https://www.overleaf.com/project/643e781b4c813857fa68a25d

Modbus

Heansuh Lee
Electronic Engineering
Hochschule Hamm-Lippstadt
Lippstadt, Germany
heansuh.lee@stud.hshl.de

Shihab Ud Doula
Electronic Engineering
Hochschule Hamm-Lippstadt
Lippstadt, Germany
shihab-ud.doula@stud.hshl.de

https://www.overleaf.com/project/643e781b4c813857fa68a25d

***Abstract*—Modbus is a communication protocol widely used in industrial automation systems to facilitate data exchange between devices. It was developed in the late 1970s by Modicon, now Schneider Electric, and has since become a de facto standard for connecting electronic devices in various industries. Modbus has evolved over time and is now available in several variants, including Modbus RTU, Modbus ASCII, and Modbus TCP/IP, which operate on different physical layers and communication mediums. Modbus is known for its simplicity, robustness, and versatility, making it a popular choice for integrating devices from different manufacturers and platforms. This paper provides an overview of Modbus, its history, variants, and applications, as well as its advantages and limitations.**

***Index Terms*—Modbus**

I. Introduction

Modbus is a communication protocol that plays a vital role in industrial automation systems. It was developed by Modicon, now Schneider Electric, in the late 1970s as a means to enable communication between programmable logic controllers (PLCs) and other electronic devices. Since then, Modbus has become one of the most widely used communication protocols in industrial automation systems, and it has evolved to include several variants that operate on different physical layers and communication mediums.

Modbus is a master-slave protocol, meaning that one device acts as the master, initiating communication, while other devices act as slaves, responding to the master's requests. The protocol is known for its simplicity, robustness, and versatility, making it a popular choice for integrating devices from different manufacturers and platforms. Modbus allows for data exchange in real-time and can be used to control devices such as motors, valves, and sensors.

Modbus has several variants, including Modbus RTU, Modbus ASCII, and Modbus TCP/IP. Modbus RTU and Modbus ASCII are both serial communication protocols that operate on different physical layers, with Modbus RTU operating on a two-wire twisted pair cable and Modbus ASCII operating on a single-wire cable. Modbus TCP/IP, on the other hand, is a newer variant that operates on Ethernet networks, providing higher bandwidth and more reliable communication over longer distances.

Modbus has found widespread use in various industries, including manufacturing, oil and gas, water treatment, and building automation. It is used to integrate devices such as sensors, actuators, and controllers into a larger system and to exchange data between different systems. Modbus is also used in supervisory control and data acquisition (SCADA) systems, which are used to monitor and control industrial processes in real-time.

Despite its advantages, Modbus also has some limitations, such as a lack of built-in security features and limited bandwidth compared to other communication protocols. Nevertheless, it remains a popular choice for industrial automation systems due to its simplicity, robustness, and widespread support across different platforms.

In conclusion, Modbus is a communication protocol that has played a significant role in the development and implementation of industrial automation systems. Its simplicity, robustness, and versatility have made it a popular choice for integrating devices from different manufacturers and platforms, and its various variants have allowed it to operate on different physical layers and communication mediums. While it has some limitations, Modbus remains a valuable tool for exchanging data and controlling devices in industrial settings.

II. Concept

Modbus was contrived by Modicon (now a brand of Schneider Electric) in 1978, and it was a revolutionary, monumental networking protocol, as industrial networking has since kicked off. Modbus is the world's first bus protocol unfeignedly used in industrial scenes, first designed for data exchange between PLCs(Programmable logic controllers) [1].

In 1998, Schneider introduced the new generation of Modbus TCP based on TCP/IP. Modbus TCP continues to adopt the Modbus protocol for the application layer, which is simple and efficient; the TCP protocol is deployed in the transport layer and TCP port 502 is used, which is user-friendly and reliable; the IP protocol is deployed in the network layer, because the Internet uses this protocol for addressing, so Modbus TCP can be used not only in local area networks but also in wide area networks and the Internet [2].

