapNone
MOD Read / Write: ReadOnly
MOD Bit Index: 0
k constant of y=kx+d: 1
d constant of y=kx+d: 0

Figure 11, RTU_1, Analog input channel design menu.

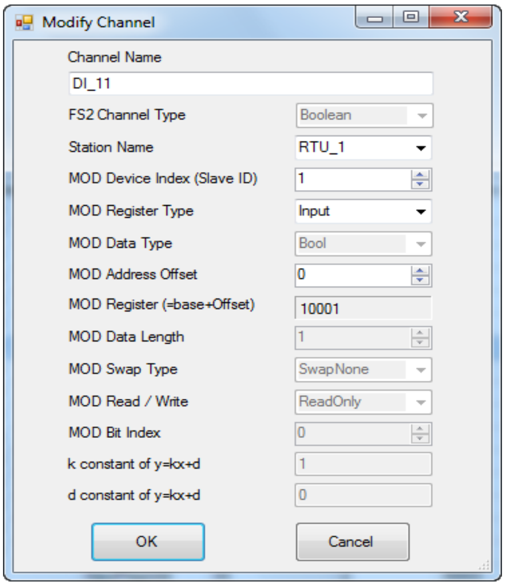
Each RTU must have its own IP and should not be repeated for other RTU in the same system. In the following four figures (11-14), the initialization menus of the analog input and output channels and the digital input and output channels for RTU_1 is given, the same is for RTU_2.

RTU-1 takes slave address of 1 and RTU-2 takes slave address of 2. In each RTU, the addresses of the analog input channels range are 30001-39999 and for analog output range is 40001-49999. Whereas the addresses of the digital input channels start at 10001 up to 19999, and for digital output start at address 1 and ends at 9999 as shown in the figures. Also in the figures we can decide that channel allocate in any RTU. The analog input is specified as read only "input register" while the analog output as "holding register" with read/write capability. The digital input is specified as read only "input", while the digital output as write only Modbus "coil".


Modify Channel

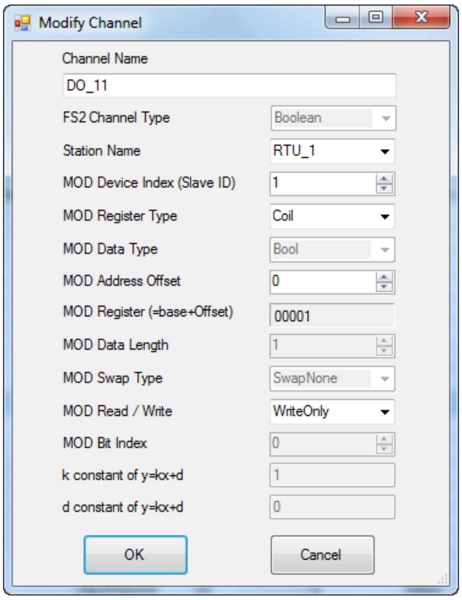
Channel Name: AO_11
FS2 Channel Type: Int32
Station Name: RTU_1
MOD Device Index (Slave ID): 1
MOD Register Type: HoldingRegister
MOD Data Type: Int
MOD Address Offset: 0
MOD Register (=base+Offset): 40001
MOD Data Length: 1
MOD Swap Type: SwapNone
MOD Read / Write: ReadAndWrite
MOD Bit Index: 0
k constant of y=kx+d: 1
d constant of y=kx+d: 0

Figure 12, RTU_1, Analog output channel design menu.


Modify Channel

Channel Name: DI_11
FS2 Channel Type: Boolean
Station Name: RTU_1
MOD Device Index (Slave ID): 1
MOD Register Type: Input
MOD Data Type: Bool
MOD Address Offset: 0
MOD Register (=base+Offset): 10001
MOD Data Length: 1
MOD Swap Type: SwapNone
MOD Read / Write: ReadOnly
MOD Bit Index: 0
k constant of y=kx+d: 1
d constant of y=kx+d: 0

Figure 13, Digital input channel design menu.



