Study of different communication methodology for slave device communication

Aniruddh Mali
Student, Department of EC Engineering
L. D. College of Engineering
Ahmedabad, India
aniruddhkmali@gmail.com

Hiren Dhuda
Assistant Professor
ITM Universe
Vadodara, India
hiren.dhuda@gmail.com

***Abstract*— This paper presents different methodologies to communicate with slave devices like Programmable Logic Controllers – PLCs. We have studied modbus slave devices and its communication techniques in LabVIEW, a GUI programming software. With advancement of technology, modbus slave devices are well known and easy to adopt for industrial automation solutions. It has also been explained that how a prototype can can be setup using LabVIEW.**

***Keywords*— modbus, PLC, GUI, LabVIEW, slave device**

I. INTRODUCTION

Logic Controllers like PLCs are widely used in industrial automation which has different requirements as well various communication topologies. There is a Master – slave architecture used for large scale industries where redundancy is highly required. Modbus protocols are available with such slave devices which can be programmed in LabVIEW with easy steps. Indeed, there are many options available instead of LabVIEW but now a days, GUI based programming solutions are very popular with cost effectiveness so industry adopt this technical approach.

II. MODBUS SLAVE DEVICE COMMUNICATION

A. Modbus protocols

It is an open source protocols which is available for different communication interfaces/ methods i.e. ASCII, RTU, Ethernet, etc. PLCs as well different logic controllers, microcontrollers and open source development boards also available with modbus support.

B. Communication methods

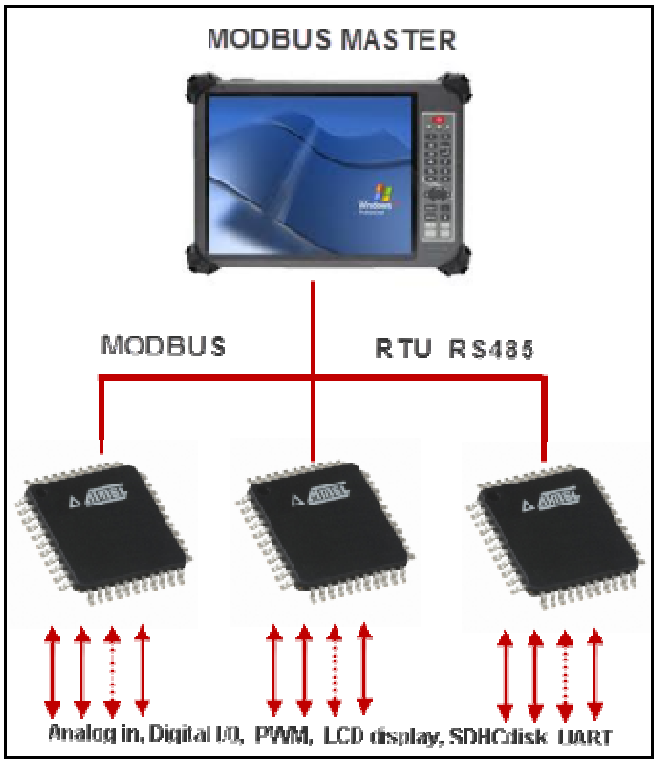

Fig.1 RTU Communication

Fig.1 shows master slave communication architecture using modbus protocols. There is a master controller device which is connected to slave devices using RTU interface which communicate with slave devices using modbus protocols on RTU environment. It is one kind of serial transmission mode in which allows the instrument to be a citizen on a data link shared with other devices that subscribe to the Modbus RTU RS-485 specification. It is respected in the physical and data link layers.

