T.C.

ISTANBUL ALTINBAS UNIVERSITY

(Information Technology)

EVALUATION OF ETHERNET-SERIAL PROTOCOL CONVERTER FOR SCADA SYSTEM USING RASPBERRY PI

Hamzah Hameed Jasim

Master Thesis

Supervisor: Prof. Dr. Osman Nuri UCAN

Istanbul, (2019)

EVALUATION OF ETHERNET-SERIAL PROTOCOL CONVERTER FOR SCADA SYSTEM USING RASPBERRY PI

By

Hamzah Hameed Jasim

Information Technologies

Submitted to the Graduate School of Science and Engineering

In partial fulfillment of the requirements for the degree of

Master of Science

ALTINBAS UNIVERSITY

2019

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Prof. Dr. Osman Nuri UÇAN

Supervisor

Examining Committee Members (first name belongs to the chairperson of the jury and the second name belongs to supervisor)

Prof. Dr. Osman Nuri UÇAN	School of Engineering and Natural Sciences, Altinbaş University	_____
Assoc. Prof. Dr. Oğuz BAYAT	School of Engineering and Natural Sciences, Altinbaş University	_____
Asst. Prof. Dr. Adil Deniz DURU	Physical Education and Sport, Marmara University	_____

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

_____

Asst. Prof. Dr. Oğuz ATA

Head of Department

Approval Date of Graduate School of Science and Engineering: ____/____/____

_____

Assoc. Prof. Dr. Oğuz BAYAT

Director

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Hamzah Hameed Jasim

ACKNOWLEDGEMENT

I would like to offer my thanks to every one of the individuals who have upheld me all through the regularly extended periods of this voyage. I would like to thank my advisor, Dr. Osman Nuri Uçan for being my compass not withstanding when I believed I was lost and being in extraordinary part in charge of the zenith of this work. I likewise want to thank my supervisors for their supportive exhortation, which incredibly enhanced the nature of this work.

Finally, I thank this institution for hosting me during these years, securely earning its place as my home. I would also like to thank my family who has always unconditionally supported and motivated me throughout this whole process, and to whom I owe my every achievement. Also to my friends and loved ones who have always been by my side and kept me going through the hardest times.

ABSTRACT

EVALUATION OF ETHERNET-SERIAL PROTOCOL CONVERTER FOR SCADA SYSTEMS USING RASPBERRY PI

Jasim, Hamzah Hameed

M.S, Information Technology, Altinbas University,

Supervisor: Prof. Dr. Osman Nuri Ucan

Date: April / 2019

Pages: 64

The network topology of Internet of Effects is an advanced form of the existing technology. This network is a mixture of sensor networks and decision-making circuits. Distinct the existing technologies, we can have control over devices from anywhere in the world. Supervisory Control and Data Acquisition (SCADA) systems are crucial for industrial systems, according to the benefits of using these systems in increasing the efficiency of the industrial processes. However, according to the high cost and complex installation of these systems, they are designed for long lifetime, where a system may be used for decades. Moreover, Remote Terminal Units (RTUs) are significantly less expensive and may be added to an industrial organization as the organization grows. Thus, modern RTUs are being used with relatively older SCADA systems. The recent growth in the availability of computer networks has encouraged the implementation of Ethernet-based protocols in the RTUs, instead of the original serial-based communication protocols. Such update requires expensive upgrade to the SCADA hardware and software in order to be able to communicate with the modern RTUs. Thus, a protocol conversion method is proposed in

this study, using a single-board computer, Raspberry Pi. Two approaches are evaluated in the study. In the first approach, the Raspberry Pi acts as the SCADA server and communicate with the RTUs and buffer all the data locally, so that, the data requested by the SCADA is delivered immediately. In the second approach, the Raspberry Pi acts as an interpreter, where the messages exchanged between the SCADA and the RTU are only converted from one protocol to another. The approach with the highest update rate, from the SCADA's point of view, is going to be selected for the propose method. Raspberry Pi is a small single-board computer that has a variety of interfaces to connect it to the environment it is working it. Ethernet and SPI interfaces present in this device, and has been used to implement different applications using different communication mediums. In this study, the Raspberry Pi is going to be used to implement a protocol converter that allows older SCADA system to communicate with the RTUs, without the installation of expensive hardware and software expansions. Using SCADA systems, unauthorized access to valves and switches could be more tightly controlled while keeping a human in the loop; that is, human supervision and interaction were, and still are, part of SCADA systems. However, technological advances and the maturation of Raspberry Pi has pushed more of the supervisory function onto the computer systems that make up modern SCADA systems. If the device were an electrical breaker and the SCADA operator was an electric utility, then turning that switch on might overload the power systems during conversion of Ethernet serial protocol.

Keywords: Ethernet protocols, Serial conversion, Modbus, Multiplexers, Serial server and client, Evaluation metrics

TABLE OF CONTECT

Pages

LIST OF TABLES

Pages

LIST OF FIGURES

Pages

LIST OF ABBREVIATIONS

ARP	:	Address Resolution Protocol
CRC	:	Cyclical Redundancy Check
DHCP	:	Dynamic Host Configuration Protocol
DNP	:	Distributed Network Protocol
DSC	:	Data-logging and Supervisory Control
EIA	:	Electronic Industries Alliance
FTP	:	File Transfer Protocol
HMI	:	Human Machine Interface
HTTP	:	Hypertext Transfer Protocol
ICMP	:	Internet Control Message Protocol
IGMP	:	Internet Group Management Protocol
IOT	:	Internet of Things
IP	:	Internet Protocol
LAN	:	Local Area Network
LRC	:	Longitudinal Redundancy Check
MAC	:	Medium Access Control
MTU	:	Master Terminal Units
PDU	:	Protocol Data Unit
RTU	:	Remote Terminal Unit
SCADA	:	Supervisory control and data acquisition

1. INTRODUCTION

Supervisory control and data acquisition (SCADA) systems became popular in the 1960's for a variety of reasons. SCADA systems allow measurement and control of physical systems to be carried from a remote location. Initially they were used by industries and utilities to monitor and control physical devices like valves and switches. Prior to the use of Raspberry Pi , opening and closing of valves or the setting of switches was done manually; this was both costly because it was labor intensive and the exposure of valves and switches (especially in a distributed system like the electrical power grid or water supply system) to human control was considered a security and safety issue. Using SCADA systems, unauthorized access to valves and switches could be more tightly controlled while keeping a human in the loop; that is, human supervision and interaction were, and still are, part of SCADA systems. However, technological advances and the maturation of Raspberry Pi has pushed more of the supervisory function onto the computer systems that make up modern SCADA systems. In the early development of SCADA systems attention was given to physical security, but virtually no attention was given to electronic or Ethernet serial protocol security. The systems were obscure and the skills and technology needed to interact with the systems were simply not readily available; security of this type is often referred to as "security through obscurity". This pattern has continued and today "most dedicated SCADA and PCS applications have not included built-in security" [1]. Unfortunately, open protocols, advanced telecommunication networks, cheap computer electronics, and unlimited access to even the most obscure information through the World Wide Web have made SCADA's security through obscurity obsolete. The move of SCADA systems to open standards and new technology has allowed SCADA system managers to realize cost savings by using commercial-off-the-shelf (Raspberry Pi) hardware and software. In addition, as computer networks and information systems have become more commonplace throughout the corporate enterprise, managers have seen the economic benefits of having access to Raspberry Pi data and have built network connections into the previously isolated SCADA networks. The connection of porous and less secure corporate networks to once isolated SCADA networks, now using Ethernet-serial protocol systems, has unintentionally exposed SCADA systems to a host of vulnerabilities and threats for which it was ill prepared. SCADA protocols provide no authentication or authorization

capabilities. When other networks are connected to the SCADA network, intentionally or unintentionally, a converter who manages to gain access to the Ethernet-serial protocol network can spoof control signals on the SCADA network. Because SCADA protocols do not provide authentication or authorization a SCADA system is unable to distinguish between a real and a spoofed control signal, allowing the converter to control SCADA devices. If the device were an electrical breaker and the SCADA operator was an electric utility, then turning that switch on might overload the power systems, or tuning it off might tum off electricity to customers. This threat is compounded by the use of Ethernet-serial protocol software, particularly Raspberry Pi operating systems, as it becomes possible for insiders to use almost any PC to run SCADA software, and thus elevates the insider threat.

1.1. MOTIVATION

The Project purpose is to design an internet which is a network and that network used to control and monitors several devices remotely by using the internet. To achieve this, we are using the Raspberry Pi to reduce the complexity of design and in a cheap cost. Using Ethernet this design communicates with the internet, through web link to make a control of the devices possible and these can be easily accessed by the computers.

1.2. PROBLEM STATEMENT

The main problem of converting the Ethernet serial protocol using Supervisory Control and Data Acquisition (SCADA) and raspberry pi came into existence in the mid 1960's coinciding with the development of the minicomputer. SCADA provides a means for remotely monitoring and controlling many kinds of industrial systems by providing users of the system with the ability to remotely control one or more specific devices and to monitor the performance of those devices from a central and physically remote location.

- Basically SCADA, combination of elements from the hardware with software that allows organization of industrial.
- Control manufacturing at remote locations or locally processes
- Process real-time data, gather and Monitor
- Through software of Human Machine Interface, converter, raspberry pi and Ethernet serial protocol interrelate directly by devices.
- A log life of a Record Event after conversion.

An excellent example of such a SCADA system is the distribution system used by electric utilities, which is one of the oldest and most familiar SCADA systems. In electricity distribution SCADA is used to collect information from remote parts of a power distribution grid; for example, the volts, amps or phase angle of a particular line in a substation, and provide it to a central control installation. In addition, SCADA allows an operator at the centralized control station to trip breakers at remote substations in response to conditions reported by the SCADA system.

1.3. SCOPE OF STUDY

The developed and implemented functionalities for Ethernet serial protocol conversion in this work are highly demanded in industrial automation and advance networking, supervision and control systems. The results and conclusions of the study will serve to provide some ready-to-use solutions to Industrial Shields customers and thus, upgrade the value-for-money of the products. The ultimate application of this study is to provide a prototype of a SCADA system based on Industrial Shields for conversion of Ethernet serial protocol. Following are the main points that this study will cover.

- Implementation of converter for Ethernet serial protocol over (master and slave mode) for conversion of Ethernet serial protocol.
- Development of a Human Machine Interface (HMI) for supervision and control purposes.
- Implementation of TCP protocol over Ethernet and Local Area Network (LAN) for interaction between the control unit and the HMI using raspberry pi.
- Practical integration of sensors, data acquisition equipment and actuators with a raspberry pi.
- The SCADA system prototype is focused on a real - case project, which is a better for conversion of Ethernet serial protocol where an upgrading of the current supervisory and control system is wanted.

1.4. OBJECTIVE

The main objective of this work is to adapt or develop industrial communication protocols to the open-source based Industrial Shields Ethernet serial protocol conversion. A second goal of the study is to provide a prototype of a SCADA system using Industrial Shields Ethernet serial protocol conversion. Finally, the work developed should serve as a proof of concept and reference showing that developing a SCADA system using low-cost alternative hardware, based on open-source, is nowadays a feasible alternative to traditional and closed-standard automation solutions and conversion of Ethernet serial protocol conversion using SCADA systems and raspberry pi.

1.5. OUTLINE

This thesis describes the research and work developed and it is organized as follows:

Chapter 1: Aims to explore relevant studies on this field.

Chapter 2: Provides the backbone structure, requirements and the solution's architecture.

Chapter 3: Describes the implementation of the solution as well as the technologies involved in its development.

Chapter 4: Describes the evaluation tests performed and the corresponding results.

Chapter 5: Describes the developed work, and result analysis with result evaluation.

Chapter 6: Summarizes the developed work, as well as conducts the comparison of this work with all work done previously.

Chapter 7: Conclude the thesis as well as future work prediction in this domain.

2. BACKGROUND

This section provides an overview of same major contributions in this area. Section 2.2 provide an insight on some of the existing tools to perform flow analysis. Section 2.3 and 2.4 proceeds to point out some network monitoring applications that were built in a flow-based fashion. Section gives a full view of some of the most addressed network intrusions and respective works that show how to detect them using a flow-level analysis rather than payload inspection. At last, Section 2.5 and 2.6 gives a brief explanation of what RTU in SCADA systems and section 2.9 describes a machine learning is and how it can be used to achieve our goal.

2.1. COMPONENTS OF SCADA

There are four main components that make up a SCADA systems: the supervisory system or master terminal unit (MTU), remote terminal units (RTU), a communications network, and field instruments or devices [8-10]. The exact nature of the different components depends greatly on the specific SCADA system and its topology. A typical supervisory system and each subsystem is explained in detail in the following paragraphs. A small SCADA system might consist of only one MTU and one RTU, and is referred to as single-master, single-remote. A more common configuration is the single-master, multiple-remote system with a single MTU connected to many RTUs. In large SCADA systems it possible to have multiple MTUs and hundreds of RTUs.

2.2. MASTER TERMINAL UNITS (MTU)

The master station or master terminal unit (MTU) has traditionally been located in a control room where human operators interact with the system through a user interface (UI). The MTU is responsible for polling remote devices for data, processing the data, providing various representations of the data (including alarms) and sending operator initiated control signals back to the field devices. In some situations, the UI is carried out by a separate system called a HMI (human machine interface) system. The HMI system provides an interface between an operator and the MTU, freeing up the MTU from providing a UI. In this case the MTU continues to carryout polling and control activities, but the high level

representation is left to the HMI machine. A sample operator screen typical of an HMI or MTU display.

2.3. REMOTE TERMINAL UNITS (RTU)

Remote terminal units (RTUs), also referred to as remote telemetry units, are standalone systems that can acquire data from devices or equipment at the remote site, control devices or equipment at the remote site, and transfer acquired data back to a master station. RTUs are typically built to withstand the much harsher operating environments that can be associated with remote locations like a plant floor, or an electric utility substation [2]. RTUs provide four basic types of connections for interfacing with field devices: analog inputs, analog outputs, digital inputs, and digital outputs. Leads from field devices are directly connected to these interfaces on the RTU [18-21]. An RTU also includes some communications capability through a combination of serial ports, built in modems, and more recently Ethernet ports [1-3]. Other RTU components include a CPU, memory, power supply with battery backup, watchdog timer, surge protection, and real-time clock. A sample RTU specification is given in appendix A and figure 2.3 shows a generic RTU hardware configuration.

2.3.1. Communications Network

The communication network of a SCADA system connects RTUs with MTUs. Remote locations may have a communications network, like a LAN, which can be used for local inter-device communication, but this is usually not considered to be part of the SCADA communications network. Communication links take many forms including leased lines, Public Switched Telephone Networks (PSTNs), Internet Protocol (IP) based landlines, radio, microwave and even satellite. SCADA communications security has traditionally referred to error detection and error correction capabilities, and not to features such as authentication and encryption [4-6].

2.3.2. Field Equipment

At the periphery of SCADA systems are field equipment or field devices [15]. These are the actual hardware components, which effectively serve as the eyes, ears, and hands of the SCADA system. Field equipment essentially consists of sensors and actuators. Sensors

directly measure a physical condition at some remote site and actuators open, close, activate or inactivate a remote physical device. Some examples of field equipment are: voltage sensor, phase sensor, circuit breaker, relay, temperature sensor, pressure sensor, and flow control valve.