Channel Name	DO_11
FS2 Channel Type	Boolean
Station Name	RTU_1
MOD Device Index (Slave ID)	1
MOD Register Type	Coil
MOD Data Type	Bool
MOD Address Offset	0
MOD Register (=base+Offset)	00001
MOD Data Length	1
MOD Swap Type	SwapNone
MOD Read / Write	WriteOnly
MOD Bit Index	0
k constant of y=kx+d	1
d constant of y=kx+d	0

Figure 14, RTU_1, Digital output channel design menu.

6.2. Proposed SCADA System RTU

Each of RTU is composed of microcontroller Ethernet units (AT328 microcontroller board and Ethernet controller) serves as system server. The microcontroller board features as:

• High Performance, Low Power AVR, 8-Bit Microcontroller with the following capabilities:

- 32 x 8 General Purpose Working Registers
- Up to 20 MIPS Throughput at 20 MHz
- 32 Kbytes program memory, 1 Kbytes EEPROM, 2 Kbytes Internal SRAM
- Programming Lock for Software Security
- Two 8-bit Timer/Counters, one 16-bit Timer/Counter
- Six PWM Channels
- 6-channel 10-bit ADC

The microcontroller pins can be configured as input or output for flexibility. Figure 15 and Figure 16 show the proposed microcontroller and Ethernet boards.

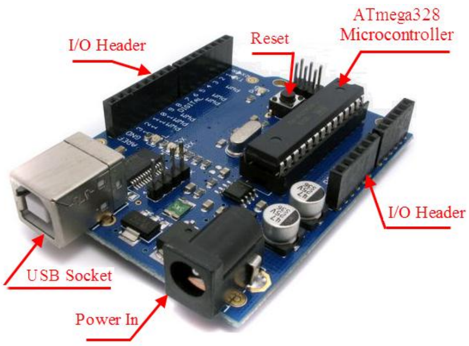


Figure 15, AT328 microcontroller board.

For prove of concept, we design and implement an emulator board (EMB). The EMB is supported with 6 analog signals (6 blue pots.), 4 analog output (detected by the brightness of the 4 white LED's, 2 digital output (2 green LED's) and 2 digital input (2 red LED's). these list of signals represents the capability of RTU itself. Figure 17 shows the designed and implemented EMB.

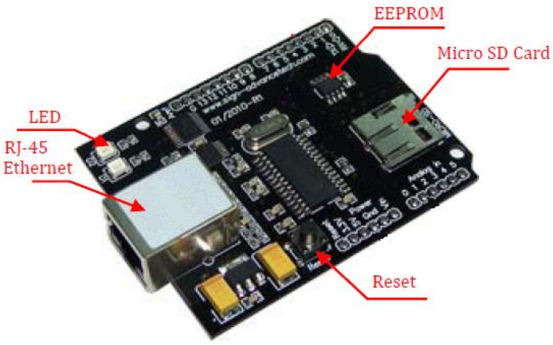


Figure 16, Ethernet Controller Board.

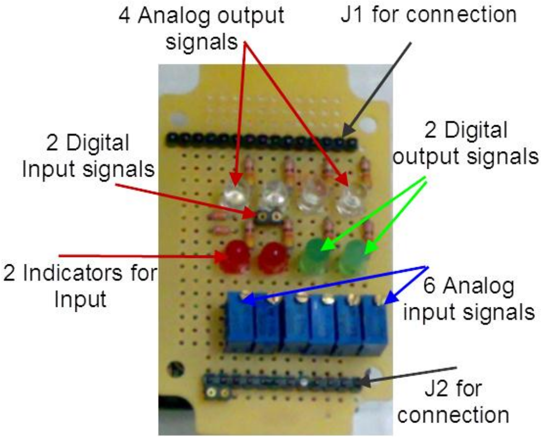


Figure 17, The implemented emulator board (EMB).