Modbus TCP remains to employ all standard Ethernet hardware, along with Ethernet solution TCP/IP Ethernet provides users with more services such as I/O scanning, global data, faulty device replacement, network management, email alarms, clock synchronization and other functions, bringing more additional benefits to users [3].

In 2004, to popularize and promote the distributed application of Modbus on the basis of Ethernet, Schneider has since transferred the ownership of Modbus protocol to the Modbus-IDA (Interface for Distributed Automation) organization, setting the foundation for the future development of Modbus. This marks a commitment to openness (the protocol is free for download and there are no subsequent license fees for using the Modbus or Modbus TCP/IP protocols); by now, there are several Modbus TCP variants in Modbus TCP/IP because of its openness, simplicity, low cost of development, and minimal hardware required to support it, while Modbus ASCII and Modbus RTU are widely used in small embedded devices.

In 2018, the Modbus Organization declared the publication of the Modbus Security protocol, which is a fundamental division in efforts to secure Industrial Control System (ICS) traffic. Secure protocols can extenuate numerous common cyber-attacks, including replay and man-in-the-middle exploits [2].

According to approximate statistics, as of mid-2004, the number of Modbus nodes installed had exceeded 8 million, and 75 percent of the products are non-Schneider products, installed in regions all over the world, which shows its popularity and has become a de facto protocol standard. Although 44 years old, Modbus is still active today in industries of construction, infrastructure and other areas [3].

The great success of Modbus can be attributed to the following three factors:

1) Standardized and open: Users can utilize the Modbus protocol for free and with ease, without paying licensing fees and violating the intellectual property rights.
2) Modbus is a message-oriented protocol that supports a variety of electronic connections, such as RS232, RS422, RS485, etc. It can also be transmitted on a wide range of media, such as twisted-pair cable, fiber optic cable, wireless radio frequency, and so forth. To specify: Unlike many of the field buses, it does not require specific chips and hardware, but adopts completely commercially available regularized components. This guarantees the lowest cost of products employing Modbus.
3) The framework format of Modbus is one of the simplest and most compact of protocols, so to speak: simple, efficient and easy to understand. So it is easy for users to use and vendors to develop. Users and suppliers can access the website to download sample programs, modules, and Modbus tools in various languages to make better use of Modbus.

Although Modbus has been around for a while, the 3 advantages mentioned above still carry over many newer communications protocols available today. They enable the 40-Year-Old Modbus protocol to have a future in newer innovations such as IoT. With a giant amount of instrumentality that it is installed on. A huge quantity of machines and systems have used it and still use it. With that comes a significant community of skilled technicians and professionals who recognize it and, more importantly, trust it with expensive, complex, and large-scaled industrial systems.

Modbus is deliberately flexible and is typically customized for each installation. Hence, a “Mapping Table” or a “Memory Map” is necessitated to know where values are stationed and what they imply in any execution. It is besides extremely lightweight to operate, which is another reason it has not been supplanted by more modern protocols like OPC, Fieldbus, CIP, and Profinet. Besides dealing with sensor data and actuator control, it is used to transmit an extensive amount of diagnostic information as well, which is can be valuable in the age of Machine Learning [4].

For Modbus’ simplicity, many automation devices such as PLCs will remain to be built conveniently with it, with intelligent gateways that conceal the underlying technologies. In the foreseeable future, Modbus will become one of the basics that industries rely on. However, due to the advanced technology bringing inevitable obsolescence of electronic components of older generations, such as RS232 and RS485, PC-based controllers nowadays have a hard time germinating themselves with the requirements of traditional Modbus RTU and ASCII since the connecting cables aren’t as commonly accessible as they were a decade ago. for this reason, the Modbus organization introduced themselves in the cooperation ”Wireless Cooperation Team”(WCT), in order to keep Modbus subsisting in the foreseeable future of technology industry [5].

