



Thu Dau Mot University  
Journal of Science

ISSN 2615 - 9635

journal homepage: [ejs.tdmu.edu.vn](http://ejs.tdmu.edu.vn)



## Control and communication between PLC S7-1200 and ATV310 drive via modbus protocol

by *Phạm Hồng Thanh, Nguyễn Thành Đoàn (Thu Dau Mot University)*

**Article Info:** Received May 25th, 2020, Accepted Aug. 20th, 2020, Available online Sep. 15<sup>th</sup>, 2020

Corresponding author: [thanhph@tdmu.edu.vn](mailto:thanhph@tdmu.edu.vn) (Đặng Hồng Thanh)

<https://doi.org/10.37550/tdmu.EJS/2020.03.066>

### ABSTRACTS

*Variable Frequency Drives (VFDs) are electronic power controllers that allow for accurate control of the speed of alternating current (AC) induction motors that used in many kinds of machines including fans, pumps and compressors. These motors are used in most heating, ventilation and air-conditioning (HVAC) systems and account for a significant percentage of the total HVAC energy consumption. More efficient operation of these motors using VFDs can result in significant energy savings. Besides, the communication between VFDs and PLC for Supervisory Control And Data Acquisition (SCADA) is also important. Modbus protocol has many kinds as RS485, RTU, Profinet,...In this paper we will present the communication between a PLC Siemens S7-1200 and an ATV310 Drive (Schneider) via Modbus RTU protocol which is supported.*

**Keywords:** *Modbus, Schneider, Siemens, industry communication protocol*

### 1. Introduction

Some of the largest opportunities to save energy and reduce operating costs in buildings and industrial facilities come from optimizing electric motor systems (Concordia and Ihara, 1982). The key is to choose the right-sized, energy-efficient motor and to integrate it into an optimized drive power system.

Schneider ATV310 is a compact-sized drive that is diverse and powerful in terms of control features. The use of drives ATV310 helps to tackle a good variety of purposes and different needs. Moreover, with integrated intelligent functions, Modbus RTU communication, ATV310 allows users to do more than expected compared to other common open loop controllers.

The Modbus RTU protocol is an open protocol, which uses a RS-232 or RS485 physical transmission line and a Master-Slave model (Hittanaga, Ramesh, Kumar and Mahadeva, 2017). This is a protocol widely used in many fields such as BMS (Building Management Systems), automation, industry, electricity... (Geogre and Champa, 2018). You'll probably wonder why this Modbus protocol is so common, where to go, hitting on what device is almost also this protocol? Well, the answer to the above question must just wrap in few words: stable-simple-easy to use.

Modbus is considered the operational communication protocol on the "Application" floor, which provides the ability for Master/Slave communication between connected devices through buses or networks (Robinson, 2017; You and Ge, 2019). On the OSI model, Modbus is placed in grade 7 (Pavlou, McCarthy, Bhatti and Knight, 1995). Modbus is defined as a protocol that works under "Q/A" and uses the corresponding "function codes" to inquire (Luo, Zou, Jiang, Gao, Jioa and Sun, 2019).

In this paper, the system with the Schneider-ATV310, a Siemens-PLC S7-1200 connected via Modbus RTU to controlled and supervised an 1HP 3 phases motor. The next part was referred to theoretical of Modbus RTU protocol and specify of ATV310 Modbus function. After that, the paper would be presented technique to setup the communication between PLC S7-1200 and ATV310 Drive via Modbus to control, supervise parameters shown on the SCADA monitor. The conclusion and future work will be present in the last part of this paper.

## **2. Theoretical principle**

### ***Modbus RTU***

Modbus RTU is an open serial protocol derived from the Master/Slave architecture originally developed by Modicon (now Schneider Electric). It is a widely accepted serial level protocol due to its ease of use and reliability. Modbus protocol the grandfather of industrial networking. It truly is as old as the hills and has the whiskers to prove it. In today's age of Internet connectivity and Web Services, Modbus' unconnected message and simple request-response communication structure are almost quaint. Almost as old as the first Programmable Logic Controller. Modbus is used widely by many manufacturers

throughout many industries. Modbus is typically used to transmit data from control instrumentation to a logic controller or a system for archiving data.