6.3. Proposed SCADA Implementation

The RTU's of the proposed SCADA system are connected to an air station wireless high power device through an Ethernet cable. The power of each RTU (during testing) is sinking from the USB port of the Laptop computer. The data link between the RTU's and the MTU is a wireless data communication. The used air station wireless device has a moderate rang indoor and a wide range in the free area. It is well known for most of the internet users that the length of net cable could reach to 100m without affecting the signal quality and the air station wireless unit has at least 100m also [10, 11]. So, the RTU's could be distributed in a circle with ~200m radius from the MTU location.

Figure 18 shows two RTU's while connected with the Air Station Wireless High Power Module. The Air

Station is available in many models, in our case it has four Ethernet cable inputs, and by means it is able to handle four RTU's at the same time. Expansion could be made either by adding a network switch or replace the Air Station with more inputs one. **Figure 19** shows the proposed SCADA system while in the operation mode. RTU_1 and RTU_2 connected to the Air Station whereas it powered from the USB ports of the laptop. A wireless communication is done between the laptop and the Air Station.

7. Proposed System Operation Modes

The proposed system is tested in three modes of operation; modes 1, 2, and 3.

7.1. Operation Mode 1

The first mode of operation is when the Air Station Wireless High Power Module not connected to WLAN. It is in a local LAN mode only and the communication between MTU and RTU's is done locally. Another Laptop with the GUI HMI software is put in the range of Air Station Wireless High Power Module radio signal interacts, displays and controls the RTU's, **Figure 20**.

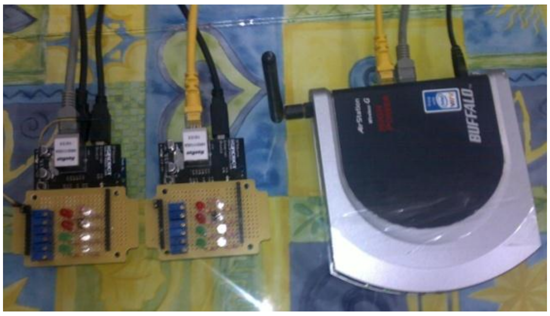


Figure 18, Two RTU's with Air Station Wireless High Power Module.

7.2. Operation Mode 2

The second mode when the Air Station Wireless Module is connected to WLAN (internet), and there is not a dynamic DNS assignment. The system is in a local LAN mode only and the communication between MTU and RTU's is done locally. In this case the two RTU's has a local IP addresses. So, communication is done locally (through the same LAN). This mode is similar to mode 1.

7.3. Operation Mode 3

The third mode is called Dynamic DNS mode. Dynamic DNS allows us to create a hostname that points to our local IP address via Internet Service Provider, making an easy-to-remember URL for quick access or access from another remote LAN. As disconnect and reconnect to the internet, the IP address is changing. So the Dynamic DNS mode allows us to track the changes in the IP as it is varied. In our Air Station Wireless Module, it can be configured using Game & Application Sharing menu, so it can be accessible using this alias rather than varied local IP address assigned by our Internet Service Provider. In this mode the Air Station Wireless Module is connected to WAN, and the IP of one RTU or both are operating in dynamic URL mode.

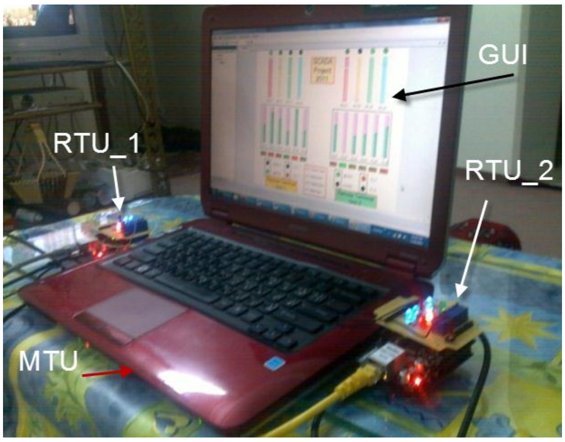


Figure 19, Two RTU's with MTU in operation mode.

