

Versatile Communication Solution for PLC Based Control Systems

AUREL GONTEAN, MARIUS OTEȘTEANU
 SANDRA RUGINĂ, GEORGIANA SÂRBU-DOAGĂ
 Department of Communications
 “Politehnica” University Timisoara
 Bd. Vasile Parvan, nr. 2
 ROMANIA

Abstract: - The main objective of this paper is to present a working solution for measuring data from a sensor array via a PLC and simultaneously controlling a set of actuators by the means of a Labview application. A demo box was developed in order to test the communication protocol and finally to simulate a closed loop control for a slow system.

Key-Words: - PLC, automation, virtual instrumentation, communication protocol, control, Labview.

1 Introduction

Any control system goes through four stages from conception to a working plant. A PLC system brings advantages at each of those stages: design, construction, installation, commissioning and to the additional fifth stage, maintenance. The application described below presents the features offered by LabView and PLC applications. The experimental setup contains: a PLC Siemens S7-200(CPU 214, Expansion Module 231), a PC and the demo box.

The controlled elements can be easily substituted with actuators. In this particular application several LEDS are used as indicators, a bulb is proportionally lit via the PWM output of the PLC; the state of three switches, 2 potentiometers and a photoresistor are read via the analog inputs of the PLC.

2 Developing the PLC Software

Step 7-Micro/Win32 is the software used with the S7-200 PLC to create operating programs. Step 7 contains a number of instructions that must be arranged in a logical order in order to obtain a desired sequence of PLC operations. The instructions are divided into three groups: standard, special, and high-speed.

Standard instructions are those that are found in most programs. Examples including timer, counter, math, logical, increment/decrement/invert, move, and block instructions.

Special instructions are used to manipulate data. They include shift, table, conversion, for / next and real time instructions.

High-Speed instructions allow for events and interrupts to occur independently of the PLC scan time. These include high-speed counters, interrupts, output, and transmit instructions.

2.1 Developing the Closed Loop Simulation – Information Flow

The protocol relies on telegrams periodically exchanged between the PLC and the PC Labview application; one interesting feature of the protocol is that a missing telegram (or a failure on the

Table 1 General characteristics of the PLC used

General Features	
Power dissipation: 8W at 3 A load	
Maximum I/O expansion modules: 7	
Digital I/O supported: 64 inputs / 64 outputs	
Analog I/O supported: 16 inputs / 16 output	
Power Supply	
Voltage range: 20 to 28.8 VDC	
Input current: 85 mA typical	
CPU 900 mA (maximum load)	
Input Points	
Input type (IEC 1131-2): Type 1 sinking	
On state nominal: 24VDC, 7 mA	
OFF state maximum: 5VDC, 1 mA	
Optical isolation: 500VAC, 1 min	
DC Sensor Supply	
Voltage range: 16.4 to 28.8VDC	
24 VDC available current: 280 mA	
Short circuit current limit: < 600 mA	
Isolated: No	
Output Points	
Output type: Sourcing Transistor	
Voltage range: 20.4 to 28.8 VDC	
Leakage current: 100uA	
Switching delay: 25us ON, 120us OFF	
Optical isolation: 500VAC, 1 min	

transmission) does not imply the end of the control mechanism – by default, the last valid values received are stored and used.

The operator sets a control value to the PC (fig. 1); the computer sends a telegram to the PLC. The PLC controls the actuators, acquires the physical data and sends back a telegram to the PC as response.

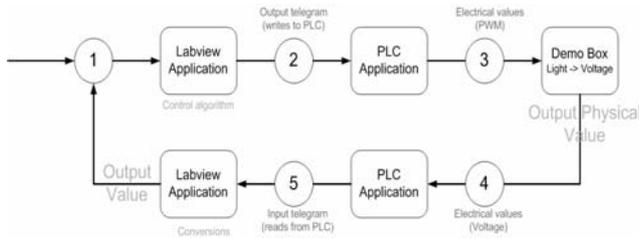


Fig. 1 The Closed Loop Simulation – Information Flow

2.2 Developing the Closed Loop Simulation – Hardware

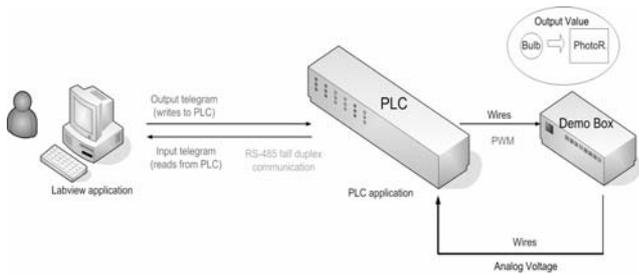


Fig.2 Hardware for Closed Loop Simulation

The physical link uses the RS485 port, 9600 bauds, no parity, 1 stop bit. An overview of the hardware is presented in figure 2.