III. Applications

A. Modbus frame

Modbus is a messaging protocol positioned in the layer of application, which locates at the seventh level of the OSI model, as shown in Fig. 1. The initial version of Modbus was built upon a serial communication method. Later variants were introduced such as Modbus RTU, Modbus ASCII, and Modbus TCP/IP. These additions allow Modbus to pass messages across different buses and networks by changing the packet format or by using TCP/IP protocol networks. Although there are different types of Modbus, they all have the same core element, which is called a Protocol Data Unit (PDU). PDU includes a Function code field and a Data field. A general Modbus frame is built out of an Application Data Unit (ADU), which can be seen as an envelope containing messages, which are PDUs. In Modbus RTU, the Application Data unit is a PDU encapsulated by the address of the target memory and cyclic redundancy checksum (CRC). For Modbus TCP/IP, the Application Data Unit consists of the MBAP header and the PDU. Further discussions on Modbus variants can be found in the following sections [13] [15].

B. Modbus Transaction

Modbus data transfer is established in a Server-Client network. Modbus Client is the device that requests information and Modbus Server is the device that provides the information. The transaction process completes in two steps, request and response. Modbus Client initiates the request for data by delivering ADUs which contain function codes and data requests.

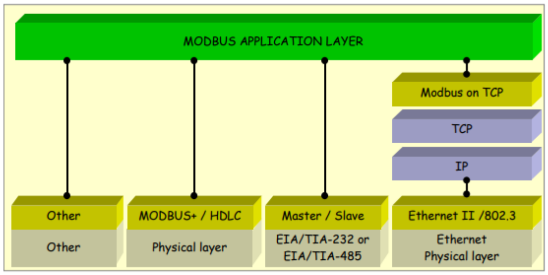
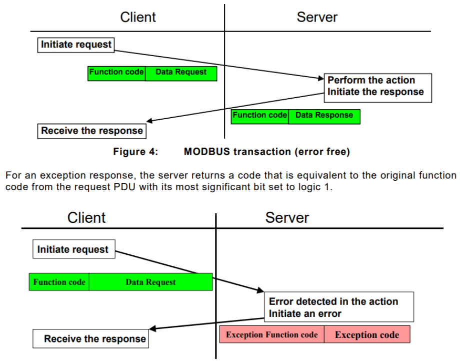


Fig. 1. Modbus in OSI architechture

The function code is a one-byte Modbus request sent by the master to specify the memory type to be accessed and actions to be performed, while the Data field provides up to 252 bytes of the necessary information required by the function codes e.g. discrete and register addresses, the number of items in the field and the actual bytes of data carried. If the Modbus server successfully receives the request, it initiates a response to the client with the requested data. However, whenever an error occurs between the two ends, the Modbus client does not get the requested data but receives an exception response [11].

The Modbus Transaction state diagram demonstrates the transaction processing in Modbus Server in Fig. 2. Two types of responses are generated based on the result of the transaction. In an error-free Modbus response, the response function code is identical to the request function code. In the case of an exception response, an exception code is created to provide the client with information concerning the error and the cause. The exception function code is a combination of a request function code with the most significant bit set to a logic 1 and a hexadecimal number 0X80. Basic Modbus function codes and exception codes are explained later in this paper.

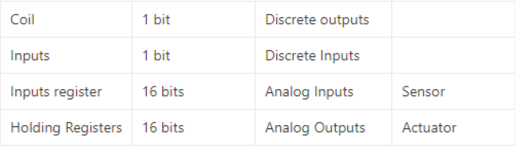


For an exception response, the server returns a code that is equivalent to the original function code from the request PDU with its most significant bit set to logic 1.