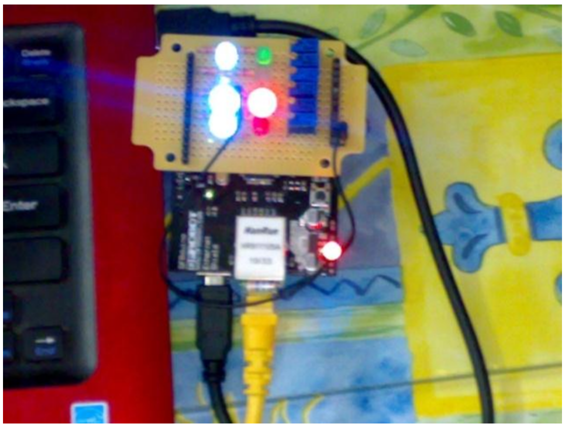


Figure 20, EMB while in operation.

8. SCADA Application in PV Systems

With the wide use of renewable energy resource (RES), traditional energy resource structure have been adjusted and modulated. Solar energy becomes ideal alternative energy of traditional fossil energy for its wealthy resource, wide distribution and availability in environmental protection. In recent years, SCADA system has been widely applied in power system substation automation and becomes a focus of electric utility. At the same time, SCADA system has been used in PV power generation area, especially in large-scale application of PV plants

[12]. The proposed SCADA system is suitable for the distributed PV plants since it is cost effective, reliable, cheaper and low power consumption.

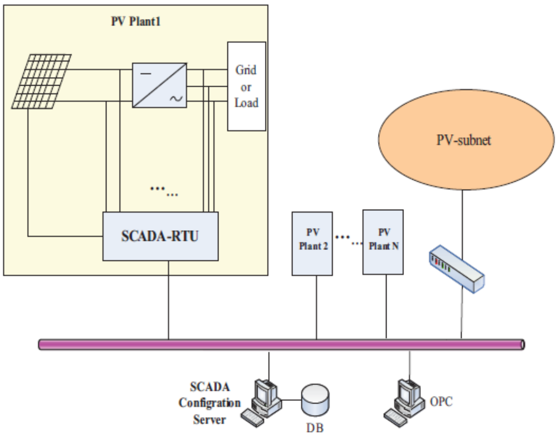


Figure 21, Distributed network of PV power plants

9. How Serious Are the Security Issues of SCADA?

Imagine living in a world where a common cold could easily kill you—Welcome to the world of SCADA. A simple Port-Scan as typically run within most enterprise networks to determine what ports are open on a SCADA component has the potential of causing the SCADA system to crash due to the

10. Conclusion

The proposed SCADA systems is simple, cheap, and reliable, plug and play, has no software or hardware restrictions. The system is microcontroller-based SCADA system that has the same function; same ability of PLC-based SCADA system while it uses commercial components, and open source software. The system composed of MTU, two RTUs, and HMI. The key of lowering system cost is RTU and open source software. The designed HMI allows of designing and/or modification of the menus to be applicable any system at time without the need of permission or license. The results show that the proposed system is more simple compared to other commercial systems, and has no restriction or required license that making rapid widespread of information technology.

11. References:

[1] Elmir Babovic and Jasmin Velagić, “Lowering SCADA development and implementation costs using PtP concept”, 978-1-4244-4221-8/09/$25.00 ©2009 IEEE.

[2] M. M. Ahmed, and W. L. Soo, “Supervisory Control and Data Acquisition System (SCADA) Based Customized Remote Terminal Unit (RTU) for

improper support of TCP/IP error handling [5, 13, 14].