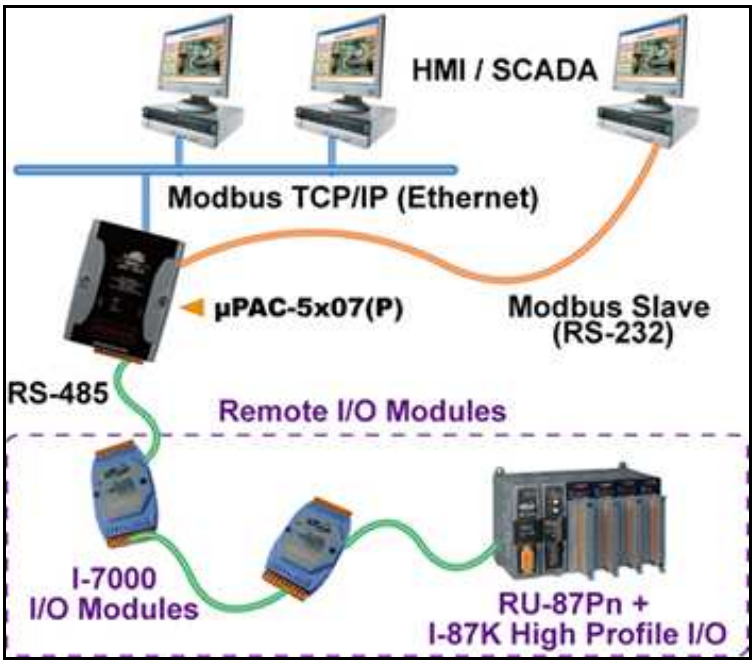

Fig.2 Ethernet Communication

Fig.2 shows modbus communication using Ethernet where different slave devices are connected to a Ethernet cable or network with necessary I/O modules. It is high speed communication methods because of its data speed as well redundancy capabilities. Now a day, slave devices are available with Ethernet port.

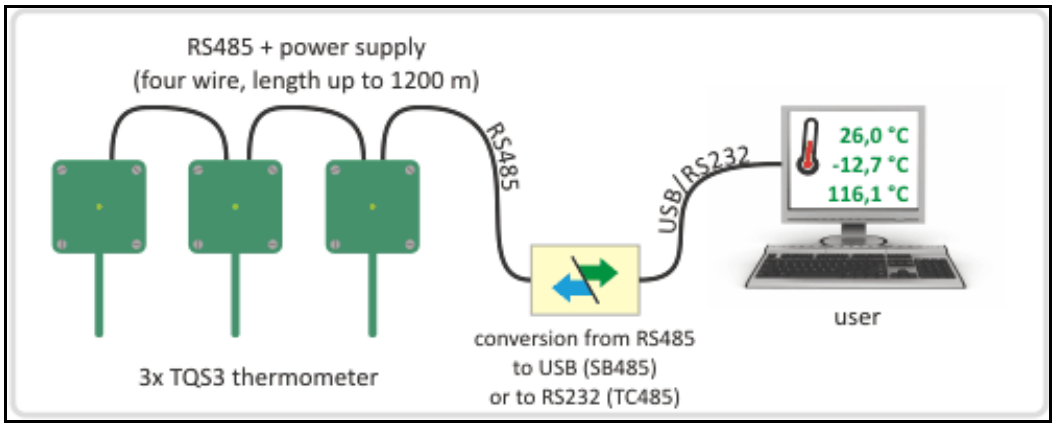

Fig.3 ASCII communication

Fig.3 shows RS 232 bases serial communication using modbus which use ASCII frames for communication instances. It is widely used in industry where robustness and ruggedness is preferred. It has options of conversion to RS 485 which is an added advantage. It used for small scale production unit where strings of data continuously comes but it has limitations of long distances.

