24th International Conference
ENGINEERING MECHANICS 2018
Svratka, Czech Republic, May 14 –17, 2018
Paper **#231**, pp. 205–208, doi: 10.21495/91-8-205

IMPROVEMENT OF PLCS PROGRAMMING TO OPERATE IN AN MODBUS INDUSTRIAL NETWORK

K. Dzierżek*, M. Rećko**

Abstract: *This article presents an improvement in the programming of a ModBus industrial network. It is used for data exchange amongst Programmable Logic Controllers (PLCs). Paper also presents Master and Slave controllers' setup process. It proposes authors' own, simplified, method of GE INTELLIGENT PLATFORMS PLCs programming to work with ModBus network.*

Keywords: PLC, industrial networks, PLC programming, ModBus network

1. Introduction

ModBus protocol was developed in 1978 by the Modicon company. It is an open protocol and can be implemented without a license fee. Modbus is a Master-Slave network, where:

- Master – the device that initialises transmission,
- Slave – the device corresponding to the initialisation and commands.

The communication between controllers can be executed by means of numerous communication protocols. ModBus is the most universal one, implemented widely by many manufacturers of PLCs. Seems natural that presented simplification of programming would be based on this protocol. Modicon created it as an open protocol. Modbus is used by most producers for asynchronous data exchange between measuring system and control devices. Modbus network has found broad application in industrial applications with low requirements regarding speed and frequency of data transfer. The network works with low data transfer speed, from 1.2 Kb/s up to 115.2 Kb/s over restricted distance. The solution presented in this paper allows easier and quicker programming of industrial network controllers during tests and experiments of their performance. It is especially useful for research, eliminating the need for configuration of industrial networks done by paid professionals. PLC controllers use conventional, lengthy, method. Our new, streamlined solution is enabled thanks to use of parametrized function blocks.

2. Controllers' setup

It is required to setup Master and Slave controllers (Fig. 1-3) to enable communication. The controller needs correct communication port configuration (GE, 2009, 2015):

- port mode (RTU),
- port type (master, slave),
- data rate (od 1.2 Kb/s do 115.2 Kb/s),
- flow control (hardware, none)
- parity (none, odd, even),
- station address (1-247),
- duplex mode (2 wire).

* Kazimierz Dzierżek, DSc, PhD, Eng.: Faculty of Mechanical Engineering, Bialystok University of Technology, str. Wiejska 45c; 15-351, Bialystok; PL, k.dzierzek@pb.edu.pl

** Maciej Rećko, MSc, Eng.: Faculty of Mechanical Engineering, Bialystok University of Technology, str. Wiejska 45c; 15-351, Bialystok; PL, maciej.recko@outlook.com

One ModBus network facilitates connection amongst 1 Master and up to 247 Slave controllers. Stations can be connected by means of three cable types:

- point-to-point network connection cable,
- 4-wire multidrop network connection cable,
- 2-wire multidrop network connection cable.

Some controller types do not enable port two configurations as RTU Master, even though being ModBus compliant. In such case, it is required to configure the port using sending configuration data using COMMREQ command as shown in Table 1. Following controller setup, it is time to send prepared configuration. During Slave controllers programming it is required to erase their control program (Dzierzek, 2007; GE, 2003).

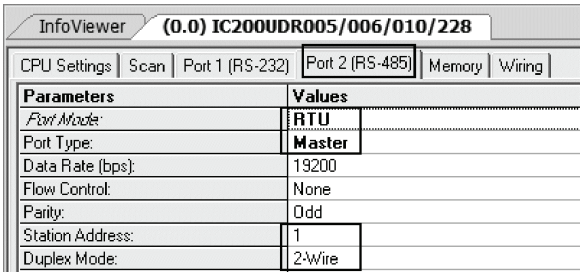

Fig. 1: VersaMax Micro (Master conf.)

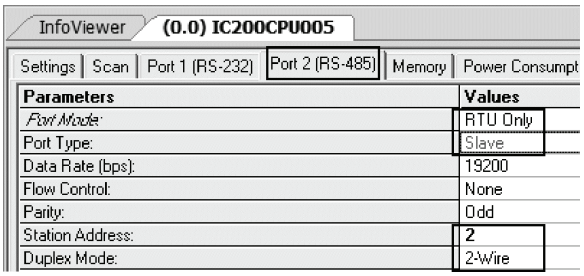

Fig. 2: VersaMax (Slave conf.)

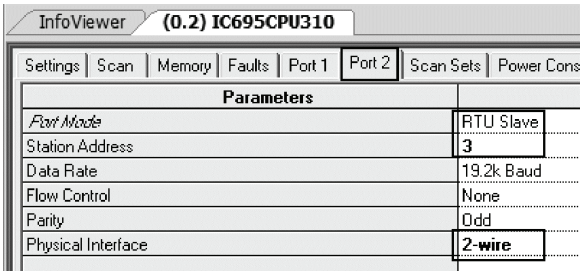

Fig. 3: PACSystems RX3i (Slave conf.)