Fig. 2. Modbus transaction

C. Modbus Data type

Originally, Modicon developed Modbus with the intention of establishing a communication protocol for Programmable Logic Units (PLCs). Intelligent devices PLCs, Remote Terminal Units (RTUs), and loop controllers possess their own memories to store data, programs, or backups. A Modbus compliant device would have a portion of its memory dedicated to Modbus, and this area is described as Modbus Memory Area. In most cases, the Modbus Memory Area is divided into 4 separate subareas and all the areas are composed of memory blocks(see fig.3). Coils and Discrete inputs consist of one single-bit memory block, whereas Input registers and Holding registers are both 16-bit registers. These four areas have different types of access to memory storage. Discrete Inputs and Input registers are READ-ONLY, while Coils and Holding registers do READ and WRITE [15].



Coil	1 bit	Discrete outputs	
Inputs	1 bit	Discrete Inputs	
Inputs register	16 bits	Analog Inputs	Sensor
Holding Registers	16 bits	Analog Outputs	Actuator

Fig. 3. Modbus data type

To describe data more precisely or on a larger scale, the 32-bits data type is sometimes used in memory reading and memory writing. However, they do not fit in 16-bit registers, therefore some formats are created to handle data like floating-point numbers and long integers. For example, a 32-bit floating-point will be sent as a pair of registers. Big-endian is the most common format, where the ”big end” (most significant byte) is stored before the less significant byte. In contrast, Little-endian reverses the bytes, with the least significant byte being placed before the most significant byte. The endianness is beyond the scope of this paper, it is only provided as additional information only.

D. Modbus Function code

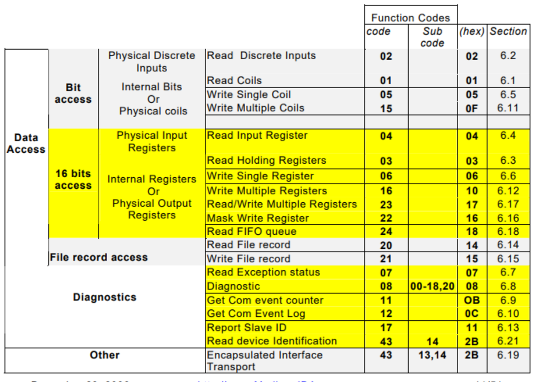
Modbus function codes define the actions to be taken in a data transaction and are divided into three categories. They are: Public Function Codes, User-Defined Function Codes and Reserved Function Codes.Public Function Codes are defined and validated by the MODBUS.org community, they are well documented and are guaranteed to be unique.For specific implementation or requirements, users can choose and implement User-Defined Function Codes that range from 65 to 72 and 100 to 110 decimal. The Reserved Function Code is limited to the private use of certain companies and their products. A list of public function codes can be found in fig.4. [13]

E. Modbus Exception code

As mentioned in the previous section, exception responses result from errors occurring in Modbus transactions.

IV. Modern Applications in Industry

Modbus protocol is used widely by many manufacturers throughout many industries. Modbus is typically used to transmit signals from instrumentation and control devices back to a main controller or data gathering system, for example a system that measures temperature and humidity and communicates the results to a computer.



				Function Codes			
				code	Sub code	(hex)	Section
Data Access	Bit access	Physical Discrete Inputs	Read Discrete Inputs	02		02	6.2
		Internal Bits Or Physical coils	Read Coils	01		01	6.1
			Write Single Coil	05		05	6.5
			Write Multiple Coils	15		0F	6.11
	16 bits access	Physical Input Registers	Read Input Register	04		04	6.4
		Internal Registers Or Physical Output Registers	Read Holding Registers	03		03	6.3
			Write Single Register	06		06	6.6
			Write Multiple Registers	16		10	6.12
			Read/Write Multiple Registers	23		17	6.17
			Mask Write Register	22		16	6.16
			Read FIFO queue	24		18	6.18
	File record access		Read File record	20		14	6.14
			Write File record	21		15	6.15
Diagnostics			Read Exception status	07		07	6.7
			Diagnostic	08	00-18,20	08	6.8
			Get Com event counter	11		OB	6.9
			Get Com Event Log	12		0C	6.10
			Report Slave ID	17		11	6.13
			Read device Identification	43	14	2B	6.21
Other			Encapsulated Interface Transport	43	13,14	2B	6.19