The Modbus RTU protocol uses a Master/Slave technique to communicate between devices. Meaning, any application that utilizes the Modbus RTU protocol will have a Modbus Master and at least one Modbus Slave. A Modbus Master is typically a host supervisory that will communicate with one or more Modbus Slave devices.

In RTU mode, messages start with a silent interval of at least 3.5 character times. This is most easily implemented as a multiple of character times at the baud rate that is being used on the network (shown as T1–T2–T3–T4 in the figure below). The first field then transmitted is the device address.

The allowable characters transmitted for all fields are hexadecimal 0–9, A–F. Networked devices monitor the network bus continuously, including during the ‘silent’ intervals. When the first field (the address field) is received, each device decodes it to find out if it is the addressed device. Following the last transmitted character, a similar interval of at least 3.5 character times marks the end of the message. A new message can begin after this interval.

The entire message frame must be transmitted as a continuous stream. If a silent interval of more than 1.5 character times occurs before completion of the frame, the receiving device flushes the incomplete message and assumes that the next byte will be the address field of a new message.

Similarly, if a new message begins earlier than 3.5 character times following a previous message, the receiving device will consider it a continuation of the previous message. This will set an error, as the value in the final CRC field will not be valid for the combined messages. A typical message frame is shown below (Robinson, 2017).

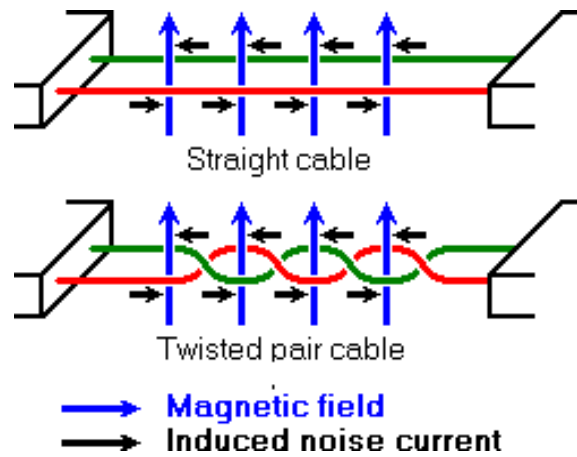
START	ADDRESS	FUNCTION	DATA	CRC CHECK	END
T1-T2-T3-T4	8 BITS	n x 8 BITS	16 BITS	16 BITS	T1-T2-T3-T4

**Figure 1.** Typical Message Frame

### **RS 485**

RS485 is the most versatile communication standard in the standard series defined by the EIA, as it performs well on all four points. That is why RS485 is currently a widely

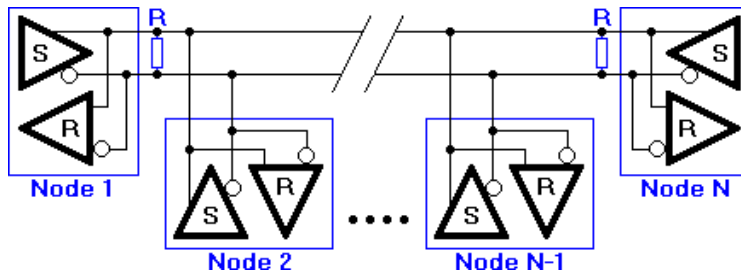
used communication interface in data acquisition and control applications where multiple nodes communicate with each other (Xinjian, 2005; Kumar, 1995)



**Figure 2.** RS485 Communication Interface

In the picture above, noise is generated by magnetic fields from the environment. The picture shows the magnetic field lines and the noise current in the RS485 data lines that is the result of that magnetic field. In the straight cable, all noise current is flowing in the same direction, practically generating a looping current just like in an ordinary transformer. When the cable is twisted, we see that in some parts of the signal lines the direction of the noise current is the opposite from the current in other parts of the cable. Because of this, the resulting noise current is many factors lower than with an ordinary straight cable.

Network topology is probably the reason why RS485 is now the favorite of the four mentioned interfaces in data acquisition and control applications. RS485 is the only of the interfaces capable of internetworking multiple transmitters and receivers in the same network. When using the default RS485 receivers with an input resistance of 12 k $\Omega$  it is possible to connect 32 devices to the network. Currently available high-resistance RS485 inputs allow this number to be expanded to 256. It is the reason why RS485 is so popular with computers, PLCs, micro controllers and intelligent sensors in scientific and technical applications (Xinjian, 2005; Chen, Jin and Chen, 2016).