III. MODBUS AND LABVIEW

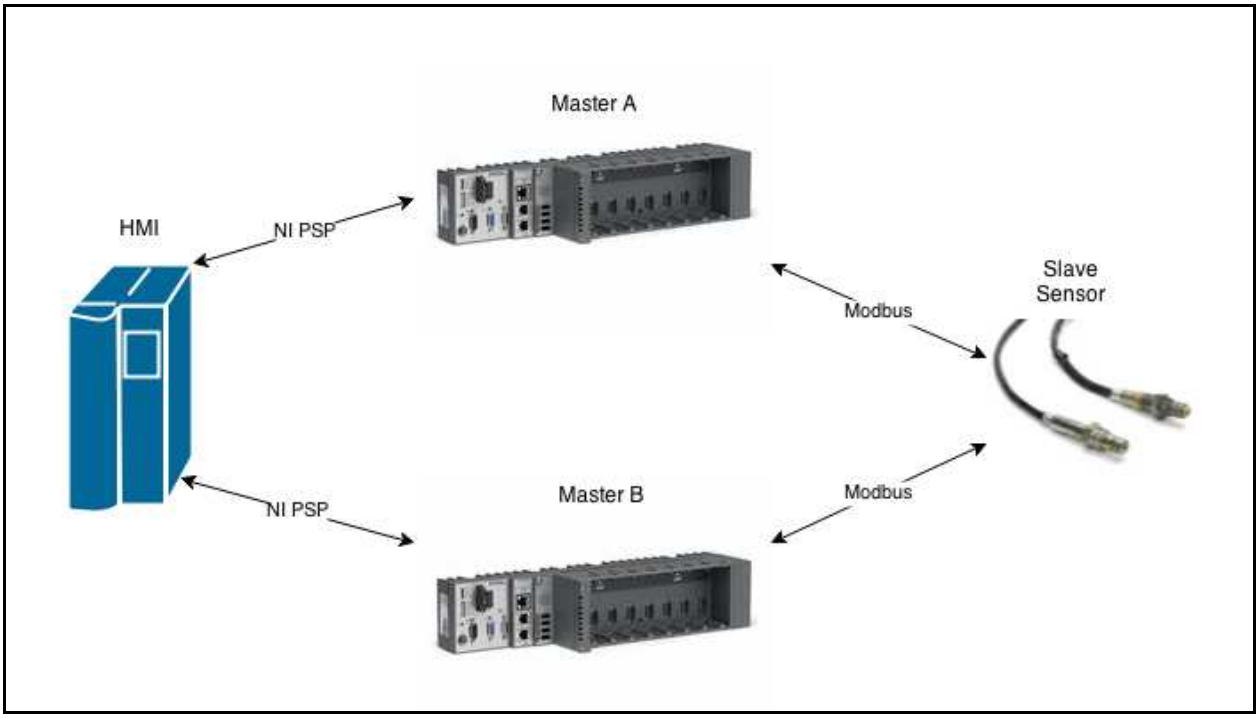

Fig.4 LabVIEW Modbus API (Courtesy: http://www.ni.com/white-paper/7675/en/#toc4)

The low-level Modbus API is the preferred option when your application needs a high level of control over the sequencing and timing of Modbus requests. The low-level API is typically also the preferred choice where flexibility is paramount. In contrast, the flexibility and power offered by the LabVIEW Modbus API also means that your application code must be more complex to correctly manage the API.