Tab. 1:Initialize RTU Master Port (GE, 2000, 2003)

Location	***Value***	***Description***
Word 1	15	Port Setup Command/Data Block Length in words. The Data Block portion starts at Word 7.
Word 2	0	NOWAIT Mode (required)
Word 3	8	Status Word Memory Type (8 - %R)
Word 4	0	Status Word Address (0 - %R1)
Word 5	0	Ignored
Word 6	0	Ignored
Data Blok Area		
Word 7	65520	Command – Port Setup (FFF0)
Word 8	3	Protocol (3 - Modbus RTU)
Word 9	1	Mode (1 - Master)
Word 10	6	Data Rate (6 - 19200)
Word 11	1	Parity: 0 = none, 1 = Odd, 2 = Even
Word 12	1	Flow Control: 0 = hardware, 1 = none
Word 13	0	Ignored
Word 14	0	Response message time-out (0 - 8s)
Word 15	1	Bits per Character (Modbus RTU requires 8bits)
Word 16	0	Ignored
Word 17	1	Port Interface (VersaMax: 0=Port1, 1=Port2)
Word 18	0	Ignored
Word 19	0	Character-gap time-out in 100-microsecond units (0 to 6.5535 seconds)
Word 20	0	Ignored
Word 21	0	Ignored

3. ModBus protocol communication

Modbus master device is connected to the bus along with Slave devices. The network can interconnect up to 247 Slave devices. The maximum distance between the Master controller and any slave is around 1200 m. In Modbus protocol, two basic types of data exchange can be distinguished:

- inquiry/command - response being an exchange of information with a selected Slave controller,
- broadcast - makes transmission to every Slave subscribers, no Slave response.

PLCs use three serial data transmission standards: RS-232, RS-422 and RS-485. Depending on the protocol, controllers can be connected using point-to-point (1 Master controller and 1 Slave controller) or a multidrop (1 Master and several Slave controllers) networks. The point-to-point communication can be based on all the transmission standards. However, multi-drop connection can be obtained using RS-422 or

RS-485 only, due to many controller transmissions. The physical connection can be executed by means of 2- or 4-wire standard (GE, 2000).

Commands are sent, from the Master to the Slave controllers, in the form of a data frame with a specific beginning and ending. This approach allows the receiver to reject incomplete frames, thus signalling the possible errors. The ModBus interface enables two modes of data transmission (ASCII and RTU). Depending on the type of the device one can use ASCII or RTU frame (GE, 2000, 2003).

Sending control commands is done using COMMREQ command. Modbus protocol data transfer does not require sending the Attach command because the communication frame includes the Slave controller's address. In order to read or write data in the ModBus protocol, it is necessary to execute Read / Force / Preset: 8002 (GE 2003) command. The command sends a specific message to the slave controllers, that includes parameters contained in the block command/data block.

4. ModBus protocol modification of the communication

Controllers' configuration to operate in a network requires advanced knowledge in PLC programming field. Complex procedures of controller programming are the cause of frequent data transfer errors and performance issues. In order to accelerate and simplify the network data exchange amongst PLCs, the programmer can benefit from using function blocks. In a function block, some repeating variables can be permanently entered, and other variables can be parameterized.

After careful analysis of the transmitted frame structure during RTU Master/Force/Preset commands execution, We have determined that the following variables must be parameterized in this block:

- command,
- station address identifier,
- memory type and address in Slave controller,
- memory type and address in Master controller,
- a number of elements to be transmitted.

This modification is based on a parameterised subprogram execution, in which constant transmission parameters are written. Execution of communication is based on calling up the parameterised function block and then executing transmission by means of the COMMREQ function. Fig. 4 presents the block of the parameterised function. Fig. 5 depicts a data block that is written per the conventional method, and Fig. 6 presents the subprogram.