2.3.3. Methods of communication by SCADA

- Wired directly.
- Carrier line power.
- Microwave.
- Spread spectrum (radio).
- Fibers optic.

2.3.4. Overview on Reliability

- Balancing Demands and Generations
- Balancing power supply and demand
- Observe limits of thermal and monitoring the flow
- Stability limits of voltage and observe the power
- A reliable system (Plan, Design and maintenance)

2.3.5. Functions of SCADA

- Control of supervisory
- Acquisition of Data
- Database(Real-Time)
- Interface with Graphical operator
- Processing of Alarm
- Historian of DATA/Trending on strip chart

2.4. SCADA ARCHITECURE

As computer and network technology have evolved and matured, so have SCADA systems [7-8]. The evolution of SCADA systems is generally broken down into three separate successive generations [14; 15]: monolithic, distributed, and networked. The changing

architecture of SCADA systems has been a contributing factor to the Ethernet serial protocol security issues faced by modem SCADA systems.

2.4.1. First Generation: Monolithic

At the time that SCADA systems were first developed, the mainframe computer was the dominant computer technology. Networks were virtually non-existent making mainframes standalone machines. The SCADA systems of this era reflect this paradigm. They were special purpose standalone systems that were not intended to be connected to other systems and tended to be very hierarchical and centralized in nature. A standard first generation SCADA architecture. The master station in these SCADA systems was typically a single mainframe computer. A second redundant master station was usually present and shared the communications bus with the active master station. In the event of a system failure the second system could take over [23]. The lack of network technology led vendors of SCADA systems to develop solutions that allowed RTUs to communicate with the MTU mainframe often over long distances. The communication technology they developed was driven solely by this goal and in the absence of any of today's WAN protocols [27]. In general, the communication protocols developed by different vendors were lean, supporting only the minimal functionality needed to achieve scanning and control of points within a remote device. The transmission medium used to connect RTUs and MTUs lacked a high degree of fidelity, leading to communication security focused exclusively on error detection and error correction codes. In addition, each vendor tended to view their protocols as proprietary, preventing other vendors from developing equipment that could communicate using these protocols.

2.4.2. Second Generation: Distributed

The distribution of system functionality across multiple machines increased the overall processing capability of the system, but LAN technology was only capable of handling relatively short distances, typically hundreds of feet, this meant that the systems still had to be housed within a single room. Off-the-shelf LAN protocols were available, but some vendors still choose to use propriety protocols. Communication links with RTUs were largely unchanged relative to first generation systems, and in general vendors maintained

control over what hardware, software, and devices were available for a specific SCADA system.

2.4.3. Third Generation: Networked

Third generation systems are similar in many ways to second generation systems, but with one important difference, which is the move to an open system architecture instead of a vendor controlled proprietary environments [15]. Open standards have removed the limitations that proprietary protocols placed on SCADA systems and therefore make it much easier to use COTS (commercial-off-the-shelf) components to build SCADA systems. One consequence of this move has been the use of WAN protocols like TCP/IP for communication between SCADA components like master stations, RTUs, field communication equipment, and HMIs [15]. Figure 2.5 shows a typical third-generation SCADA architecture. Some advantages of internet based SCADA systems are discussed in [16]; the primary advantage cited is lower costs.

An alternative architecture for the information and communication network of power systems is proposed by Xie, Manimaran, Vittal, Phadke, and Centeno [22]. The proposed architecture includes all the traditional elements of SCADA systems. The primary object of the architecture is to provide greater reliability through redundancy, though communication security is considered as well. Redundant communication 35 channels are combined with VPN and firewall technology to provide reliable but secure communications among entities. Another next generation SCADA communications architecture is proposed by Hauser, Bakken, and Bose [23]. The proposed architecture, referred to as GridStat, is a middleware framework with API stubs that correlate with traditional SCADA functions polling, events status, and control settings. GridStat was designed to support flexible communications, making new types of controls and better situational awareness possible. GridStat also provides schemes for trust management, with the ability to approve new subscription, make routing decisions, and manage access control.

2.5. SCADA PROTOCOLS

At the heart of SCADA networks are SCADA protocols. These provide the template for communication between SCADA components, typically between the MTU and the RTU. Early SCADA systems, the first and second generation SCADA architectures discussed

previously, used proprietary protocols, but in more recent years there has been a move to open standards in SCADA protocols. RTUs are connected to MTUs by a variety of different communication channels and both the cost and availability of the communication channels has affected protocol design.

The limited bandwidth of early communication channels resulted m a very compact message format, supporting only the most basic information needed to achieve RTU to MTU communication. The structure of the basic SCADA message format. The four bit RTU address allows multiple RTUs to share a single communication channel, rather than requiring a separate communication channel for each 15 RTU [24]. The eight-bit function code specifies what operation is to be performed by the RTU. The bits following the function code are an addressing scheme that indicates the set point, control point, or data on which the operation is to be carried out. This address has no special meaning to the RTU, and it is up to the MTU and SCADA software to correctly associate an RTU address with the real world value it represents. According to the American Gas Association's AGA-12 standard there are about 150-200 SCADA protocols [18]. Some of the more popular SCADA protocols, as shown in table, are: MODBUS, IEC 60870-5-101, and DNP3, but none of these currently contain security. At the same time there has been increased public availability of network access and computer technology. As a result, there is now almost always the possibility of an external connection being able to reach the control network, whether through an intranet, a business partner's networks, or the Internet. In addition to these standard network paths, many SCADA systems make use of modems to provide connectivity which can also allow an external connection into the SCADA network [31]. For example, the use of war-dialers to connect to remote SCADA equipment. The assumption that SCADA networks are isolated and therefore protected from potential conversion is simply not true today.

The request message includes an operation, a point type and an index [32]. These indicate the operation to be carried out (read or write), the point type (analog input, analog output, digital input, digital output), and the point index (beginning at index 0) respectively in figure 2.1. These fields are sufficient for the middleware layer to translate the request into an appropriate protocol. Upon receipt of a request the Ethernet-serial Security Middleware

first consults the protocol access control policy to determine whether the operation is allowed or not by calling the check access function, based on the algorithm.

2.6. GENERAL SCADA NETWORK

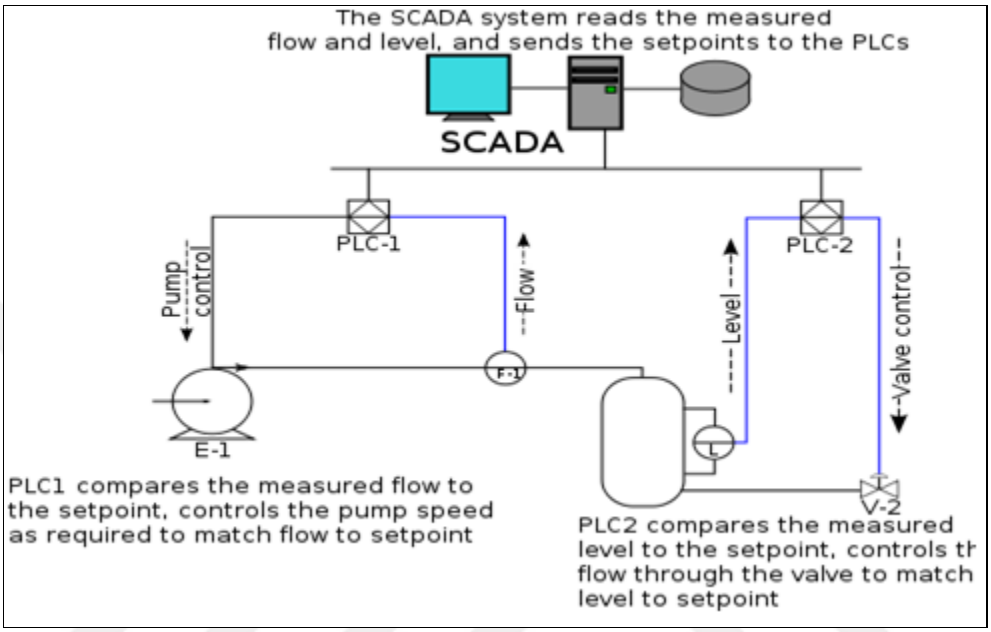


Figure 2.1: General SCADA Network and measuring the flow and level through RTU's

Systems of SCADA are typical for the organization of industrial; meanwhile these systems work to continue the level of the efficiency [34]. These processes perform operation on the data for smarter decisions [35]. These System of communication problems to solve the mitigate down-time. Simple Supervisory control and data acquisition structure instigates through program logical controller/remotes terminals units (RTUs). SCADA and remotes terminals units are microcomputers which talk through array of the gadget including HMI, machines of the factory, Sensor and more give up gadgets, course of records to the ones of the object the computer systems with help of Supervisory control and data acquisition software.

Earlier in SCADA systems used to rely on serial communications, mainly RS-232 protocol, to communicate with the RTUs and SCADAs. With the fast development about the communications and the high available of internet connection, newer protocols are proposed that rely on communications through networks [37]. However, as the SCADA systems are quite expensive, most of the earlier systems are still working, which makes

them incompatible with the newer RTUs and SCADA. Thus, it is important to allow SCADA systems with the RTUs by converting the communications among them from the source protocol to the destination protocol [34-35].

2.6.1. Raspberry Pi

A Raspberry pi vulnerability comes from the increased data exchanges between businesses achieved through network connectivity [39]. For example, deregulation in the power industry has created vulnerabilities for electric power generation, transmission, and distribution Raspberry pi. As a result of deregulation, data exchanges between single vertically integrated organizations have been replaced by many horizontal relationships among independent entities. Some of the vulnerabilities that result from deregulation are described. The complex interaction among entities not only increases the network connectivity of raspberry pi but can require multiple master and multiple remote architectures with many different entities needed varying degrees of access.

2.6.2. SOC (System on Chip)

Multifaceted is the solution of the major elements of function into a chip/chipset. This is a Processor which can be programmable, on the memory of the chip. Accelerate hardware functions (for example G.P.U) Soft-ware and Hard-ware, Components of the Analog, Benefit System on Chip, Less cost of overall Systems, Performance will increase, Less consumptions of power, Reduction in Size [38].

2.6.3. SOC in Raspberry Pi: Broadcom BCM2835 System on Chip Multimedia processor

Central Processing Unit

ARM 1176JZF-S 700MHz.

RISC Architecture and less power draw.
Not compatible with traditional PC software.

Graphic Process Unit:

a. Broad-com Video 4.
b. Special. Graphic instructions set.

Random Access Memory

a. 512 MB
b. 256 MB

2.7. DISPLAY AND AUDIO CONNECTIVITY

Comprehensive guidelines for creating secure Display and Audio Connectivity and control networks are also being developed by several industry organizations [39]. These documents provide guidelines for establishing secure Display and Audio Connectivity through definitions and best practices; in some cases, specific technologies are discussed, but in others only the desired result is given.

SCADA systems have different performance requirements than do traditional IT systems. Though not all SCADA systems and process control systems have hard real time requirements, it is important that the SCADA system (in this case the RTU) have reasonably short response times. Since different systems have different requirements, there is no established targeted response time.

1. Compositing RCA.
2. Signals of Analog.
3. 480i, 576i resolutions.
4. Digital signals.
5. Signals of Audio and Video.
6. 480i, 576i resolutions.
7. 3.5mm (jack).
8. DVI cannot carry audio signals.
9. Up to 1920x1200 resolutions.

2.8. STORAGE

Layered security deploys security elements in each of three layers of a computing environment, personnel, network, and operating system [40]. Each layer includes some form of examination, detection, and prevention. According to this model, the storage are segmented and compartmentalized based on functional groups and access control plans. Access control matrices are developed that provide a detailed security policy, which is then

implemented using security products for examination, detection, prevention, and encryption at the various layers.

Form factor

– SD, Mini SD, Micro SD

Cards Type:

– SDSC (Secure Digital) 1MB to 2GB.

– SDHC (from 4 GB max 32 GB).

– SDXD of 2 TB or up.

The system state is the fourth and final context function for storage. The internal representation of system state is based on definition. System state is determined by the internal state of the storage, and is stored in a location accessible to the security middleware layer [40]. The prototype begins in the startup state then transitions to the operating state. The system state could be changed by other processes (if they have permission). For testing purposes the system state can be manually manipulated by changing the current system state stored in the file in figure 2.2.

2.8.1. Connecting

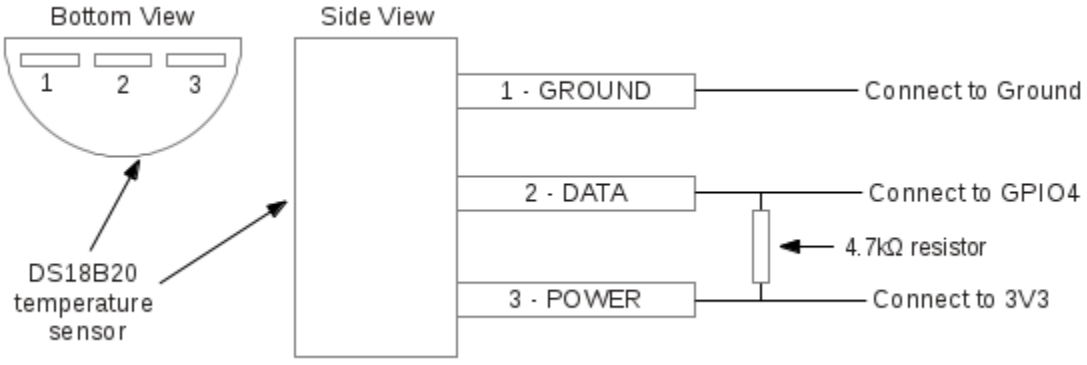


Figure 2.2: Raspberry pi connecting to respective network using ground power and data.

3. METHODOLOGY

This section will describe the methodology of the Ethernet serial protocol conversion with SCADA system using the raspberry pi. All of the following modules were developed in Python, as it has a wide open source community, with a number of advance networking and switching libraries available, already optimized for the purpose of this methodology work.

3.1. THE RROTOCOL AND COMMUNICATIONS TRENDS (SCADA)

SCADA systems contain on SCADA hosts. RTU and other field devices with the equipment process control and monitor from the different locations. And interchange or communicate data from the different locations by LAN/WAN.

Most SCADA Protocols and Communications Trends have no security features and is unable to authenticate or authorize individual requests [42-44]. A device using Modbus/TCP typically lacks packet filtering capabilities and therefore will carry out any legitimate command that reaches it. A common network security solution would be to filter the protocols and communication trends as it passes through a firewall or router, enforcing an access control policy for device connection. However, this only allows access control at a source level, while some organizations' security policy may dictate that some hosts have read access to data, while other hosts have both read and write access to the device. An application layer filtering firewall is presented by Franz that allows filtering of packets based on Modbus header values. This makes it possible to grant some hosts the ability to read from the Modbus slave device while not writing to it, and to other hosts the ability to both read and write.

A new access control model for protocol and communications trend described two reduced kernel approaches for strengthening RTU security. This chapter describes the development and testing of a prototype hardened RTU. The prototype implements the developed RTU role based access control model as a middleware layer available to other RTU processes, and uses a reduced Ethernet-serial protocol kernel [43]. A security enhanced DNP protocol similar to that described by Patel was included to provide SCADA access to the prototype [45]. RTUs and other industrial controllers usually have less available memory and processing power than traditional computing systems and have different performance

requirements as well. The prototype was developed on actual RTU hardware from six-net and evaluated in a test bed environment including actual SCADA hardware. Both performance analysis and security testing were conducted in the prototype evaluation.