Fig. 4. Modbus public function code

Modbus protocol is an open protocol - therefore, Modbus protocol is usually backed by a combination of corporations, user groups, professional societies, and governments. This provides users with a much wider choice of devices or systems that can be utilized to meet specific applications. This can include support by multiple manufacturers (Siemens, Alley-Bradley, ABB), software vendors (Modbus Utility, Simply Modbus Slave, BaseBlock), and install/service organizations (MOXA, molex, ProSoft), active community groups for support, the ability to stay current and add capabilities in the future.

Modbus communicates over several types of physical media: over Ethernet, RS-485, RS-422 and RS-232. It works more efficiently and better in a descending order.

RS-232 was used first, but it moved on to RS-485 eventually as time passed. It allowed longer distances, higher speeds and the possibility of multiple devices on a single multidrop network. Master/Slave Modbus communication over serial RS-485 physical media showing two-wire transmit and receive connections. They are simple interfaces. The Modbus messages are sent in plain form over the network, and the network will be dedicated to only Modbus communication.

Modbus is able to function on both point to point and multidrop networks. Modbus devices communicate using a master/slave (client-server for Ethernet) technique in which only one device can initiate transactions (called queries).

Communication with slave devices or master PLCs or computers can be accomplished with Modbus simulator software to run on your personal computer. The connection can be serial or Ethernet, and in the form of a master or slave. The software will allow you to perform all of the Modbus Protocol Communication function code to simply read or write to an existing slave. You can set up on PC to run the slave simulation software, and another to run the Master simulation software.

There are several companies providing products and support to help with protocol communication, including ProSoft Technologies, HMS Industrial Communication and MOXA.

Baud rate, with the units of bauds per second (BPS) (e.g. 9600, 38400 per second), data length of 8, parity bits (serial communication can be unstable) and stop bits should be taken into account when using the applications.

Fig. 5. Modbus hardware from RT Automation.

Modbus protocol has new products from **RT Automation** that deal with MQTT (Message Queueing Telemetry Transport) Industrial Protocol gateways, which enable factory floor data to be transmitted to cloud applications services. The product from Real Time Automation can move data between up to 32 Modbus RTU server devices and an MQTT applications, which can include AWS (Amazon Web Server) or Microsoft Azure [7].

Fig. 6. Modbus hardware from CLICK PLUS PLCs.

CLICK PLUS PLCs combine the simplicity of the original CLICK PLC with advanced features, including Wi-Fi capability, MQTT communication, data logging, and mobile access. These new products show that there are more flexibility in data transmission through wireless network, which makes it easier for the industry making use of slave devices that rely heavily on serial communication with Modbus [8].

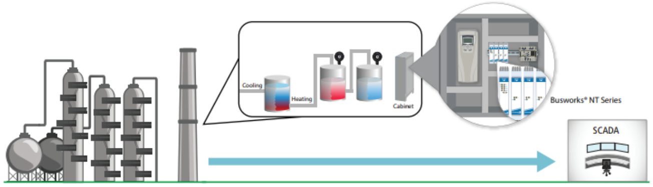


Fig. 7. A diagram from Acromag on how Modbus protocol can be used in the chemical process.

Acromag allows complicated temperature measurement points that are distributed throughout the chemical process. Since the SCADA system license fee is determined by the number of IP addresses on their network, this can be tackled by using only one IP address with Acromag's BusWorks NT Series products [9].

Fig. 8. Modbus hardware from Acromag.

Modbus RTU devices can also be connected to a wireless network in the industry. WirelessHART applications and networks can be connected to Modbus device using a product from **Fieldbus International** [10].