**Figure 3.** RS485 Network Topology

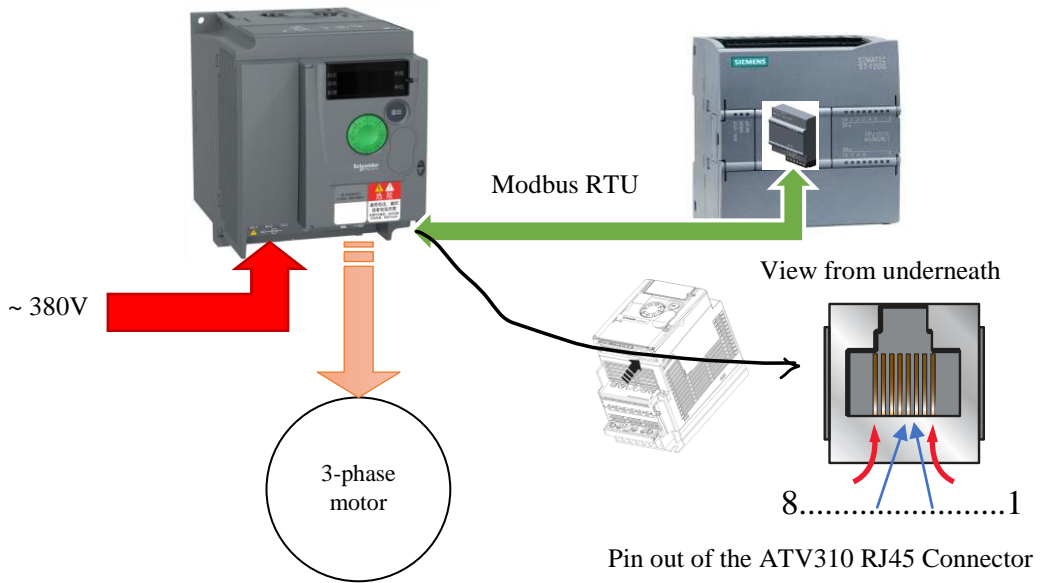
### 3. Hardware configuration and installation

#### *Hardware Configuration*

In this paper, we proposed a system included 1 PLC S7-1200 DC/DC/DC and an ATV310 Drive to drive a motor 0.75KW-3 phases 380V 50Hz in open loop. The Equipment list is below Figure 4.

 <p><b>PLC S7-1200 DC/DC/DC</b>                      Digital Input: 8                      Digital Output: 6                      Analog Input: 2                      HSC support                      Bit Memory: 4096 bytes</p>	 <p><b>CB 1241</b>                      Max Length transmission: 1000m                      Support: Modbus RTU, USS, ASCII, Freeport</p>
 <p><b>Schneider ATV310</b>                      Power: 3 kW                      Input Voltage: 3 phases 220v 50/60 Hz                      Output Voltage: 3 phases 0-220 V                      output Frequency: 0-1500 Hz</p>	 <p><b>Motor 3 phase</b>                      Power: 0.75KW                      Voltage: 3 phases 380V 50Hz</p>

**Figure 4.** Equipment



**Figure 5.** Equipment Connections

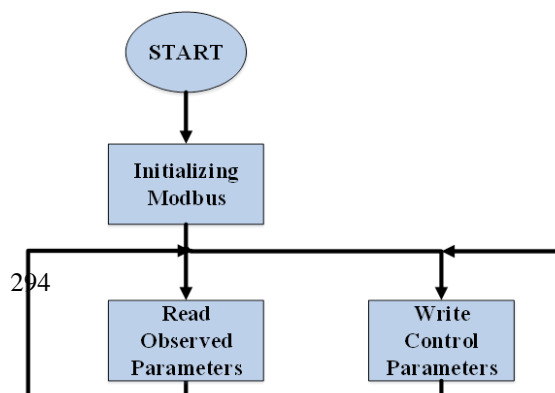
The communication Modbus protocol with RJ45 Interface would be connected to CB 1241 Module with a pair twisted cables as table as follows instruction (Electric, 2014a):

Pin	Signal
1	-
2	-
3	-
4	D1 (1)
5	D0 (1)
6	-
7	-
8	-

**Figure 6.** Pin Signal

**Algorithm Flowchart**

Initially when starting the program, the PLC initial-lizing Modbus. Then the program will jump into the loop of reading and writing data. Without the change of control para-meters, the PLC will continuously read data such as voltage, current, frequency and