3.2. PROTOCOLS

At the time that SCADA systems were first developed, the mainframe computer was the dominant computer technology. Networks were virtually non-existent making mainframes standalone machines. The communication technology they developed was driven solely by this goal and in the absence of any of today's WAN protocols. Communication links with RTUs were largely unchanged relative to first generation systems, and in general vendors maintained control over what hardware, software, and devices were available for a specific Ethernet-serial protocols.

Open standards have removed the limitations that proprietary protocols placed on SCADA systems and therefore make it much easier to use Ethernet-serial protocols to build SCADA systems [46]. A secure protocol that addresses message integrity and sender authentication is presented by Patel. Several approaches are highlighted, such as SSL-TLS wrapping, the use of digital certificates, and the use of challenge response with a pre-shared secret. The DNP3 protocol is extended to include the necessary authentication objects so that RTUs or MTUs can use the proposed protocol to verify sender authenticity and detect modifications to messages. A threat analysis and formal proof techniques support security claims about the communication protocol. The focus of the protocol is on integrity of message and sender authenticity and is not concerned with confidentiality.

3.3. SCADA - RTU PROTOCOLS

At the heart of SCADA networks are SCADA protocols. These provide the template for communication between SCADA components, typically between the MTU and the RTU. Early SCADA systems, the first and second generation SCADA architectures discussed previously, used proprietary protocols, but in more recent years there has been a move to open standards in SCADA protocols. The limited bandwidth of early communication channels resulted m a very compact message format, supporting only the most basic information needed to achieve RTU to MTU communication.

3.3.1. MODBUS

The eight-bit function code specifies what operation is to be performed by the RTU. The bits following the function code are an addressing scheme that indicates the set point, control point, or data on which the operation is to be carried out. This address has no special meaning to the RTU, and it is up to the MTU and SCADA software to correctly associate an RTU address with the real world value it represents in Table 3.1. Typical conversion scenarios like those described in center around a converter making changes to control settings, physical device parameters, or sending control commands directly to field devices. These conversion would result in a malfunctioning of the Ethernet-serial protocols which might cause a disruption in service, or possibly environmental damage or loss of human life.

Table 3.1: Modbus data table for conversion [44].

Conversion Type	**Size**	**Function**	**Modbus Address**	**Application Used Address**
Input Control	1-Bit	Read/Write	1-9999	0-9998
Input Delay	1-Bit	Read only	10001-19999	0-9998
Input Registers	16-Bits	Read only	30000-39999	0-9998
Holding Registers	16-Bits	Read/Write	40001-49999	0-9998

3.3.2. MODBUS X

One of the most serious vulnerabilities faced by SCADA system is the commonly held misconception that control networks are isolated and therefore not accessible to converters. The goal is for converters to cease committing resources to the conversion before the puzzle difficulty level adversely impacts the delay of SCADA messages. Simulation using ns2 was done to evaluate the potential impact on the latency of Modbus x. The focus of their simulation was on routine Modbus x transactions, which they claim must have a delay

time less than 540 milliseconds. The simulation found that for normalized difficulty levels below -9.5 latency increase was acceptable.

3.3.3. DNP (Distributed Network Protocol)

The DNP is extended to include the necessary authentication objects so that RTUs or MTUs can use the proposed protocol to verify sender authenticity and detect modifications to messages in figure 3.1. A threat analysis and formal proof techniques support security claims about the communication protocol.

As computers and network technology began to become available and used throughout the enterprise, there has been increased demand by industry for connection between the plant floor and the corporate network.

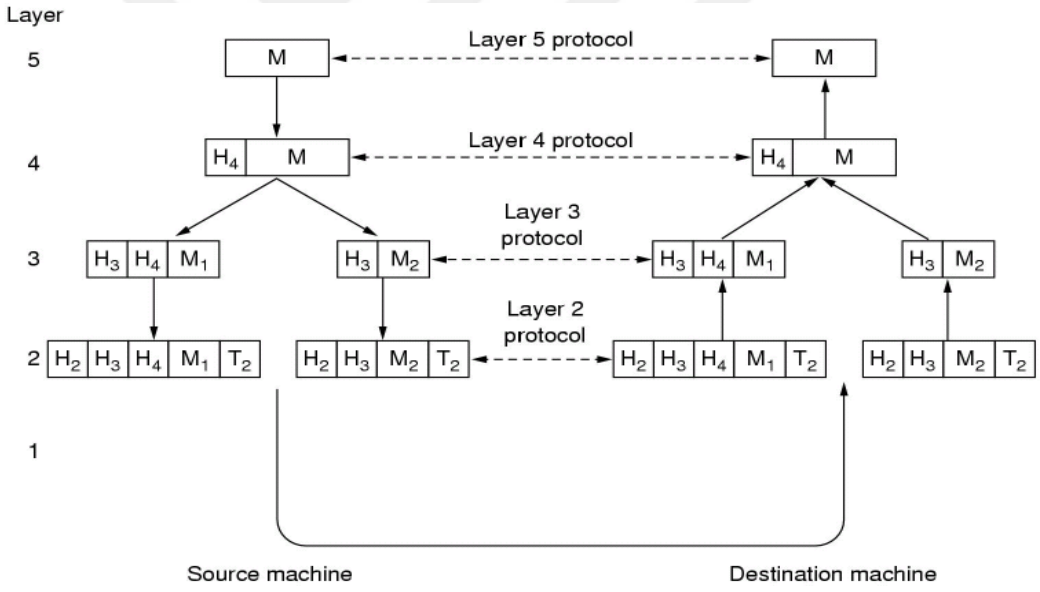


Figure 3.1: DNP3 connecting the RTU's through DATA gateway for DNP3 service [43].

The final component of the security hardened RTU is the use of a security enhanced SCADA protocol. The security enhanced protocol described by Patel in his doctoral dissertation [29] was used as the basis for this portion of the hardened RTU prototype. The security enhancements were applied to the DNP3 protocol. A more detailed description of the DNP3 protocol is described in appendix B, and Patel's enhancements are fully explained in [29]. This section provides a brief description of the modifications related to the prototype implementation [47]. The scheme as described by Patel is based on a single key and lacks the notion of a user. For the hardened RTU prototype implementation the

scheme was extended to include the notion of a user. This was done by adding a user field to challenge-response messages, identifying the user providing the response. This requires each RTU, in this case the prototype to know each users key. HMAC calculation is done using the appropriate user key identified in the response message. Key distribution was beyond the scope of this work and was part of the initial setup.

3.3.4. ASCII

In addition to these standard network paths, many SCADA systems make use of modems to provide ASCII connectivity which can also allow an external connection into the networks.

3.3.5. Protocols for Local Devices

The assumption that SCADA networks are isolated and therefore protected from potential conversion is simply not true today. The protocol prevents injection of unauthentic cipher text, modification of ciphel1ext during transmission, reordering of messages, and replaying of old messages, while introducing a fixed latency of 2 * b/8 where b is the number of bits in a block and for the local sustained devices.

Most SCADA systems are privately owned and operated, and operators are driven by economic forces [48]. For these reasons the economic advantages offered by open standards and open architectures has strongly motivate the adoption and integration in SCADA.

3.4. Communication Trends

The complex interaction among entities not only increases the network of communication trends but can require multiple master and multiple remote architectures with many different entities needed varying degrees of access. Some of the vulnerabilities that result from deregulation are described in The complex interaction among entities not only increases the network connectivity of communication trends but can require multiple master and multiple remote architectures with many different entities needed varying degrees of access.

Evidence of the vulnerabilities faced by communication trends is well documented in a recent assessment of the network security of power substations. In this assessment Oman

and colleagues found a number of security vulnerabilities, identified in 1997, still existed in 2002. These included such basic security vulnerabilities as default passwords and unsecured modem access [47-49]. They also found new potential vulnerabilities in the form of internet connectivity and wireless networks.

3.5. Legacy Networks

Further analysis of the external security incidents to identify entry points concluded that there are many routes into complex SCADA systems [48-49]. Having established and understood the weak security of modem SCADA systems the question then becomes how to secure them. However, SCADA systems and traditional IT systems are not the same, and care must be taken when applying existing security technologies to SCADA since these technologies, which acceptable in traditional IT environment, may have unacceptable adverse impacts on SCADA. The difference between traditional IT environment and SCADA or control networks.

Applying the experience, knowledge, and technologies of IT security to SCADA and PCS systems has been an essential first step in securing SCADA systems. As we have seen, the security threat to SCADA systems comes in a large part from the fact that these were once isolated networks [50]. When they can no longer be isolated, good network segmentation can help keep SCADA systems secure in figure 3.2. Segmentation can be provided by firewalls or through the use of a virtual LAN (VLAN). Network segmentation reduces the exposure of SCADA systems to external networks, improving security.

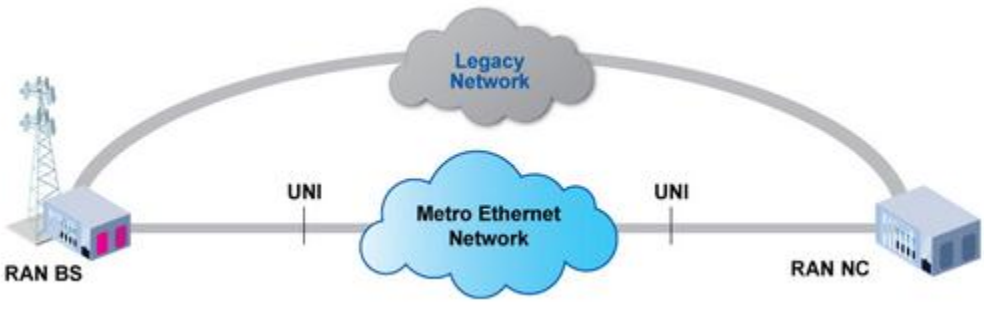


Figure 3.2: Basic architecture of legacy network using the *RAN BS* and *RAN NC*.

3.6. TELECOMMUNICATIONS

Level one is field equipment like SCADAs and RTUs. The threat identified at this level is access to data or spoofing of commands, and the recommended solution is to implement encryption. Level two is the telecom level comprised of the communication channels used to connect RTUs and field equipment to level three. The threat at this level is that these are generally unsecured communications that may be traveling over unsecured shared networks. The recommendation at this level is to consider using IPSEC. Level three is the SCADA level, essentially this is the control center. Recommendation for systems in this level include operating system hardening, patch management, network equipment access control, server access controls, physical security, converter protection strategy, and user authorization.

According to this model, the SCADA computing systems are segmented and compartmentalized based on functional groups and access control plans in figure 3.3. Access control matrices are developed that provide a detailed security policy, which is then implemented using security products for examination, detection, prevention, and encryption at the various layers. A layered security approach is advocated by Miller. Layered security deploys security elements in each of three layers of a computing environment, personnel, network, and operating system.

3.6.1. Flow of Network LAN

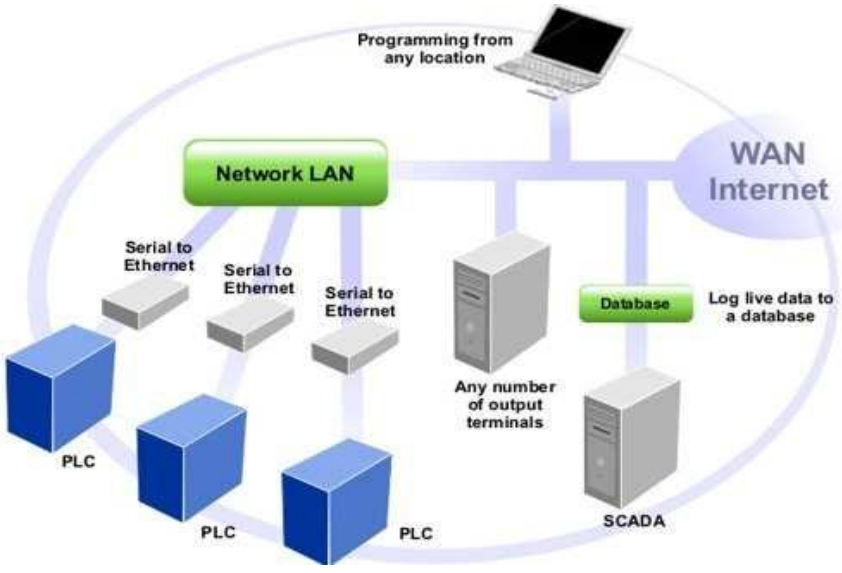


Figure 3.3: Flow of Network LAN and communication of system through SCADA and RTU's.

3.7. FRAME RELAY

Frame relay is a packet switched network. In addition to the use of specific security mechanisms and technologies for securing SCADA systems, improved management strategies and processes are also needed. Abshier summarizes ten important design and process principles for securing control systems [54]. The goal of following these principles is to ensure that due diligence has been followed in securing an organizations control systems. Cost is based on three features: committed information rates, access circuit and port speed. This vulnerable device has the highest vulnerability level. Additional research is needed in determining the initial assignment of vulnerability levels.

The Frame Relay specification asserts that the protocols or field device must be able to enforce access control on protected data based on at least the following criteria: roles, location (of the subject), and time of day / day of week. Justification is given in some detail in the frame relay document in figure 3.4. The protection profile provides only guidance about what is expected and ignores implementation specifics completely; nor does it specify how access control constraints are to be specified.

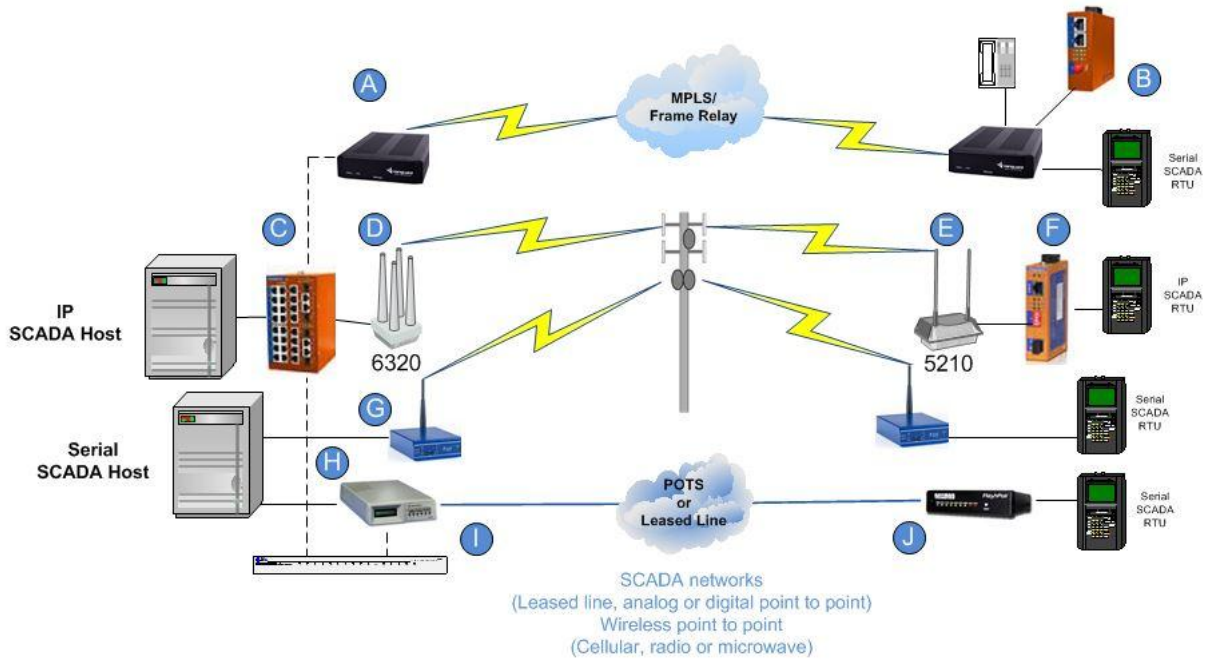


Figure 3.4: Analog to digital SCADA conversion with wireless points like cellular and radios

3.8. ETHERNET SERIAL PROTOCOL

A generic Ethernet policy framework to assist in the creation of SCADA security policies and connectivity of networks is described by Young, Stamp, Dillinger and Rumsy [55]. The framework is organized in three hierarchical layers and supports detailed specific sub policies that support generic high level policies. The framework was developed out of the author's experience in Ethernet assessments and secure communication system 34 development and implementation. The policy they develop is broken down into eight main categories at level one, ranging from data security and personnel security to network security and physical access. The goal of the policy framework is to assist asset owners in the creation of Ethernet. Implementation details for two example cases are given to demonstrate the model.