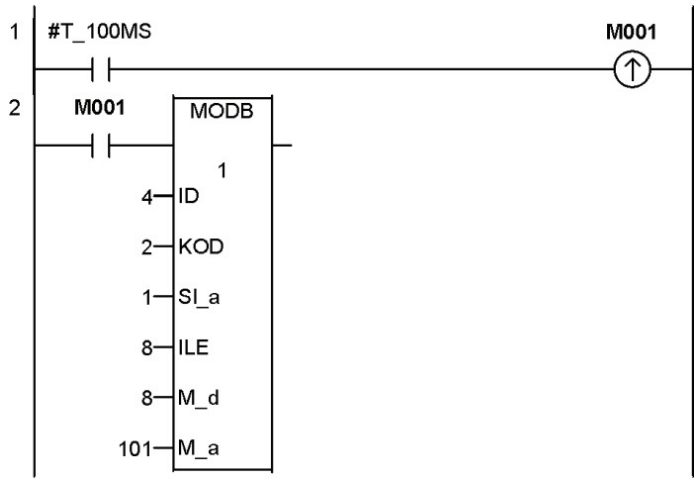

Fig. 4: Running of parameterised function

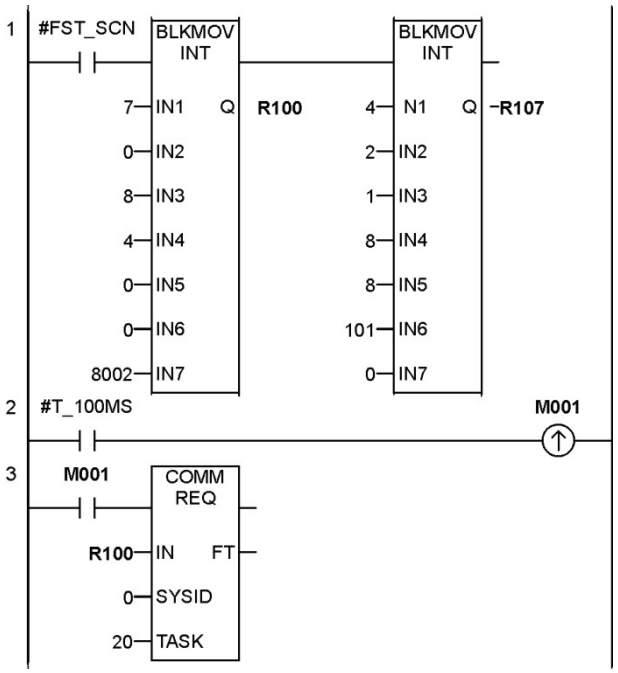

Fig. 5: Conventional programming of the data block

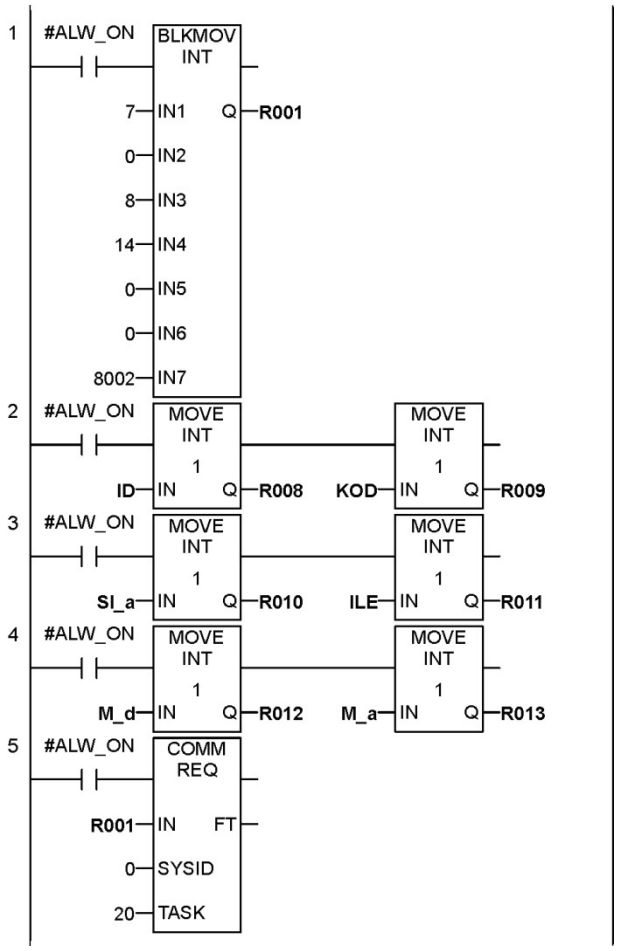

Fig. 6: Blocks contained in the subprogram

5. Conclusions

A sample program uses periodic read of 3 inputs each form: PLC 2, PLC 3, PLC 4 and PLC 5. Data is then sent to the PLC Master's memory using 1-12 outputs. Moreover, inputs 1 to 4 of Master's are connected to 1-4 PLC 2 outputs, 5-8 to PLC 3 outputs and following controllers accordingly.

To create a program that allows communication between one Master and four slaves PLCs requires using 16 data blocks and 8 COMMREQ functions, using conventional method. The task of communication

Execution of the same program using function blocks involves only 8-time invoke of MODB function block. Conventional method requires usage of 120 registers. However, using function blocks reduces memory usage to only 14 registers. The procedure placed in a function block is depicted in Fig. 6. Programming using this method is more friendly and transparent, it can also facilitate configuration and networked operation programming.

References

Dzierzek, K. (2007) GE Fanuc controller programming. Bialystok University of Technology Press, Bialystok (in Polish).

GE Intelligent Platforms (2015) PACSystems RX3i System Manual GFK-2314K.

GE Intelligent Platforms (2003) Modbus RTU Master Communications, GFK-2220B.

GE Intelligent Platforms (2012) Proficy Logic Developer – PLC, Getting Started, Version 7.5. GFK1918Q.

GE Intelligent Platforms (2014) Proficy Machine Edition, Getting Started, Version 7.5. GFK-1868Q.

GE Intelligent Platforms (2000) Series 90 PLC, Serial Communication User's Manual, GFK-0582D.

GE Intelligent Platforms (2009) VersaMax Micro PLCs and NanoPLCs, User's Manual GFK-1645J.