3 The Actual PLC Software

The communication protocol is based on telegrams exchanged between PLC and the host computer. In order to get a high efficiency communication, templates of the telegrams are kept in PLC’s memory. The numerical values are changed only if needed, although telegrams are sent periodically (for the particular example studied, dealing with slow mechanical systems, the delay between 2 telegrams is 1 second).

The PLC software consists of the Main function, one subroutine (SBR0) and 3 interrupt service routines (INT0, INT1, INT2).

3.1. PLC Data Structure

```
//
// Simatic -> Labview Telegram Template
//
VB100 41 // 41 chars in the PLC Telegram
// VB106 - value(SW01) == 0/1
// VB113 - value(SW02) == 0/1
// VB120 - value(SW03) == 0/1
VB101 'SW01_0;SW02_0;SW03_0;' // Digital inputs

// VB126-VB128 - AI01 - 000...FFF
// VB135-VB137 - AI02 - 000...FFF
VB122 'AI01_000;AI02_000;' // Analog inputs

VW140 16#0D0A // CR & LF

//
// Labview -> Simatic Telegram Template
//
VB300 33
// VB306 - value(Q0.5) == 0/1
// VB313 - value(Q0.6) == 0/1
// VB320 - value(Q0.7) == 0/1
VB301 'Q0.5_0;Q0.6_0;Q0.7_0;'

// VB327-VB330 - PWM Q0.1 - 0000H...03E8H = 1000 dec
// VB337-VB340 - PWM Q0.0 - 0000H...03E8H = 1000 dec
VB322 'Q0.1_03E8;' // PWM Output

VW332 16#0D0A // CR & LF
VW337 16#0064 // PWM initial value
VW437 16#0064 // PWM Initial value
```

3.2. PLC software description

3.2.1 Main function

MAIN function (fig.3) resets Q0.1 input for PWM operation and turns ON the Q0.5, Q0.6, Q0.7 inputs where 3 LEDs are wired (if such a command has been given from Labview). The current PWM value is compared with the previous value; if there is a new value the PWM value is updated. The PWM value can be changed from Labview.

3.2.2 SBR0

SBR_0 (fig.4) configures the serial port: baud rate, number of bits, message length and attaches the interrupts, calls INT_0 (Receive completed), INT_1 (Transmit completed), and INT_2 (inspects the analog and digital inputs).