The obvious Ethernet-serial protocol will issue control commands to connected actuators and disrupt operation of the SCADA system. Other threats include deleting or modifying stored data to hide evidence of the conversion and replacing or modifying application code to corrupt system integrity. In addition, a privilege escalation threat also exists, where a converter, having gained some level of access, attempts to increase their rights by converting the access control configuration.

3.8.1. Fiber Optic Networks

Fiber optic networks and other control system also tend to have very different performance needs from traditional IT systems. Delaying the delivery of information even for a relatively brief moment is not acceptable in SCADA systems, though they often do not require a high degree of throughput. However, IT systems typically do require a high throughput but are much more tolerant of delays or jitter. In addition, many fiber optic networks may have much greater resource constraints than would be found in traditional IT systems [55-57]. This lack of computing resources along with performance constraints can make it difficult or impossible to apply standard security technologies. One of the real challenges presented by fiber optic network is the relatively long life of SCADA components compared to their IT counter parts. The policy they develop is broken down into eight main categories at level one, ranging from data security and personnel security to network security and physical access. It is not uncommon for fiber optic networks to be in

use for fifteen to twenty years, while the average IT system as life span of three to five years.

3.8.2. TCP/IP Networks

One of the primary TCP/IP network run on the less powerful processors often found in SCADA devices prohibits the use of public key protocols for performing authentication unless highly specialized cryptographic accelerators are used. Leading the way in TCP/IP networks, particularly for legacy systems is the AGA 12 working group established in 2001 by the American Gas Association (AGA). The working group was to recommend solutions to that would help protect gas utility SCADA equipment from Ethernet serial protocol-conversion.

The group determined that unprotected serial based communication channels posed the greatest threat. In response to TCP/IP networks AGA 12 has developed a serial SCADA protection protocol (SSPP) which is implemented by a separate device called SCADA Cryptographic Modules (SCM. The proposed architecture of using two SCMs to provide encrypted communications between TCP/IP networks.

3.8.3. IP Addressing

IP addressing is created on the concept to recognize the host and the network. The two contributing factors to the good security of commercial operating systems identified included the size of the code base and their monolithic design. To address this shortcoming while maintaining the economic benefits of using less network addresses. It consists of two versions IPv4 and IPv6.

3.8.4. Virtual SCADA Networks

Communications are restricted to only these communication paths. They then describe a new format for packets within the Virtual SCADA network, and key generation, key storage, and key use associated with each communication path [58]. Only substation to substation communications use public key algorithms, and then only for key exchange. They also describe a process for recovering from a substation penetration.

3.8.5. Wireless Networks

A WAN/wireless gateway provides remote access to the WAN much like an RTU provides remote access to Distance devices [53]. The proposed security architecture is to place a smartcard with one or more pairs of public-private keys with each node. Data encryption is carried out by the smart card, and a corresponding smart card on the receiving end.

3.9. SCADA SYSTEM DESIGN CHALLENGES

The primary challenge of SCADA system design is Ethernet serial protocol based threat to SCADA systems is that an unauthorized person or agent will access the SCADA system and interfere with its operation.

Over the past several years' industry groups and academics have begun to work towards addressing the SCADA security issue. This can be seen in the increasing number of publications related to SCADA identifies three *design challenges* in the field of SCADA security. The first challenge is to improve access controls to SCADA networks to make it harder for converter to gain access to the SCADA network. The second challenge is to improve security inside SCADA networks, including developing efficient monitoring tools that make actually carrying out a conversion difficult.

SCADA protocols has no security features and is unable to authenticate or authorize individual requests. A device using Modbus/TCP typically lacks packet filtering capabilities and therefore will carry out any legitimate command that reaches it [49-51]. A common network security solution would be to filter the Modbus/TCP port as it passes through a firewall or router, enforcing an access control policy for device connection. However, this only allows access control at a source level, while some organizations' security policy may dictate that some hosts have read access to data, while other hosts have both read and write access to the device

An application layer filtering firewall is presented by Franz [61] that allows filtering of packets based on Modbus header values. This makes it possible to grant some hosts the ability to read from the Modbus slave device while not writing to it, and to other hosts the ability to both read and write.

An alternative architecture for the information and communication network of power systems is proposed by Xie, Manimaran, Vittal, Phadke, and Centeno [52-56]. The proposed architecture includes all the traditional elements of SCADA systems. The primary object of the architecture is to provide greater reliability through redundancy, though communication security is considered as well. Another next generation SCADA communications architecture is proposed by Hauser, Bakken, and Bose [57]. The proposed architecture, referred to as GridStat, is a middleware framework with API stubs that correlate with traditional SCADA functions polling, events status, and control settings. GridStat was designed to support flexible communications, making new types of controls and better situational awareness possible. GridStat also provides schemes for trust management, with the ability to approve new subscription, make routing decisions, and manage access control.

This simplifies the assignment of permissions to users, more accurately reflects how organizations think about permissions, and greatly simplifies role revocation. A user is most often a human being, but the notion of user can be extended to other entities like devices, networks or autonomous agents. Roles attempt to approximate different job functions within the organizational construct in which the system is participating. A permission is the right to carry out an operation on one or more objects. An operation is some type of function to be carried out by the system for a user. Objects are entities that contain or receive information; their exact type depends on the system. Hackers break into networks and systems for the thrill and challenge that it presents. SCADA systems are not exempt, and are now receiving the attention of hackers.

Botnets are a collection of compromised computers controlled by single person, usually referred to as a bot-herder. Botnets are used to carryout coordinated conversions, send spam, or carryout phishing schemes. Botnets make use of automated conversion software. Botnets present two threat vectors, one they can be used to carry-out a conversion on SCADA systems, or two, SCADA systems may become part of a botnet and have their resources depleted by the botnet activities. Seek to acquire trade secrets, or inside knowledge that can give one organization advantage over another. SCADA systems in manufacturing industries will have knowledge of trade secrets, or just private status data. Corruption of a competitor's SCADA system at the appropriate time could have financial

benefits for the competitor. Terrorist seek to destroy or incapacitate critical infrastructure in order to damage public moral. Ethernet serial protocol-conversions on SCADA systems are one way to achieve this and may be possible from a point of relative obscurity. Ethernet serial protocol-conversions on SCADA systems may also be used to leverage a physical conversion, for example by hiding alerts of a malicious physical conversion [40]. A converter is a program that can replicate itself and pass on malicious code to other non-malicious programs. Converter can corrupt files and disrupt or interfere with the normal operation of a computer system. Worms are automated programs that propagate themselves though networks by exploiting a common vulnerably. Worms can exhaust network and computer resources, as well as harm files on the victims. Disgruntled insiders have been main source of computer crime since they have knowledge of and access to internal systems. Insiders include employees, business partners and vendors. Insiders may not necessarily be malicious, but accidental mistakes can have the same consequences as malicious conversions.

Early SCADA installations were characterized by closed systems and proprietary protocol standards. Most SCADA systems are privately owned and operated, and operators are driven by economic forces. For these reasons the economic advantages offered by open standards and open architectures has strongly motivate the adoption and integration in SCADA. In addition to assumption the SCADA networks were isolated, was a widely held belief that it was difficult to acquire information about SCADA system. Open standards and open application layer interfaces that make use of available commodity software, such as a web interface. These additional application layer interfaces in to device introduce additional vulnerabilities and conversion vectors into SCADA systems.

4. SOLUTION

This section will describe the implementation of the Ethernet serial protocol conversion with SCADA systems using raspberry pi. We suppose that data from the SCADA system is received on the serial ports of raspberry pi. The data received on serial port is then transmitted to a remote server using TCP communication. TCP communication is the most reliable and protected communication.

Concern for the security of industrial control systems has been amplified by the fact that many, if not all, of our nation's critical infrastructures are heavily reliant on these control systems for reliable and stable day to day operation. The Patriot Act defines critical infrastructures to be "systems and assets, whether physical or virtual, so vital to the United States that the incapacity or destruction of such systems and assets would have a debilitating impact on security, national economic security, national public health or safety, of any combination of those matters". The President's Commission on Critical Infrastructure Protection found that there was a "growing Ethernet serial protocol dimensions associated with infrastructure ... and ... the defenses that served us so well in the past offer little protection from the Ethernet serial protocol conversion threat". While the United States has been fortunate that a major conversion has not been successfully carried out against any critical infrastructure SCADA systems, incidents have occurred. The slammer worm penetrated part of the network at a Davis nuclear power plant in Ohio and disabled part of the safety monitoring system for nearly five hours. Fortunately, the plant was shut down for repairs at the time. In another incident, a hacker using a radio transmitter was able to open valves and release raw sewage from an Australian sewage treatment plant. The reluctance of companies to release incident information along with the possibility that some or many incidences go unnoticed makes it difficult to accurately assess the risk. One attempt to track incidents, the Industrial Security Incident Database maintained by the British Columbia Institute of Technology (BCIT), has shown a sharp increase in security incidents. Addressing security for SCADA is an ongoing task with many challenges. One challenge is that these systems tend to have a very long deployment life, up to and even beyond twenty years; consider the difference in computing technology between today and twenty years ago. Addressing the security needs for next year is challenging, addressing the security needs two decades into the future is daunting at best. Economics also plays a role

because the cost of updating or replacing SCADA systems is significant, meaning that security solutions for legacy systems are needed. However, control systems are gaining in popularity; the global revenue from the sale of control systems is expected by 2009 [51]. As this growth continues and as network convergence becomes an increasingly un-avoidable reality, it is of utmost importance that the next generation of SCADA systems be security hardened against all types of Ethernet serial protocol based conversions. The SCADA architecture is generally broken down into a master station or MTU used by human operators to monitor and control remote terminal units, or RTUs. A communications network provides communication channels between MTUs and RTUs. Security hardening techniques are needed for the various components as well as for the SCADA system as a whole. RTUs interact with physical devices like valves and switches. A primary SCADA security objective is to prevent unauthorized or improper operation of valves, switches, or other physical devices, since these devices could have economic consequences for a SCADA operator as well as potentially disrupting normal operation of U.S. critical infrastructures. The fact that RTUs can, and often are, physically remote makes securing them that much more important. This dissertation describes research and development of a security hardened RTU. While protecting and securing existing systems is important, the aim of this dissertation is to explore the development of next generation RTUs. As existing RTUs are replaced in existing SCADA deployments and as new SCADA systems are deployed, it is important that these RTUs be security hardened against Ethernet serial protocol based conversions. This dissertation presents an RTU role based access control model for hardening RTUs. The model is developed to prevent unauthorized alteration of analog and digital 10 points. In addition, a middleware layer deployment architecture is advocated to allow fine grained and homogenous application of an RTU access control policy. Operating system support for a middleware layer deployment is a critical factor in the assurance of the security hardened RTU. Two approaches for reduced kernel RTUs are presented. A reduced commercial-off-the-shelf kernel is one approach, and is used in the development of a prototype for testing using raspberry pi.

4.1. ETHERNET PROTOCOL CONVERTER FOR SCADA SYSTEMS

Countless things happen in huge modern foundation. Each procedure you have to screen is exceptionally mind boggling on the grounds that each machine gives distinctive yield. The

supervisory control and data acquisition framework used to accumulate the information from sensors and instruments situated at remote zone [55]. The programmable controllers at that point form this information and exhibits in a convenient way. Assembles and exchange the data back to the framework while giving the conversion of Ethernet serial protocol that spillage has happened and shows the data in a legitimate and sorted out design.

4.2. ARCHITECURE OF CONVERTER

Supervisory control and data acquisition (SCADA) and some protocols are built in the current version of CORE emulator. SCADA specific communication protocols are not included in CORE emulator. A mechanism to integrate Modbus in CORE emulator. They integrated Modbus as service in the emulator. Since in SCADA system there might be large number of RTUs or SCADAs with different functionalities it is not feasible to include all such component as service in CORE emulator. In our proposed approach, we developed a python script for each of the components of the SCADA system based on their functionalities. In CORE, we represented each of the SCADA devices by a virtual node, and they are connected to each other by hubs in Local Area Networks (LAN).

Table 4.1: TCP/IP protocol suite summary chart for conversion of Ethernet through different layers of Network [34].

Data Unit	Layer	Protocols	Address
Message	Application	FTP, DHCP, TFTP, DNS, SMTP, HTTP	Application(i.e. email address)
Segment	Transport	TCP,UDP	Port (Port number)
Datagram	Network	IP,ICMP,IGMP,ARP	Logical(IP)
Frame	Data	Ethernet, Wireless	Physical or link (MAC)
Bits	Physical	Ethernet, Wireless	Conversion

In real world water distribution plant there are large numbers of water tanks and pumps. For the sake of simplicity only one pump and five water level sensors are considered as field devices in the proposed SCADA test bed. Hence, one RTU with five registers that store the values of LA, L, H, HA and current water level in the primary tank is enough to implement the field station of the plant under consideration. The RTU has also four coils to store

binary values of Mode, manual ON/OFF setting, current pump status and current alarm status [55]. All the simulation components and the process are implemented using python scripting on top of CORE framework. In the next section we present different conversion scenarios and their effects in SCADA system.

To observe the effect of conversion of Ethernet serial protocol on the proposed test bed we wrote a simple Ethernet serial protocol conversion ESPC script using python. When this script runs on the converter node, the RTU will be flooded with packets on port. In CORE emulator the maximum bandwidth of the link between two network nodes can be easily configured by a user. For this specific experiment we set the maximum bandwidth of the system [60]. By flooding the RTU with packets, the bandwidth of the link between the MTU and RTU can be consumed. The more the number of packets coming from the converter the higher the bandwidth consumed. At some point all the allowed bandwidth will be consumed by the converter. In this situation the RTU will not listen requests or commands from the MTU [61]. In such scenarios the normal operation of the SCADA system will be disrupted. The bandwidth consumed on the link between the RTU and MTU for converting the Ethernet serial during execution. The experiment was made for 120 s. In this period on the converter machine we made ESPC three times on the RTU. The first conversion was attempted between 20 to 30 s and the second and the third were from 55 to 65 and 90 to 100s respectively. In those periods the conversion easily consumed the bandwidth of the link between the RTU and MTU [62].