V. Implementation

A. Background, Use Case, and Requirements

Modbus can form centralized and decentralized control system with remote terminal unit (RTU), and it is popularly utile for Supervisory control and data acquisition (SCADA) systems, for example, environmental temperature, luminance and humidity measurement monitoring system. In our simulation, we therefore simulates a scenario where there is a monitoring and controlling client, connected to a sensory server via Modbus RTU. In the SCADA system of our simulation, there is a monitoring "Client" and a "server" used for temperature detection: the "client" can request detection results from the server.

Below are the requirements elicited for the Modbus RTU implementation:

"Client" performs message conveying and receiving from multiple "servers" via Modbus RTU, while "server" receives messages directly from "client". During which, address confirmation is firstly done between the client and the requested server. Afterwards, depending on the client's commands, acquisition of detected and collected data can be achieved, as well as the manipulation of multiple devices' performance. There are up to 247 server devices accessible in Modbus RTU, and no security is necessitated to be used. The speed of Modbus RTU can reach up to 38400 bits/s, nevertheless, it is usually not a crucial property for Modbus-affiliated devices.

B. Software Implementation

In order to implement Modbus RTU, we are using 3 softwares for the simulation: Kepware OPC Server to establish the server, Modbus Poll as a master that will send information, and lastly the WinCC for the simulation part to show that it has received the signals from the Kepware OPC server.

VI. Evaluation & Conclusion

The Modbus protocol supports different transmission modes, including Modbus RTU (Remote Terminal Unit). Modbus RTU is a binary implementation of the protocol and is commonly used over serial communication interfaces, such as RS-485 or RS-232. It uses a master-slave architecture, where the master device initiates and controls communication with one or more slave devices [3] [5].

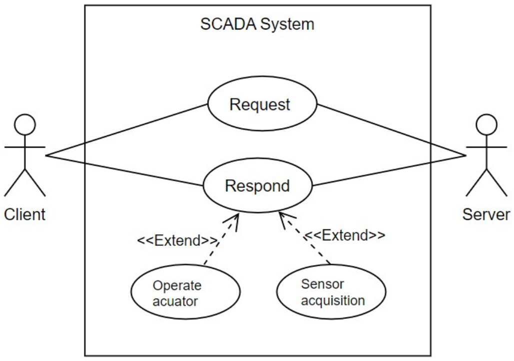


Fig. 9. Use case diagram

A. Strengths of Modbus and Modbus RTU

- Simplicity: Modbus has a simple and straightforward design, making it easy to implement and understand. This simplicity helps in reducing development time and costs.
- Versatility: Modbus is vendor-neutral and can be used with a wide range of devices from different manufacturers, fostering interoperability and flexibility.
- Efficiency: Modbus RTU uses binary encoding, resulting in compact message sizes, faster transmission speeds, and reduced bandwidth requirements. This efficiency is particularly useful when working with low-speed or noisy communication channels.
- Wide adoption: Modbus has been used extensively in the industrial automation sector for decades, creating a large ecosystem of devices, software libraries, and expertise.

B. Shortcomings of Modbus and Modbus RTU

- Limited security features: Modbus was developed at a time when industrial networks were relatively isolated. As a result, it lacks robust security features, making it vulnerable to attacks. Additional security measures need to be implemented to ensure data integrity and confidentiality.
- Lack of real-time capabilities: Modbus is not inherently designed for real-time applications. Although it can achieve reasonable response times, it may not be suitable for highly time-critical systems.

C. Alternatives to Modbus

- Profibus: Profibus is a widely used industrial network protocol that offers high-speed communication and comprehensive diagnostic capabilities. It supports both process automation (Profibus DP) and factory automation (Profibus PA) applications.

- Ethernet/IP: Ethernet/IP combines the Ethernet protocol with the Common Industrial Protocol (CIP). It offers real-time communication, high bandwidth, and compatibility with standard Ethernet infrastructure, making it suitable for industrial applications.