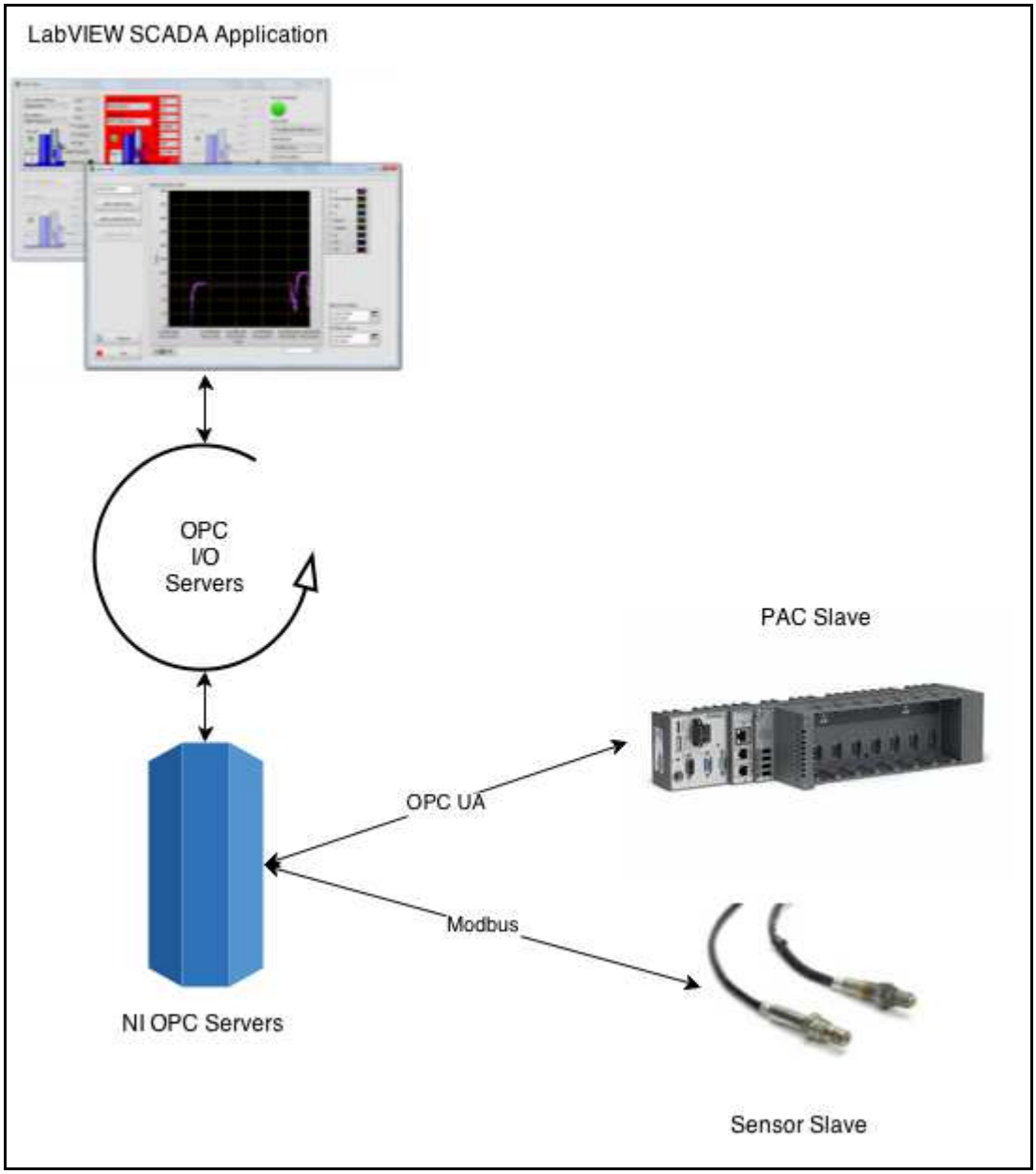

Fig. 5 OPC server communication in LabVIEW for modbus slave device

For complicated applications involving many slave devices that communicate over different protocols, the standard Modbus I/O might not suffice. A common solution is to use an OPC server, which acts as a data aggregator for all of your systems, and then use the OPC I/O servers included in the LabVIEW DSC Module to communicate with that OPC server.

Figure 5 shows an example of this architecture, with NI OPC Servers using Modbus to communicate directly with sensors and OPC UA to communicate with an NI CompactRIO PAC. After data is aggregated in NI OPC Servers, an OPC I/O server can retrieve data and share it with the LabVIEW application.

Conclusion

We have studies different communication methodologies using modbus protocols for slave devices in LabVIEW which are easy to adopt and cost effective solutions.

References

[1] http://www.ni.com
[2] https://www.honeywellprocess.com/library/support/Public/Documents/51-52-25-66.pdf
[3] http://www.intea.hr/downloads/introduction_to_modbus.pdf