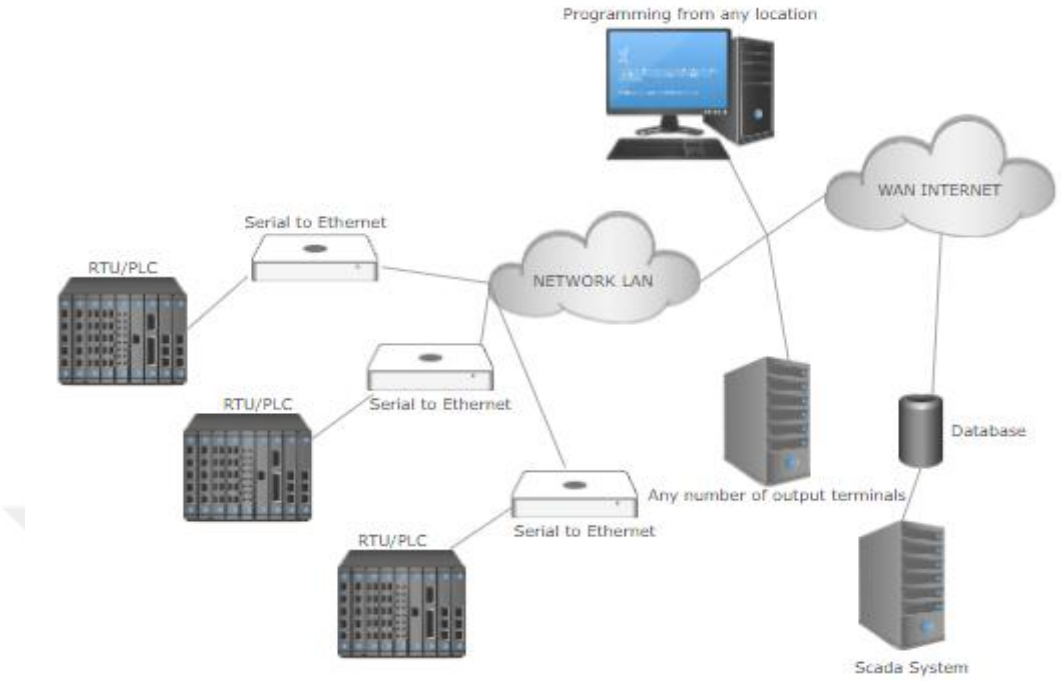


Figure 4.1: Programming from any Location

4.3. DATA COMMUNICATION

The supervisory control and data acquisition feature of the conversion with help of security researches to investigate the effect of simultaneous conversion of serial protocols on SCADA systems [55]. The test-bed was developed on top of CORE emulator. In the proposed test-bed Modbus TCP/IP is used as a communication protocol. With Ethernet serial protocol conversion ESPC and false data injection conversion scenarios, we demonstrated how conversion could disrupt the normal operation of SCADA systems. With the current version of the SCADA systems, we have already collected network and control data for SCADA security research and development [59]. The collected dataset will be freely available for security researchers in a near future. From results, we can conclude that the developed test-bed can be effectively used for Ethernet serial protocol security assessment and vulnerability investigation on SCADA systems.

4.3.1. Information/Data Presentation

Nowadays there are many types of SCADA communication protocols and data presentation is very important phase of Ethernet serial protocol conversion. Modbus, Pro-fibus and DNP3 are some of the most widely used protocols in figure 4.1. For the proposed system,

Modbus TCP/IP is used as a communication protocol between the components of the Ethernet serial protocol conversion ESCP.

4.3.2. Human Machine Interface

HMI has two variants: HMI server and human machine interface client. The HMI client requests the human machine interface server about the current status of the plant and then gives this information to the operator in the graphical form. The operator can change the parameter settings of the plant on HMI client. The HMI client forwards these values to the MTU. Based on these values, valid decisions are taken by the MTU in figure 4.2.

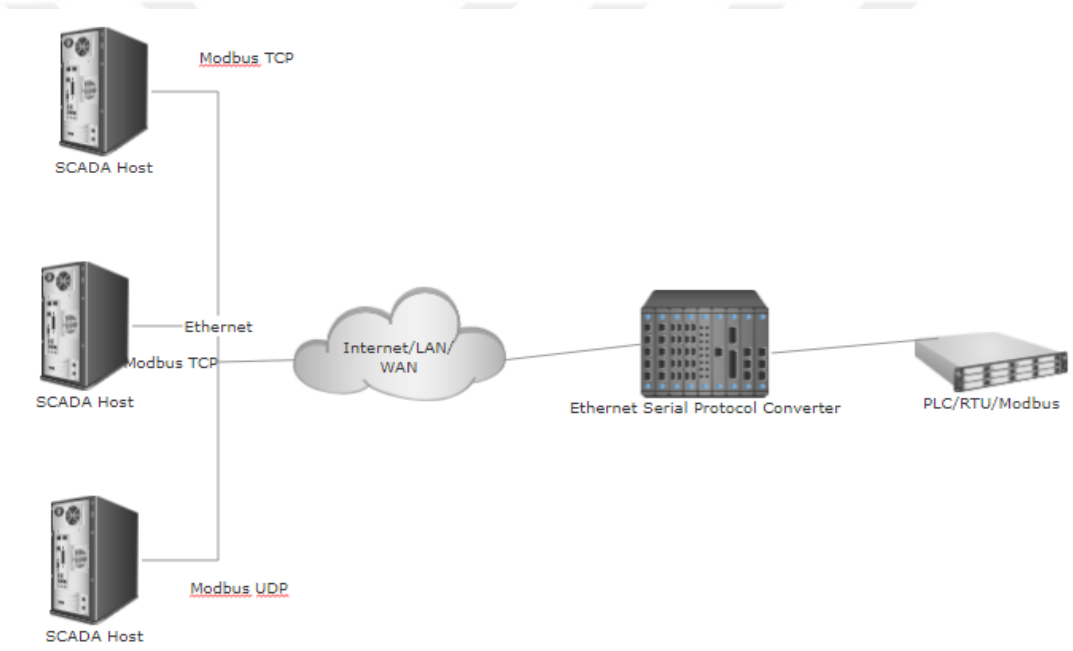


Figure 4.2: Human Machine Interface

4.3.3. Virtual Serial Server – Ethernet to Serial Conversion

SCADA ESCP have also been proposed in recent years [46–49]. However, most of these ESCP are not freely available. Even ESCP, which are developed on open software, are sector and application specific. Moreover, these ESCP lack re-configurability for different conversion scenarios. In this paper, to address these limitations, an open SCADA ESCP is proposed, which can be easily extended to various critical infrastructure sectors. More importantly, different from previously developed ESCP, the proposed SCADA ESCP is user friendly and easily reconfigurable for different types of conversions in figure 4.3. In addition to this, in this ESCP multiple conversions can be

simultaneously initiated from different places in the ESCP. This feature of the ESCP helps security researches to investigate the effect of simultaneous conversions on SCADA Systems.

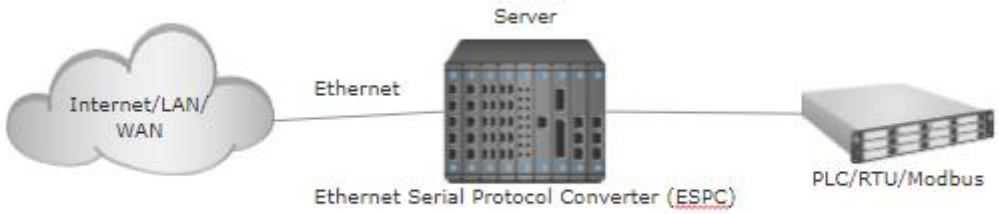


Figure 4.3: Virtual Serial Server – Ethernet to Serial Conversion

4.3.4. Virtual Serial Client

The conversion can be used for security assessment and vulnerability investigation on SCADA systems. With our client, one can also generate benchmark dataset to develop and evaluate conversion and protection technologies for SCADA systems. With the current version of the ESCP, we have already collected network and control data for SCADA security research and development in figure 4.4. Though the ESCP is primarily developed for SCADA security research, it can also be used for educational purpose.

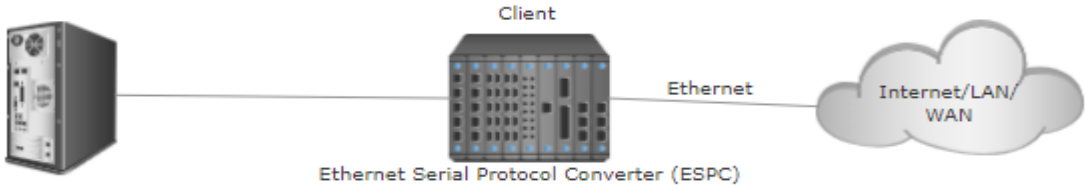


Figure 4.4: Virtual Serial client

4.3.5. Connect

Occasionally intrude on correspondence between two sequential gadgets associated with the ESPC. Under typical conditions, there is a bidirectional correspondence between gadgets on the 3 ESPC sequential ports. On the off chance that the ESPC recognizes a

Transmission Control Protocol attachment ask for, it supersedes the sequential to-sequential interchanges. While the Transmission Control Protocol attachment is built up, any information from the other sequential gadget is disposed of in figure 4.5.

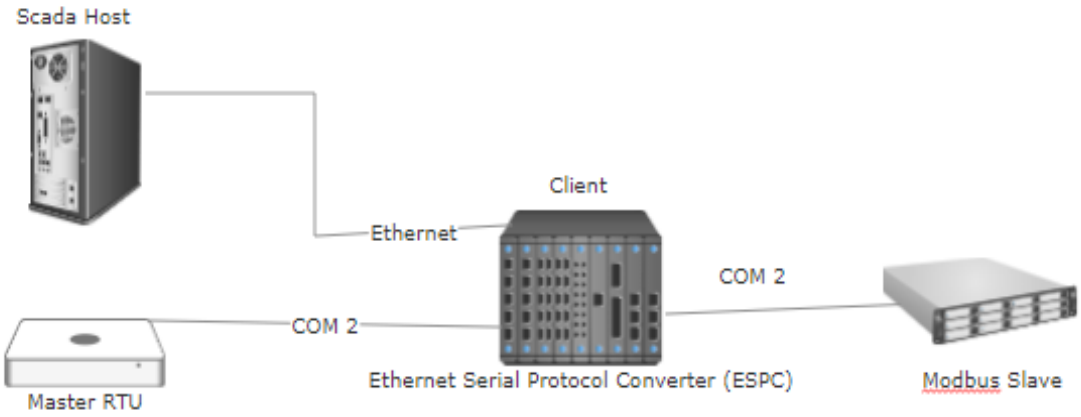


Figure 4.5: Connect methodology of Ethernet serial protocol.

4.3.6. Serial Multiplexer

Modbus Multiplexer mode empowers different sequential to impart to a solitary sequential. Up-to 2 clients can interface with a solitary sequential port in figure 4.6.

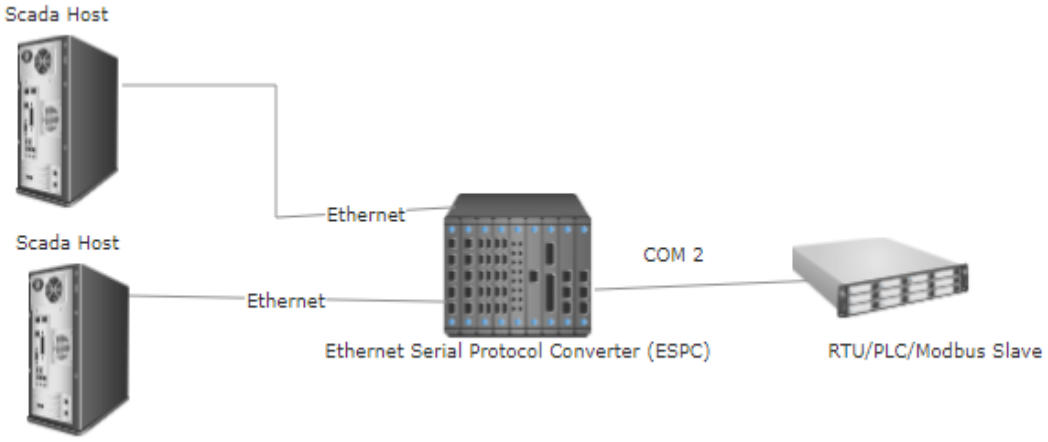


Figure 4.6: Serial Multiplexer for RTU and Modbus

4.3.7. Serial Master

This ESPC mode empowers a sequential ace to impart to sequential slaves utilizing an assortment of conventions. Two standard of Serial Master – User Datagram Protocol communicate and Remote Terminal units address based steering for conversion of Ethernet serial protocol in SCADA systems in figure 4.7. In User datagram protocol, sequential message is communicated to a rundown of User Datagram Protocol slaves. The User Datagram Protocol communicate mode is convention free. This mode can be utilized for ESPC Serial Master Mode Configured for User Datagram Protocol Broadcast.

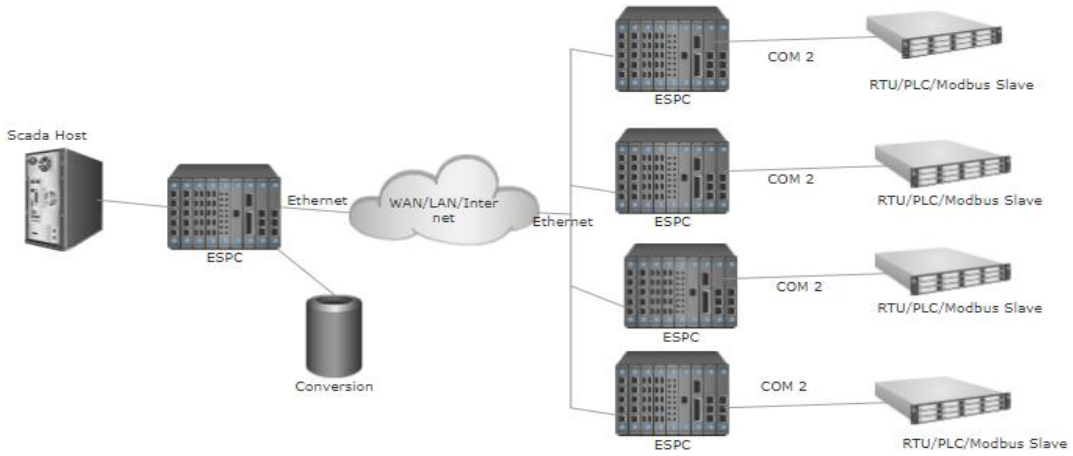


Figure 4.7: ESPC Serial Master Mode Configured for User Datagram Protocol Broadcast

Remote Terminal units address based routing utilizes a Remote Terminal Units deliver query to course to an internet protocol gadget in figure 4.8. In the model underneath, door A courses the standard utilizing the query format. Ethernet Serial Protocol Converter Configured for remote terminal units Address Standard Based Routing.