D. Future of Modbus

Despite the emergence of newer protocols, Modbus continues to be widely used due to its legacy support and vast installed base. However, to meet the demands of modern industrial systems, efforts are being made to enhance Modbus with additional features, such as improved security, better real-time capabilities, and support for Internet of Things (IoT) integration. Some variants, like Modbus TCP/IP, have already been developed to facilitate integration with Ethernet-based networks.

In conclusion, Modbus and Modbus RTU have been instrumental in industrial automation for several decades. Their strengths lie in simplicity, versatility, efficiency, and wide adoption. However, they do have weaknesses in terms of security and real-time capabilities. Alternatives like Profibus and Ethernet/IP offer different features and may be suitable for specific applications. The future of Modbus lies in its evolution to meet the changing requirements of modern industrial systems, incorporating enhanced security measures and better integration with IoT technologies.

References

[1] ”The Control Techniques Drives and Controls Handbook.” Drury, B. and Control Techniques (Firme) and Institution of Electrical Engineers. isbn=9781601190710. IEE power and energy series. 2001. Institution of Electrical Engineers

[2] ”Modbus Press Releases.” Modbus Organization. https://modbus.org/press.php

[3] Phan, Raphael C-W. ”Authenticated modbus protocol for critical infrastructure protection.” Ieee transactions on power delivery 27.3 (2012): 1687-1689.

[4] Tunkkari, Jesper. ”Mapping Modbus to OPC Unified Architecture.” (2018).

[5] Mohagheghi, Salman, J. Stoupis, and Z. Wang. ”Communication protocols and networks for power systems-current status and future trends.” 2009 IEEE/PES Power Systems Conference and Exposition. IEEE, 2009.

[6] Hui, Li, Zhang Hao, and Peng Daogang. ”Research and Application of Communication Gateway of EPA and MODBUS/TCP.” 2013 5th International Conference and Computational Intelligence and Communication Networks. IEEE, 2013.

[7] ”MQTT Industrial Protocol Gateways Create an Easy Way for Manufacturers to Move Factory Floor Data to Cloud Application Services.” Real Time Automation, Inc. https://modbus.org/member_docs/RTAutomation-10-27-21.pdf.

[8] ”CLICK PLUS Serial Communication Option Slot Module and 2-Slot CPUs from AutomationDirect.” AutomationDirect. https://modbus.org/member_docs/AutomationDirect-CLICK%20PLUS%20DCM%20&%202%20slot%20cpu%20-%20PR-November%202021.pdf

[9] ”Application Note: How to Measure Temperature with a High-density Ethernet I/O Solution.” Acromag. https://modbus.org/member_docs/Acromag-How-to-Measure-Temperature-with-a-High-density-Ethernet-IO-Solution-Application-Note.pdf.

[10] ”An affordable and straightforward solution for connecting Modbus RTU devices to a wireless network.” Fint. https://modbus.org/member_docs/FINT_Press%20Release_%20T910.pdf/

[11] Fovino, Igor Nai, et al. ”Design and implementation of a secure modbus protocol.” International conference on critical infrastructure protection. Springer, Berlin, Heidelberg, 2009.

[12] Goldenberg, Niv, and Avishai Wool. ”Accurate modeling of Modbus/TCP for intrusion detection in SCADA systems.” international journal of critical infrastructure protection 6.2 (2013): 63-75.

[13] Thomas, George. ”Introduction to the modbus protocol.” The Extension 9.4 (2008): 1-4.

[14] Kuang, Yucong. ”Communication between PLC and Arduino based on Modbus protocol.” 2014 fourth international conference on instrumentation and measurement, computer, communication and control. IEEE, 2014.

[15] Kuang, Yucong. ”MODBUS APPLICATION PROTOCOL SPECIFICATION V1.1b”,https://modbus.org/, December 28, 2006.