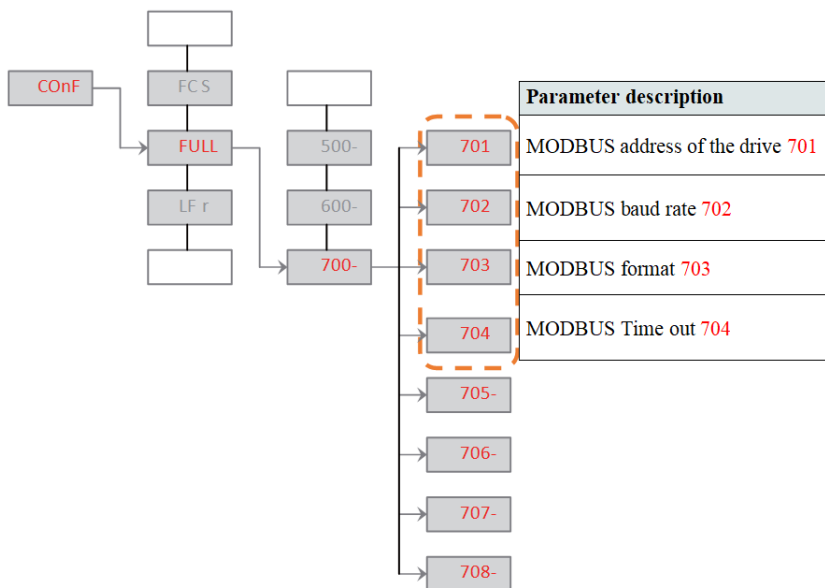


speed... However, when any control parameter is changed, the PLC immediately stops reading and advances data into the drive. After PLC determined that the data is finished to the drive, the new evolution is returned to the next read and displayed on the SCADA monitor.

**Figure 7.** Algorithm Flowchart of Modbus Communication Program

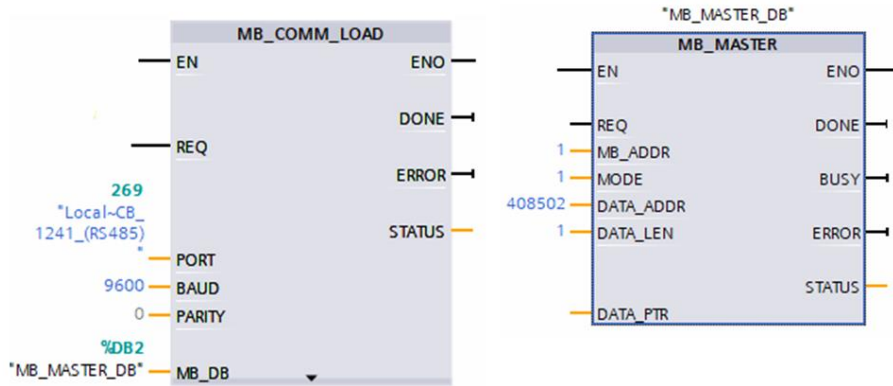
**Communication Drive Parameters Configuration**

We must configure the communication parameters on ATV310 before it can be connected to the PLC. The parameters from 701 – 704 can be configured by button on drive (Electric, 2014a). This menu is accessible as follows Figure 8.