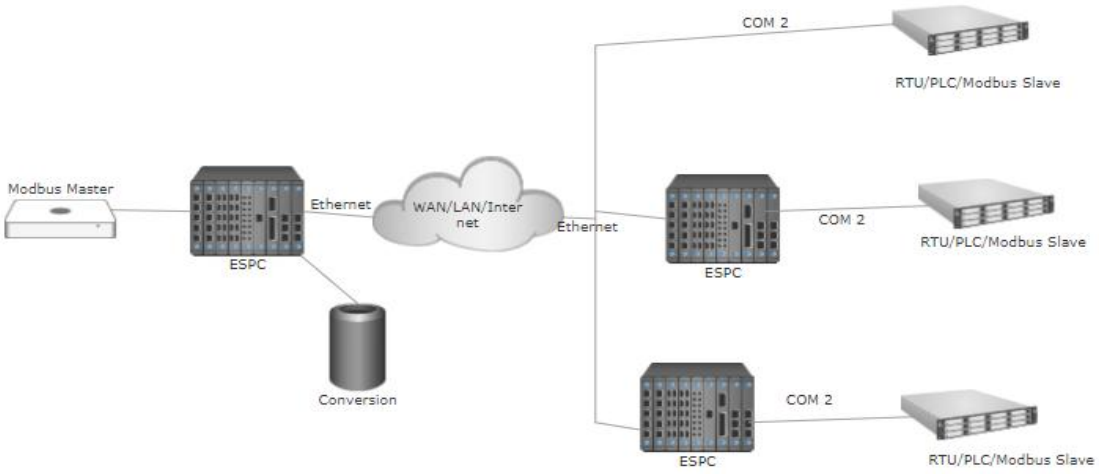


Figure 4.8: Ethernet Serial Protocol Converter Configured for remote terminal units Address Standard Based Routing.

5. RESULTS

This section will display the results of the Ethernet serial protocol conversion between SCADA system and the remote server using raspberry pi. Raspberry pi 3 B+ module is used to build a communication channel between SCADA and a remote server over Ethernet.

5.1. ANALYSIS

Building technology, event technology, house technology, conversion/evaluation requirements, private control tasks, surveillance technology, process automation, and of course automation technology as well, are moving closer together. Decentralized solutions are gaining in importance and thus the accompanying requirements for inexpensive, scalable, and industrially viable small control systems. Furthermore, there is a great need for connecting these small control systems to a network for which the need of evaluation of Ethernet serial protocol comes into way of improving the networks and our work is presented for this purpose, for example. Even the desire for our "Open Source" solutions is becoming greater and networking solutions will already soon be the standard in the industry.

Through the evaluation and conversion design of the Ethernet serial protocol, for example, different network protocols like TCP/IP based and UDP based Ethernet serial protocols have been used which can evaluate the converted the Core to digital and analog input and output modules as well as certified gateways to all important fieldbuses which were read out from the input file as displayed in the practical implementation of work.

5.2. WORK DESCRIPTION

We takes the SCADA dataset of some small values from the publically available datasets to verify our communication. Data is transferred to the server and the instructions from the server are received and saved in the database. These instructions are to be send to SCADA for control and operation. This will allow us to control and monitor the SCADA system from a control room far away from the actual installation of SCADA.

The task of the version gateways was to connect networks to different network protocols (e.g. a TCP/IP based-network to a UDP-based network). One benefit of the routing and

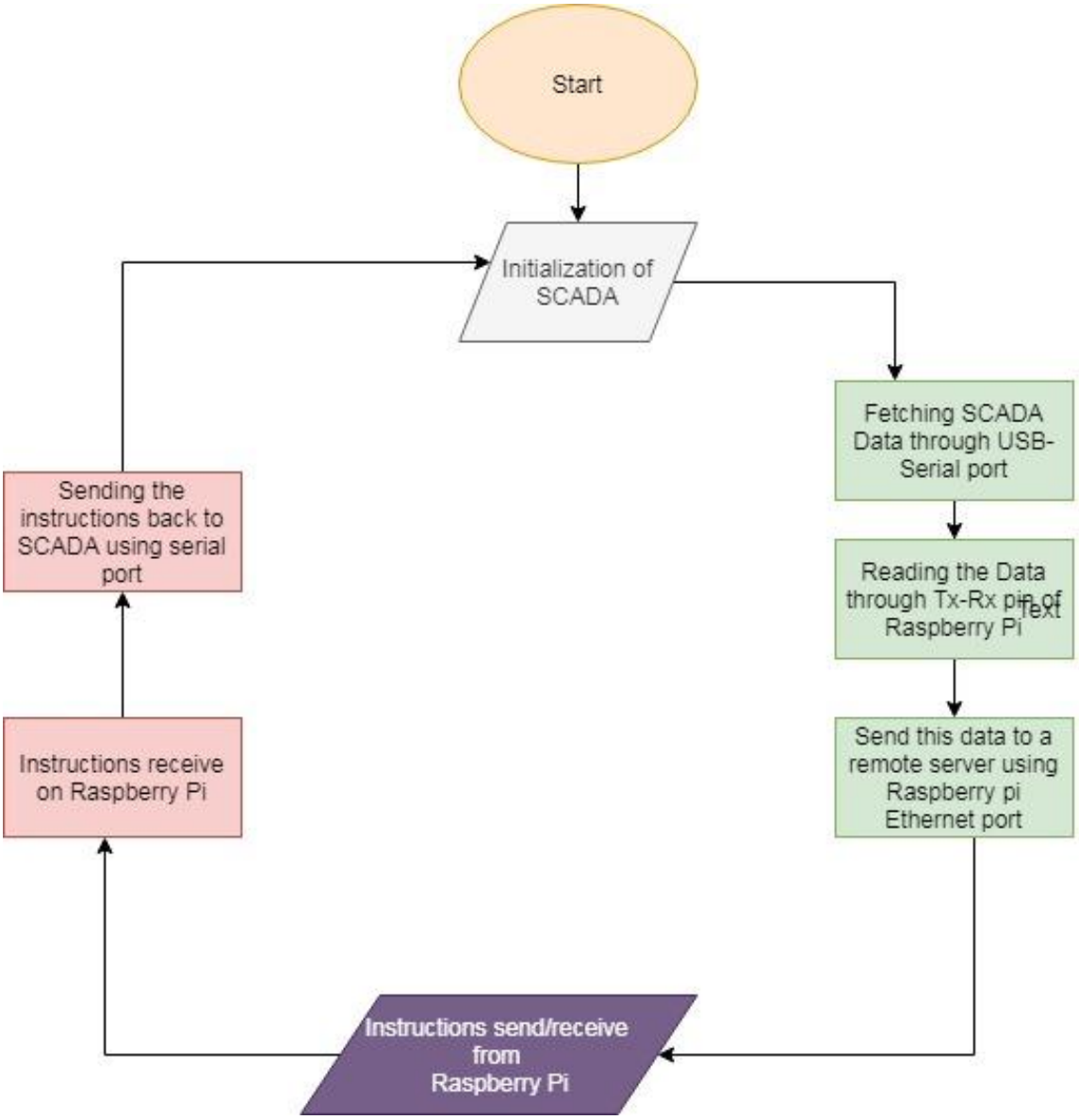
switching design is the cost factor, among other things, during changes of the technical conditions in the field. Whereas normal gateways of this conversion, for example, version 3 and 4 must be evaluated entirely when a network is changed, with our system only the evaluation and conversion of the network concerned is evaluated. The raspberry pi should advance the 4th industrial revolution. Raspberry pi behaves as a mini computer and it can do all the functions that a computer can operate. Internet and Bluetooth connectivity makes it unique for communication purpose and it will revolutionize the control and operation of industrial process. Raspberry pi has the ability to perform the following functions.

- An industrial PC (IPC)
- A PLC
- An IoT gateway
- A web server
- A software platform
- A small control unit for an HMI
- A cloud solution

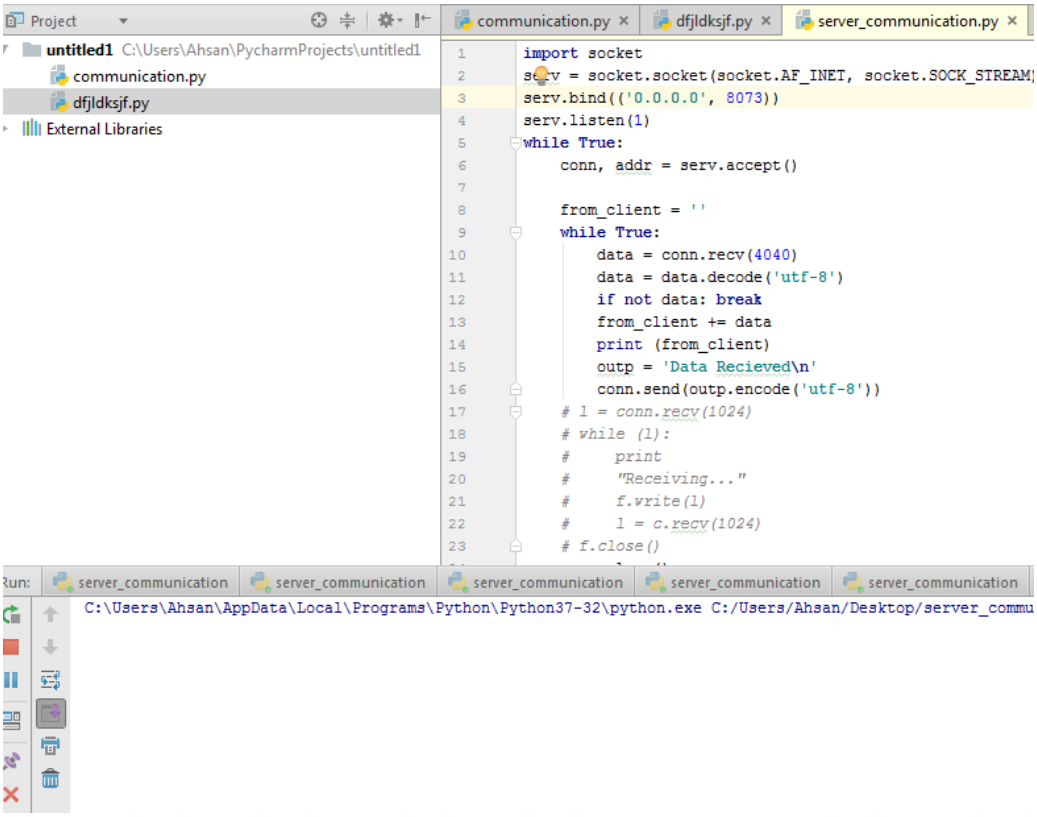
5.3. RASPBERRY PI

We use raspberry pi 3 B+ for communication between server and SCADA. We use raspberry pi to perform different tasks. It has the ability to work and give proper results for a real time system. It takes the data from SCADA using a serial pins Tx-Rx and send this data using Ethernet to a remote server using TCP communication protocols. TCP communication is the most secure communication between clients and servers. Instructions from the server are also received through the same channel and transmit these instructions to SCADA. This will allow us to control SCADA system from a remote server. We write the code in python and run it on raspberry pi and computer server both. The results shows the data received and send from both sides. The flow chart will show the working of our whole system that we build for the communication. And the screenshots of server and raspberry pi verify the communication held between them. Data is received and send on both sever and raspberry pi.

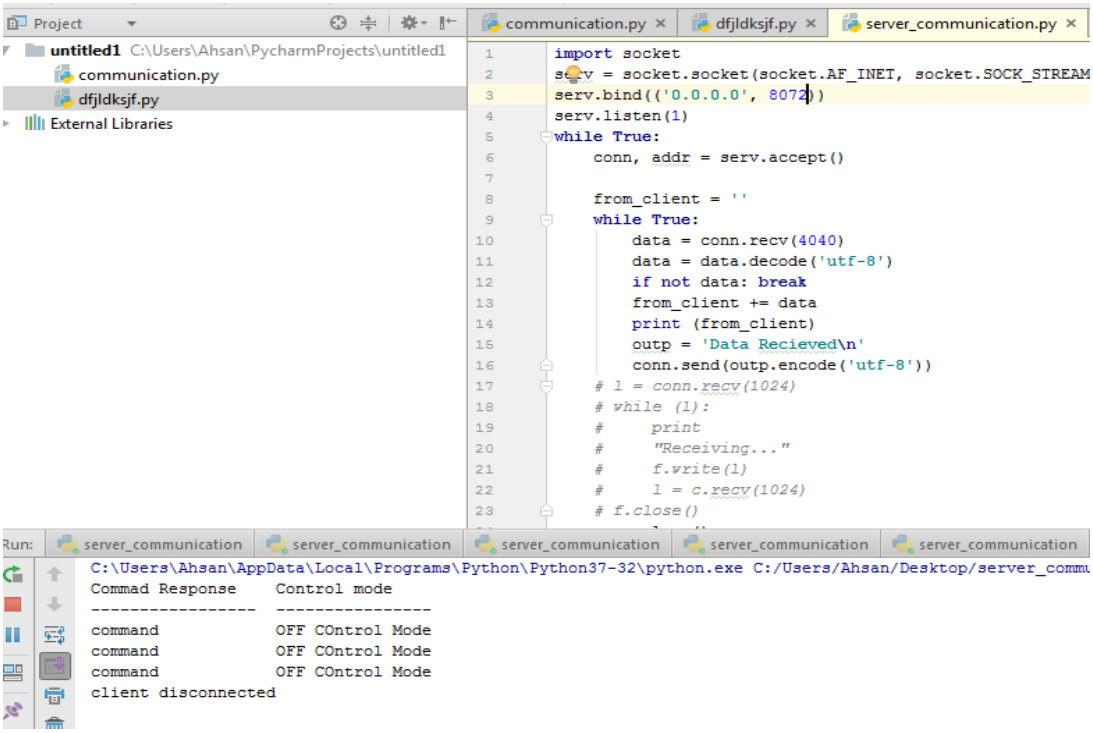
Flowchart


Start
Initialization of SCADA
Fetching SCADA Data through USB-Serial port
Reading the Data through Tx-Rx pin of Raspberry Pi
Send this data to a remote server using Raspberry pi Ethernet port
Instructions send/receive from Raspberry Pi
Instructions receive on Raspberry Pi
Sending the instructions back to SCADA using serial port

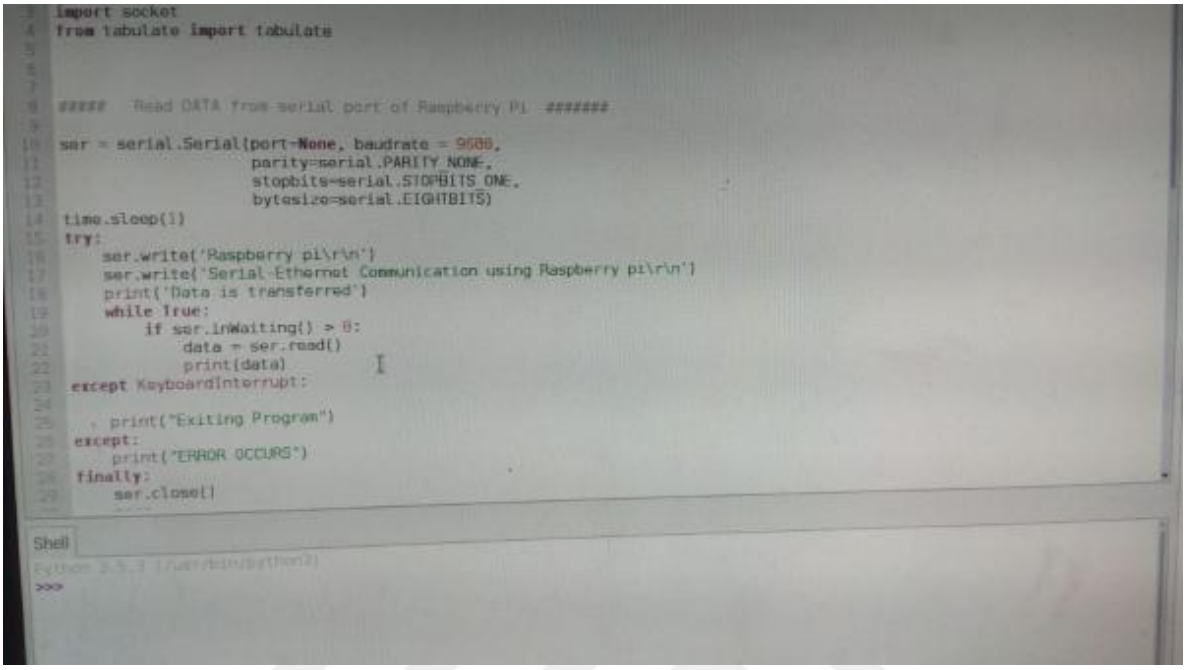
Server Screen before receiving the data:



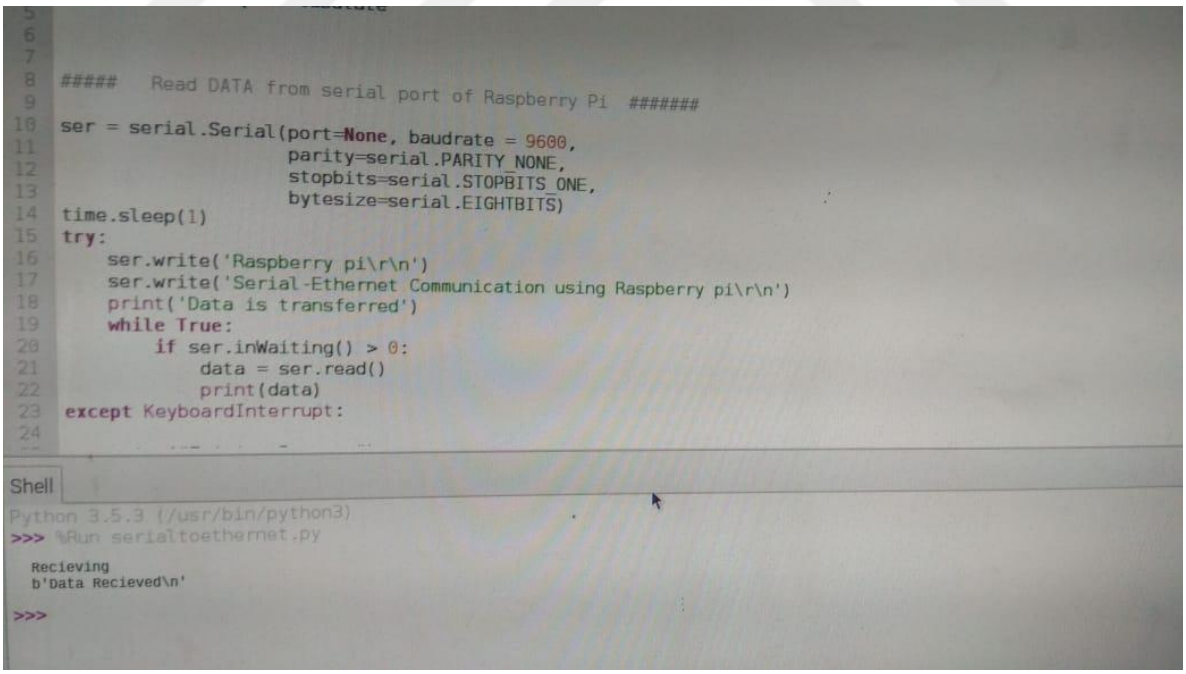
Server Screen after Receiving Data:



Raspberry Pi screen before sending Data



Raspberry Pi screen after sending Data



SCADA Data transfer from Raspberry Pi to Remote Server

Mac Address of server: 0xcc52af193063

***** IP *******

Version: 3

IHL: 8

TOS: 6

TOTAL LENGTH: 28088

IDENTIFICATION: 100

FLAGS: 00

FRAGMENT OFFSET: 8069

TTL: 27

PROTOCOL: 16

CHECKSUM: Address CommandResponse ... deltaSetPoint Label

0 0x4 Command ... 0.0 Good

1 0x4 Command ... -20.0 Good

2 0x4 Command ... 20.0 Good

3 0x4 Command ... -20.0 Good

4 0x4 Command ... 20.0 Good

5 0x4 Command ... -20.0 Good

6 0x4 Command ... 20.0 Good

7 0x4 Command ... -20.0 Good

8 0x4 Command ... 20.0 Good

9	0x4	Command	...	-20.0	Good
10	0x4	Command	...	20.0	Good
11	0x4	Command	...	-20.0	Good
12	0x4	Command	...	20.0	Good
13	0x4	Command	...	-20.0	Good
14	0x4	Command	...	20.0	Good
15	0x4	Command	...	-20.0	Good
16	0x4	Command	...	20.0	Good
17	0x4	Command	...	-20.0	Good
18	0x4	Command	...	20.0	Good
19	0x4	Command	...	-20.0	Good
20	0x4	Command	...	20.0	Good
21	0x4	Command	...	-20.0	Good
22	0x4	Command	...	20.0	Good
23	0x4	Command	...	-20.0	Good
24	0x4	Command	...	20.0	Good
25	0x4	Command	...	-20.0	Good
26	0x4	Command	...	20.0	Good
27	0x4	Command	...	-20.0	Good
28	0x4	Command	...	20.0	Good
29	0x4	Command	...	-20.0	Good
..	...	...	...	...	...

170	0x4	Command	...	20.0	Good
171	0x4	Command	...	-20.0	Good
172	0x4	Command	...	20.0	Good
173	0x4	Command	...	-20.0	Good
174	0x4	Command	...	20.0	Good
175	0x4	Command	...	-20.0	Good
176	0x4	Command	...	20.0	Good
177	0x4	Command	...	-20.0	Good
178	0x4	Command	...	20.0	Good
179	0x4	Command	...	-20.0	Good
180	0x4	Command	...	20.0	Good
181	0x4	Command	...	-20.0	Good
182	0x4	Command	...	20.0	Good
183	0x4	Command	...	-20.0	Good
184	0x4	Command	...	20.0	Good
185	0x4	Command	...	-20.0	Good
186	0x4	Command	...	20.0	Good
187	0x4	Command	...	-20.0	Good
188	0x4	Command	...	20.0	Good
189	0x4	Command	...	-20.0	Good
190	0x4	Command	...	20.0	Good
191	0x4	Command	...	-20.0	Good

192	0x4	Command	...	20.0	Good
193	0x4	Command	...	-20.0	Good
194	0x4	Command	...	20.0	Good
195	0x4	Command	...	-20.0	Good
196	0x4	Command	...	20.0	Good
197	0x4	Command	...	-20.0	Good
198	0x4	Command	...	20.0	Good
199	0x4	Command	...	-20.0	Good

[200 rows x 27 columns]

IP SOURCE: HAMZA-PC

IP DESTINATION: 192.168.8.104

***** Evaluation of Ethernet Protocol *******

Source Port of Ethernet Serial Protocol: 8069

Destination Port of Ethernet Serial Protocol: 4049

Sequence Number:

Header Length: (28088, 27)

Bits: 2048

Window Size: 5120

Checksum: 18079

6. DISCUSSION

This study does not hold a relevant environmental implications, neither positive nor negative. As described in the objectives section, its ultimate purpose was not to improve the impact to the environment of the current technology [43]. Yet, the developed solution for the reference conversion implies an upgrading of the management and maintenance system, which in turn derives to savings of resources and time, and thus diminishes the effect on the environment. For instance, avoiding regular personnel check-ups to the network as a consequence of remote availability of information of the system status through the developed SCADA system. On a different note, another environmental advantage in this work ensues from using conversion for Ethernet serial protocol in open-source [47]. The flexibility and adaptability that these devices offer result, in many cases, in a reduction of reinvestment in new equipment as a consequence of close-standard incompatibilities or outdated versions between existing installations when carrying out extensions of it or retrofitting tasks.

The development of this work is of particular interest to conversion of Ethernet serial protocol in a firm and, more generally, to the industrial and automation sector [51]. The paradigm in this field is beginning to evolve from the closed-systems concept to the open systems. Studies on the ‘open’ concept related to Industry and the Internet of Things (IoT) are on the rise in the scientific literature. The open-source and open-standard concepts are regarded as a key plank for the evolution of the industry [54]. This new technologies will broaden the range of solutions for automation systems and boost the interaction capabilities between devices and systems of different nature or firms. The present project is encased within that framework and it will provide an example on how a SCADA system can be developed based on open technology, integrating devices from different firms, which communicate with each other via different open protocols.

7. CONCLUSION

In first place, the ultimate goal of the study was to develop and implement conversion of Ethernet serial protocol and communication protocols with SCADA systems using Raspberry Pi. It can be assert that this objective has been wholly fulfilled. In Section 3.3 it has been shown how to establish a TCP-IP communication between client and a server and define the functions and structures for exchanging data. Furthermore, Modbus RTU protocol over serial bus has been adapted and implemented to a conversion of Ethernet serial protocol in order to interface with a data acquisition device from another firm, Section 3.4. These two developed protocols are widely extended in the industry nowadays, and TCP-IP has still a vast potential of expansion, as the standard protocol for Internet. The integration of these two communications with Industrial Shields products has already had a positive effects for customers, which have been implementing it from the beginning see Raspberry Pi. Real case examples. Moreover, the integration of this communication protocols has boost interest on the devices. Another of the goals of this work was to develop a SCADA system integrating conversion of Ethernet serial protocol along with other systems. This point has been achieved indeed, as a SCADA system itself has been developed and tested. The communications between the Raspberry Pi and the interactive HMI comprising the required variables and parameters to meet the specifications set in Section 4.2 has been shown to work.

7.1. FUTURE WORK GUIDELINES

The objectives set at the beginning of the study have been fulfilled as argued in Section 1.4. It has been proved that developing a SCADA system with alternative technology is possible and its core functionalities have been implemented successfully. From that standpoint, there are some aspects that will upgrade the SCADA system and make possible to move from a prototype to a final product for conversion of Ethernet serial protocol.

- Development of a thorough reconnection capability for conversion.

In this study it has been presented how a TCP communication can be established between an conversion of Ethernet serial protocol and a PC hosting a TCP server. Furthermore, it has been exposed how to exchange data between the conversion of Ethernet serial protocol and the PC, and monitor and control the application parameters from an HMI in the PC. All

this has been proved to work fine and serves as a prototype for a SCADA system. An important functionality for a final application would be a communication control algorithm on the server side. This would monitor the communication and ensure that whenever the transmission is disturbed or broken, the Ethernet serial protocol follows the required procedure to reconnect to the server autonomously through raspberry pi.

- Creation of a database and web server application for better interface.

Another improvement to be considered for future developments is to create a web server HMI to interact with the raspberry pi instead of a desktop based application. Furthermore, a database could also be created for data storage.

- Development of security protocol for converting with more secured means.

Security is critical nowadays in data systems. For the reference case considered in the present work this might not be an issue, but if the study is to be applied to different application where data might be confidential, encrypting the information and protecting the system against conversion type errors will be paramount.

REFERENCES

[1] McClanahan, SCADA and IP: is network convergence really here, Industry Applications Magazine, IEEE, Volume: 9 , Issue: 2 , March-April 2003 Pages:29 - 36

[2] Bin Qiu, Hoay Beng Gooi, Yilu Liu, Eng Kiat Chan, Internet-based SCADA display system, Computer Applications in Power, IEEE , Volume: 15 , Issue: 1 , Jan. 2002 Pages:14 – 19

[3] Shyh-Jier Huang, Chih-Chieh Lin, Application of ATM-based network for an integrated distribution SCADA-GIS system, Power Systems, IEEE Transactions on , Volume: 17, Issue: 1 , Feb. 2002 Pages:80 - 86

[4] Marihart, D.J., Communications technology guidelines for EMS/SCADA systems, Power Delivery, IEEE Transactions on , Volume: 16 , Issue: 2 , April 2001 Pages:181 - 188

[5] Qiu, B., Gooi, H.B., Web-based SCADA display systems (WSDS) for access via Internet Power Systems, IEEE Transactions on , Volume: 15 , Issue: 2 , May 2000 Pages:681 - 686

[6] Bruce, A.G., Reliability analysis of electric utility SCADA systems, Power Systems, IEEE Transactions on, Volume: 13, Issue: 3, Aug. 1998 Pages: 844 – 849

[7] Ghoshal, K., Distribution automation: SCADA integration is key, Computer Applications in Power, IEEE , Volume: 10 , Issue: 1 , Jan. 1997 Pages:31 – 35

[8] Marcuse, J., Menz, B., Payne, J.R., Servers in SCADA applications, Industry Applications, IEEE Transactions on , Volume: 33 , Issue: 5 , Sept.-Oct. 1997 Pages:1295 – 1299

[9] Luque, J., Gomez, I., The role of medium access control protocols in SCADA systems, Power Delivery, IEEE Transactions on , Volume: 11 , Issue: 3 , July 1996 Pages:1195 – 1200

[10] Luque, J., Gomez, I., Escudero, J.I., Determining the channel capacity in SCADA systems using polling protocols, Power Systems, IEEE Transactions on , Volume: 11, Issue: 2 , May 1996 Pages:917 - 922

[11] Sciacca, S.C., Block, W.R., Advanced SCADA concepts, Computer Applications in Power, IEEE , Volume: 8 , Issue: 1 , Jan. 1995 Pages:23 - 28

[12] Dagle, J.E., Widergren, S.E., Johnson, J.M., Enhancing the security of supervisory control and data acquisition (SCADA) systems: the lifeblood of modern energy

infrastructures, Power Engineering Society Winter Meeting, 2002. IEEE, Volume: 1, 27- 31 Jan. 2002 Pages: 635 vol.1

[13] Qian Wang, Qingquan Qian, Design and analysis of communication network for distributed SCADA system, Power Engineering Society Winter Meeting, 2000. IEEE, Volume: 3, 23-27 Jan. 2000 Pages: 2062 - 2065 vol.3

[14] Wu Sitao, Qian Qingquan, Using device driver software in SCADA systems, Power Engineering Society Winter Meeting, 2000. IEEE, Volume: 3, 23-27 Jan. 2000 Pages: 2046 - 2049 vol.3

[15] Chen Qizhi, Qian Qinquan, The research of UNIX platform for SCADA, Power Engineering Society Winter Meeting, 2000. IEEE, Volume: 3, 23-27 Jan. 2000 Pages: 2041 - 2045 vol.3

[16] Ebata, Y., Hayashi, H., Hasegawa, Y., Komatsu, S., Suzuki, K., Development of the Intranet-based SCADA (supervisory control and data acquisition system) for power system, Power Engineering Society Winter Meeting, 2000. IEEE, Volume: 3, 23-27 Jan. 2000 Pages: 1656 - 1661 vol.3

[17] Chen Qizhi, Optimization of a SCADA system based on client/serve mode, Power System Technology, 1998. Proceedings. POWERCON '98. 1998 International Conference on, Volume: 2, 18-21 Aug. 1998 Pages: 1237 - 1240 vol.2