Unfortunately, in order to reap the full financial benefits of a SCADA system, interconnection to the enterprise network is necessary to provide real-time process data to enterprise back-end systems. It is this interconnection of SCADA systems and the enterprise network that is the weakest link in SCADA system security. The first step in understanding the risks associated with SCADA while operating in today’s digital world is to accept that SCADA systems have always been designed to operate in closed environments. While most organizations claim that their SCADA system is not connected to their enterprise network, it has been estimated that, in reality, 80 to 90 percent of SCADA systems are in fact connected to the enterprise network. It is that connection to the enterprise network that opens the door for Internet hackers to attack SCADA systems.

“Many believe that it is the interconnection of SCADA and business systems across the enterprise network that poses the greatest risk to SCADA. In other words, SCADA systems were not initially intended to operate within the enterprise environment. At issue is the inability within SCADA components to deal with the exposure to viruses, worms, and malware that are commonplace today within the enterprise network”.

Distribution Automation System”, 2nd IEEE International Conference on Power and Energy (PECon 08), December 1-3, 2008, Johor Baharu, Malaysia.

[3] Tai-hoon Kim, “SCADA Architecture with Mobile Remote Components”, Wseas Transactions on Systems and Control, Issue 8, Volume 5, August 2010.

[4] Farkhod Alsiherov, Taihoon Kim, “Research Trend on Secure SCADA Network Technology and Methods”, Wseas Transactions on Systems and Control, Issue 8, Volume 5, August 2010.

[5] Paul A. Henry, “Supervisory Control and Data Acquisition”, www.syngress.com.

[6] Adnan Salihbegovic et. al, “Web based multilayered distributed SCADA/HMI system in refinery application”, Computer Standards & Interfaces 31 (2009) 599–612.

[7] Esmail Fathi Loshani And Maryam Sharifkhani, “An Optimum Solution for Telemetry of Distributed Wells in South of Tehran”, Proceedings of the 8th WSEAS International Conference on Signal Processing, Robotics and Automation, ISSN: 1790-5117, ISBN: 978-960-474-054-3

[8] N.Yellamandamma et. al, "Low Cost Solution for Automation and Control of MV Substation using MODBUS-SCADA", 2009 Third International Conference on Power Systems, Kharagpur, INDIA December 27-29, 2009

[9] Tai-hoon Kim, "Weather Condition Double Checking in Internet SCADA Environment", Wseas Transactions on Systems and Control, Issue 8, Volume 5, August 2010.

[10] http://fatboys.co.za/content/view/71/25.

[11] http://www.buffalotech.com/files/products/whr-hp-g54_DS.pdf

[12] Hu Guozhen, Cai tao, Chen Changsong and Duan Shanxu, "Solutions for SCADA system Communication Reliability in Photovoltaic Power Plants", 2009 IEEE 6th International Power Electronics and Motion Control Conference (IPEMC 2009), Shangri-La Hotel, Wuhan, China, 2009

[13] Farkhod Alsiherov, Taihoon Kim, "Secure SCADA Network Technology and Methods", Proceedings of the 12th WSEAS International Conference on Automatic Control, Modelling & Simulation, Catania, Sicily, Italy, May 29-31, 2010.

[14] Vinay M. Igure*, Sean A. Laughter, Ronald D. Williams, "Security issues in SCADA networks", compute r s & security 25 (2006) 498 – 506.

[15] Engin Ozdemir, Mevlut Karacor, "Mobile phone based SCADA for industrial automation", ISA Transactions, Volume 45, Number 1, January 2006, pages 67–75

[16] Jin-Maun Ho Chung-Chih Lu, "The Design and Implementation of Stand-Alone Solar Power LED Lighting Systems", 9th WSEAS International Conference on Circuits, Systems, Electronics, Control & Signal Processing (CSECS '10) ISBN: 978-960-474-262-2, Vouliagmeni, Athens, Greece, December 29-31, 2010