**Figure 8.** Communication Drive Parameters Configuration

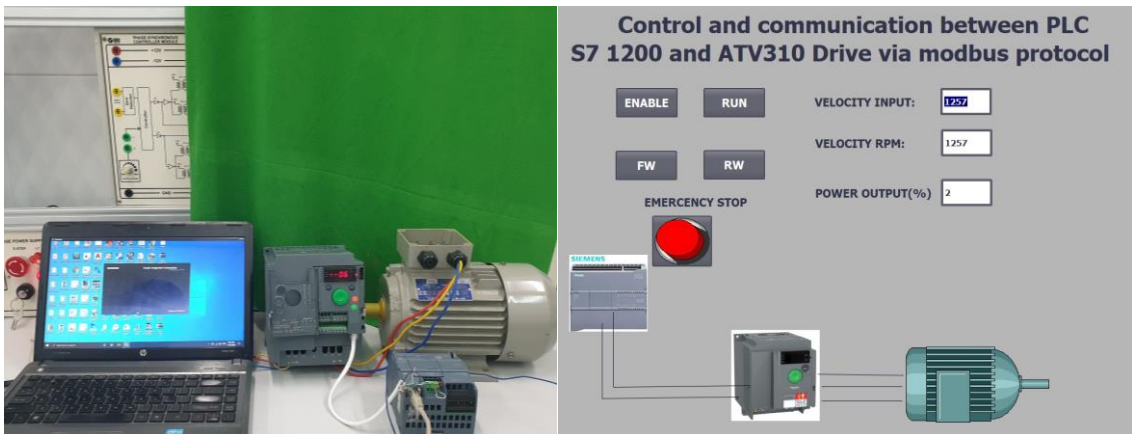
After done, the drive was ready to be full controlled by PLC via Modbus connection by parameters logic address referred in (Electric, 2014b). In PLC program networks, the main function blocks and parameters were set as follow picture for connecting with ATV310 drive via Modbus RTU protocol.



**Figure 9.** Modbus Function Block in PLC Program Network

#### 4. Result and conclusion

We had established the wire-connection of a PLC S7-1200, an ATV310 Drive and an Electric Motor for testing with PLC program for Modbus RTU and SCADA simulation with TIAPortal V14 of Siemens. The final production was shown in the below picture.



**Figure 10.** Demonstration Equipment and SCADA Simulation with TIAPortal V14

The result showed that with a simple hardware of communication of Modbus protocol, PLC can fully controlled and collected data from the Drive beside the HMI monitor of SCADA system also could supervise all data we need and control many functions of the Drive easily



by programming. This research was valuable in automation field also it could be applied in teaching the industry network protocol subject at Thu Dau Mot University.

In future, we will establish a complex Modbus network with many kinds of equipment such as multi-meter, many Drives,...to evaluate the ability of Modbus RTU. We also may be present a research about other Industry Network Protocol.

## References

- C. Concordia and S. Ihara (1982). Load representation in power system stability studies. *IEEE transactions on power apparatus and systems*, 4, 969-977.
- K. N. Hittanagi, M. Ramesh, K. R. Kumar, and S. Mahadeva (2017). PLC based DC drive control using Modbus RTU communication for selected applications of sugar mill. *International Conference on Circuits, Controls, and Communications (CCUBE)*, 80-85: IEEE.
- S. George and H. Champa (2018). FPGA based adaptive implementation of Data acquisition with Modbus RTU protocol. *Perspectives in Communication, Embedded-systems and Signal-processing-PiCES*, 2(4), 86-87.
- D. T. Robinson (2017). Modbus Monitoring for Networked Control Systems of Cyber-Defensive Architecture. Howard University,
- W. You and H. Ge (2019). Design and Implementation of Modbus Protocol for Intelligent Building Security. *IEEE 19th International Conference on Communication Technology (ICCT)*, pp. 420-423: IEEE.
- G. Pavlou, K. McCarthy, S. Bhatti, and G. Knight (1995). The OSIMIS Platform: Making OSI Management Simple, in *Integrated Network Management IV: Proceedings of the fourth international symposium on integrated network management, 1995*, A. S. Sethi, Y. Raynaud, and F. Faure-Vincent, Eds. Boston, MA: Springer US, pp. 480-493.
- Z. Luo, F. Zuo, Y. Jiang, J. Gao, X. Jiao, and J. Sun (2019). Polar: Function Code Aware Fuzz Testing of ICS Protocol. *ACM Transactions on Embedded Computing Systems (TECS)*, 18(5), 1-22.
- L. G. Z. Xinjian (2005). How to Improve RS485 Communication Reliability in Measure and Control System. *Chinese Journal of Scientific Instrument*, p. S1.
- V. A. Kumar (1995). Overcoming data corruption in RS485 communication. *International Conference on Electromagnetic Interference and Compatibility (INCEMIC)*, pp. 9-12: IEEE.
- K. Chen, Z. Jin, and H. Chen (2016). Effect of common-mode interference on communication performance of a motor drive system. *IEEE Vehicle Power and Propulsion Conference (VPPC)*, pp. 1-6: IEEE.
- S. Electric (2014). *ATV310 Modbus Communication Manual*. Schneider Electric.
- S. Electric (2014). *ATV310 - Communication Parameters*, ed: Schneider Electric.