[18] Medida, S., Sreekumar, N., Prasad, K.V., SCADA-EMS on the Internet, Energy Management and Power Delivery, 1998. Proceedings of EMPD '98. 1998 International Conference on, Volume: 2, 3-5 March 1998 Pages: 656 - 660 vol.2

[19] Zecevic, G., Web based interface to SCADA system, Power System Technology, 1998. Proceedings. POWERCON '98. 1998 International Conference on, Volume: 2, 18-21 Aug. 1998 Pages: 1218 - 1221 vol.2

[20] Marcuse, J., Menz, B., Payne, J., Servers in SCADA applications, Industry Applications Conference, 1995. Thirtieth IAS Annual Meeting, IAS '95., Conference Record of the 1995 IEEE , Volume: 3 , 8-12 Oct. 1995 Pages:2124 - 2129 vol.3

[21] Bruce, A.G., Lee, R., A framework for the specification of SCADA data links, Power Industry Computer Application Conference, 1993. Conference Proceedings, 4-7 May 1993 Pages: 117 - 121

[22] McDonald, J.D., Developing and defining basic SCADA system concepts, Rural Electric Power Conference, 1993. Papers Presented at the 37th Annual Conference, 25-27 April 1993 Pages:B3/1 - B3/5

[23] Quartey, B., Shaw, D., Waked, P., An application of SCADA's as an RTU in SCADA systems, Petroleum and Chemical Industry Conference, 1992, Record of Conference Papers., Industry Applications Society 39th Annual , 28-30 Sept. 1992 Pages:271 - 274

[24] Hoge, D.J., Jensen, J.R., A comparison of protocol conversion methods for the retrofit of SCADA systems, Petroleum and Chemical Industry Conference, 1988, Record of Conference Papers., Industrial Applications Society 35th Annual , 12-14 Sept. 1988 Pages:245 - 248

[25] IEEE recommended practice for master/remote supervisory control and data acquisition (SCADA) communications, IEEE Std 999-1992, 12 Feb. 1993

[26] Blackman, J.M., Hissey, T.W., Impact of local and wide area networks on SCADA and SCADA/EMS systems, Advanced SCADA and Energy Management Systems, IEE Colloquium on , 6 Dec 1990 Pages:8/1 - 815

[27] Kwok-Hong Mak, Holland, B.L., Migrating electrical power network SCADA systems to TCP/IP and Ethernet networking, Power Engineering Journal, Volume: 16 , Issue: 6, Dec. 2002 Pages:305 – 311

[28] Su, C.-L., Lu, C.-N., Lin, M.-C., Migration path study of a distribution SCADA system, Generation, Transmission and Distribution, IEE Proceedings- , Volume: 146 , Issue: 3, May 1999 Pages:313 - 317

[29] Cheung, R.W.-L., Yu-Fai Fung, Wireless access to SCADA system, Advances in Power System Control, Operation and Management, 2000. APSCOM-00. 2000 International Conference on, Volume: 2 , 30 Oct-1 Nov, 2000 Pages:553 – 556

[30] Ball, R., Berresford, D.R., Crook, E., Squires, R., Interfacing between SCADA systems and substation communications networks, Developments in Power System Protection, 1993, Fifth International Conference on , 1993 Pages:9 – 12

[31] Slater, A., PC and SCADA based energy management techniques, Advanced SCADA and Energy Management Systems, IEE Colloquium on , 6 Dec 1990 Pages:4/1 - 4/2

[32] Lai, L.L., The impact of new technology on energy management systems and SCADA, Advanced SCADA and Energy Management Systems, IEE Colloquium on , 6 Dec 1990 Pages:1/1 - 1/3

[33] IEEE Recommended Practice for Data Communications Between Remote Terminal Units and Intelligent Electronic Devices in a Substation, IEEE Std 1379-2000 (Revision of IEEE Std 1379-1997),21 September 2000

[34] Curtis, Ken, A., DNP3 Protocol Primer, DNP Users Group, 1 June 2000

[35] Fundamentals of Utilities Communication Architecture, Computer Applications in Power, IEEE, Volume: 14 , Issue: 3 , July 2001 Pages:15 – 21

[36] Bernard, J.-P., Durocher, D., An expert system for fault diagnosis integrated in existing SCADA system, Power Systems, IEEE Transactions on , Volume: 9 , Issue: 1 , Feb. 1994 Pages:548 - 554

[37] Bruce, A., Lee, R., A framework for the specification of SCADA data links, Power Systems, IEEE Transactions on , Volume: 9 , Issue: 1 , Feb. 1994 Pages:560 - 564

[38] Ghoshal, K., Douglas, L.D., GUI display guidelines drive winning SCADA projects, Computer Applications in Power, IEEE , Volume: 7 , Issue: 2 , April 1994 Pages:39 – 42

[39] Gaushell, D.J., Block, W.R., SCADA communication techniques and standards, Computer Applications in Power, IEEE, Volume: 6, Issue: 3, July 1993 Pages: 45 – 50

[40] Chan, E.-K., Ebenhoh, H., The implementation and evolution of a SCADA system for a large distribution network, Power Systems, IEEE Transactions on , Volume: 7 , Issue: 1, Feb. 1992 Pages:320 – 326

[41] Pollet, J., Developing a solid SCADA security strategy, Sensors for Industry Conference, 2002. 2nd ISA/IEEE, 19-21 Nov. 2002 Pages: 148 – 156

[42] Duo Li, Serizawa, Y., Mai Kiuchi, Concept design for a Web-based supervisory control and data-acquisition (SCADA) system, Transmission and Distribution Conference and Exhibition 2002: Asia Pacific. IEEE/PES, Volume: 1, 6-10 Oct. 2002 Pages: 32 - 36 vol.1

[43] Hayashi, H., Takabayashi, Y., Tsuji, H., Oka, M., Rapidly increasing application of Intranet technologies for SCADA (supervisory control and data acquisition system),

Transmission and Distribution Conference and Exhibition 2002: Asia Pacific. IEEE/PES, Volume: 1, 6-10 Oct. 2002 Pages: 22 - 25 vol.1

[44] Xiaodong Zhang, Yun Gao, Guangyuan Zhang, Guangguo Bi, CDMA2000 cellular network based SCADA system, Power System Technology, 2002. Proceedings. PowerCon 2002. International Conference on, Volume: 2, 13-17 Oct. 2002 Pages: 1301 - 1306 vol.2

[45] Zhihao Ling, Jinshou Yu, The design of SCADA based on industrial Ethernet, Intelligent Control and Automation, 2002. Proceedings of the 4th World Congress on, Volume: 4, 10-14 June 2002 Pages: 2786 - 2789 vol.4

[46] Alexander, R.L., Intelligent electronic device (IED) technology SCADA and 3Ø metering, Rural Electric Power Conference, 2002. 2002 IEEE, 5-7 May 2002 Pages: C6 - C6_3

[47] Flowers, T., Houle, B., Refzer, J., Ramanathan, R., Routing SCADA data through an enterprise WAN, Computer Applications in Power, IEEE, Volume: 8, Issue: 3, July

[48] Abbas, H.A. and Mohamed, A.M. (2011) 'Review in the design of web based SCADA systems based on OPC DA protocol', International Journal of Computer Networks, February, Vol. 2, No. 6, pp.266–277, Malaysia

[49] Alfredson, J., Holmberg, J., Andersson, R., Wikforss, M. (2011). Applied Cognitive Ergonomics Design Principles for Fighter Aircraft, D. Harris (Ed.): Engin. Psychol. and Cog. Ergonomics, HCII 2011, LNAI 6781, pp. 473–483.

[50] Arghira, N. et al. (2011) 'Modern SCADA philosophy in power system operation – a survey', University' Politehnica' of Bucharest Scientific Bulletin, Series C: Electrical Engineering, Vol. 73, No. 2, pp.153–166.

[51] Aydogmus, Z. (2009). Implementation of a fuzzy-based level control using SCADA, Expert Systems with Applications, 36 (2009) 6593–6597.

[52] Chakrabarti, S., Kyriakides, E., Bi, T., Cai, D. and Terzija, V. (2009) 'Measurements get together', IEEE PEM, Vol. 7, No. 1.

[53] Chang, R. F., Chang, C. W., Tseng, K. H., Chiang, C. L., Kao, W. S., Chen, W. J. (2011). Structural planning and implementation of a microprocessor-based human–machine interface in a steam-explosion process application, Computer Standards & Interfaces 33 (2011) 232–248.

[54] Di Marzo Serugendo, G. et al. (2005) ‘Self-organization in multi-agent systems’, The Knowledge Engineering Review, Vol. 20, No. 2, pp.165–189, Cambridge University Press.

[55] Fan, R., Cheded, L. and Toker, O. (2005) ‘Internet-based SCADA: a new approach using JAVA and XML’, The Journal of Comput. Control Eng., October, Vol. 16, No. 5, pp.22–26, IET Digital Library.

[56] Jennings, N.R. (2000) ‘On agent-based software engineering’, Journal of Artificial Intelligence, March, Vol. 117, No. 2, pp.277–296, Elsevier Science Publishers.

[57] Jennings, N.R. (2001) ‘An agent-based approach for building complex software systems’, Communications of the ACM, Vol. 44, No. 4, pp.35–41.

[58] Jennings, N.R., Corera, J.M. and Laresgoiti, I. (1995) ‘Developing industrial multi-agent systems’, in Proceedings of the First International Conference on Multi-agent Systems (ICMAS-95), pp.423–430.

[59] Johannsen, G., Alty, J. L. (1991). Knowledge engineering for industrial expert systems, Automatica, Vol. 27, No. 1, pp. 97- 114.

[60] Karnouskos, S. and Colombo, A.W. (2011) ‘Architecting the next generation of service-based SCADA/DCS system of systems’, IECON 2011 – 37th Annual Conference on IEEE Industrial Electronics Society, pp.359–364, 7–10 November.

[61] Kong, J. S., Maute, K., Frangopol, D. M., Liew, L. A., Saravanan, R. A., Raj, R. (2003). A real time human– machine interface for an ultrahigh temperature MEMS sensor–igniter, Sensors and Actuators A, 105 (2003) 23–30.

[62] Leitao, P. and Restivo, F. (2006) ‘ADACOR: a holonic architecture for agile and adaptive manufacturing control’, Computers in Industry, Vol. 57, No. 2, pp.121–130, Elsevier.

APPENDIX A

CLIENT

```
import os

import pandas as pd

import socket

import uuid

import socket

server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

#ip = socket.gethostbyname(socket.gethostname())

port = 8069;

ip = '127.0.0.1';

print ("\nServer_IP:" , ip)

ip_client = '192.168.8.104';

print ("\nClient IP: ",ip_client)

client_port = 4049

print("\nclient_port : ", client_port)

d = pandas.read_csv(r"C:\Users\Ahsan\OneDrive - Higher Education Commission\Fiverr 2\AddressScanScrubbedV2.xlsx"); # path to file + file name

print ("Client port: " ,client_port)

address = (client_ip, port)

print("Address Port: ", address)

server.bind(address)
```

```
server.listen(1)

print "Started Listening on ", ip, ": ", port

client.addr = server.accept()

print "Got a connection from ", addr[0], ": ", addr[1]

while True:

    data = client.recv(1024)

  if (data == "Hello Server" ):

        client.send("Hello client")

 elif (data == "disconnect"):

      client.close()

       break

  else:

       print "invalid data"

    if (data == "disconnect"):

        client.close()

    else:

        print(data)

df = pd.read_csv(r'C:\Users\Ahsan\OneDrive - Higher Education Commission\Fiverr
2\AddressScanScrubbedV2.csv')

dt = df.head(100)

#print(df.iloc[1,:]):

#print((df.shape[0]))
```

```
print ("Mac Address of server: " , hex(uuid.getnode()))

print("*  IP  ***")

print("Version: 3 ")

print("IHL: 8")

print("TOS: 6")

print("TOTAL LENGTH: ", df.shape[0])

print("IDENTIFICATION: ", dt.shape[0])

print("FLAGS: 00")

print("FRAGMENT OFFSET: ", port)

print("TTL: ", df.shape[1])

self.protocol = self.convToDecimal(self.str[8:16])          # Ethernet Protocol Conversion

if self.protocol == 6:

self.protocol = 'TCP'

elif self.protocol == 17:

self.protocol = 'UDP'

print("PROTOCOL: 16" )

print("CHECKSUM: ", df.iloc[:200]);

self.str = self.str[32:]

print("IP SOURCE: ", socket.gethostname());

print("IP DESTINATION: ", ip_client);

print();

def getTCP(self):
```

```
print("*  Evaluation of Ethernet Protocol  ***")

print("Source Port of Ethernet Serial Protocol: ", port);

print("Destination Port of Ethernet Serial Protocol: ", client_port);

self.str = self.str[32:]

print("Sequence Number: ", )

self.str = self.str[32:]

print("Acknowledgement Number: ", self.convToDecimal(self.str[:32]))

self.str = self.str[32:]

print("Header Length: ", df.shape)

print("Reserved: ", )

print("Bits: 2048")

print("Window Size: 5120")

self.str = self.str[32:]

print("Checksum: ")

print("Urgent Pointer:")

def getUDP(self):

print("*  Conversion of User Datagram Protocol for Ethernet Protocol  ***")

print("Source Port of UDP for Ethernet Serial Protocol: ",
self.convToDecimal(self.str[:16]))

print("Destination Port of UDP for Ethernet Serial Protocol: ",
self.convToDecimal(self.str[16:32]))

self.str = self.str[32:]

print("Length: ", self.convToDecimal(self.str[:16]))
```

```
print("Checksum: ", self.convToDecimal(self.str[16:32]))
```

APPENDIX B

SERVER-COMMUNICATION.PY

```
import time

import serial

import socket

from tabulate import tabulate

#####   Read DATA from serial port of Raspberry Pi  #######

ser = serial.Serial(port=None, baudrate = 9600,

                parity=serial.PARITY_NONE,

                stopbits=serial.STOPBITS_ONE,

                bytesize=serial.EIGHTBITS)

time.sleep(1)

try:

   ser.write('Raspberry pi\r\n')

   ser.write('Serial-Ethernet Communication using Raspberry pi\r\n')

   print('Data is transferred')

   while True:

      if ser.inWaiting() > 0:

         data = ser.read()

         print(data)

except KeyboardInterrupt:
```

```
    print("Exiting Program")
except:
    print("Recieving")
finally:
    ser.close()
    pass
#####   Send DATA to server using TCP Communication  #######
client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
client.connect (('192.168.8.64', 8072))
output = 'Raspberry pi send data on server'
client.send (output.encode('utf-8'))
d = tabulate([['command','OFF COntrol Mode'], ['command','OFF COntrol Mode'],
['command','OFF COntrol Mode']]
        , headers=['Commad Response','Control mode'] )
print(d)
client.send(d)
client.send (d.encode('utf-8'))
from_server = client.recv(4040)
f = open('book1.png','rb')
I = f.read(4040)
while (1):
 print ('sending.....')
```

```
client.send(I)

I= f.read(4040)

f.close()

print ("Done Sending")

print (client.recv(4040))

print(from_server)

data_1 = from_server

print (data_1)

client.close()
```

CURRICULUM VITAE

Hamzah Hameed Jasim was born in Salah Ad Din, Iraq on November 26, Iraq. He earned a B.Tech Computer Science from the University of Tikrit. He was accepted in the graduate program in 2017-2018 in Information Technology Altinbas University. Istanbul Campus and will graduate in 2019-2020.