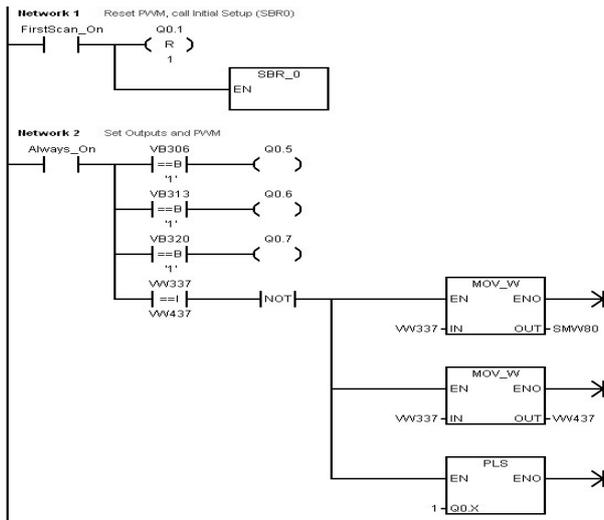


Fig.3 Main function

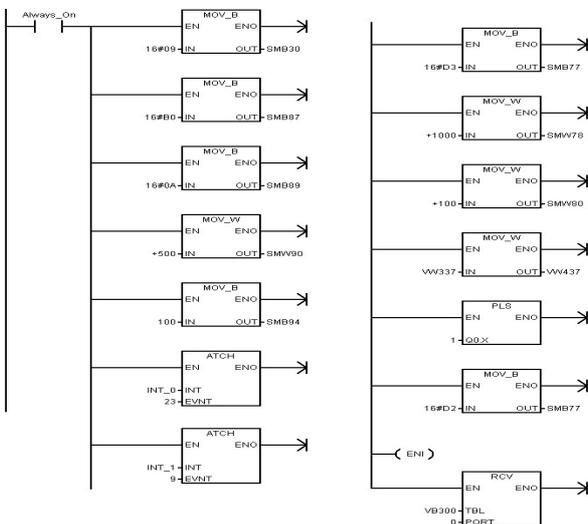


Fig.4 SBR0 Ladder view

3.2.3 Interrupt 0

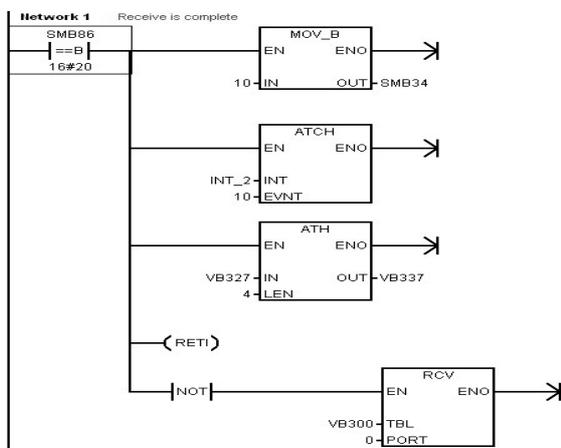


Fig.5 Interrupt 0

INT_0 (fig. 5) checks the length of the message and if an end of message char is detected the message has been received, if not - keeps on receiving.

3.2.4 Interrupt 1

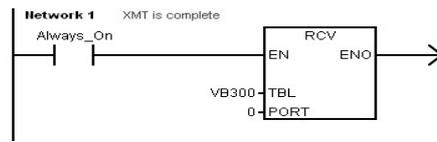


Fig. 6 Interrupt 1

INT_1 (fig.6) simply transmits the message from memory location VB300.

3.2.5 Interrupt 2

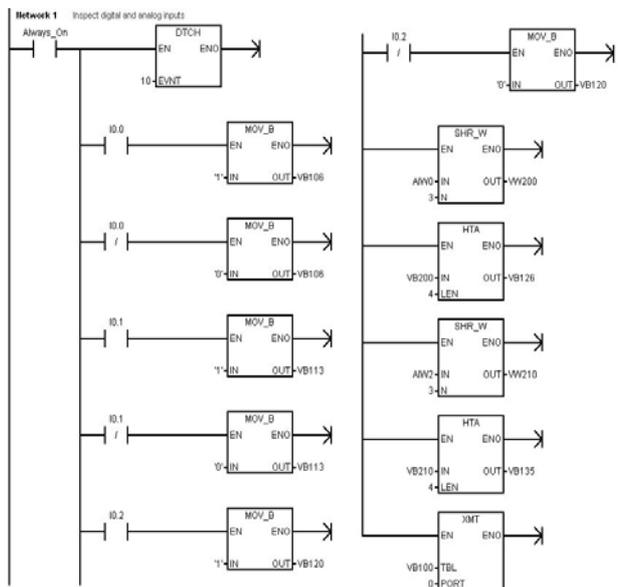


Fig. 7 Interrupt 2

Interrupt 2 (fig. 7) checks the state of digital inputs (I0.1, I0.2, I0.3) where the 3 switches are wired, the value of the analog inputs (AIW0, AIW2); the result is shifted 3 bits right and converted in ASCII format.

3.3 Connecting the PLC via Labview Application

The S7-200 CPU communication ports use RS-485 to allow multiple devices to be attached to the same network. The PC/PPI cable (fig. 8) allows one to connect the RS-232 port of a personal computer to the RS-485 port of an S7-200 CPU. The PC/PPI cable can also be used to connect the communication port of an S7-200 CPU to other RS-232 compatible devices.

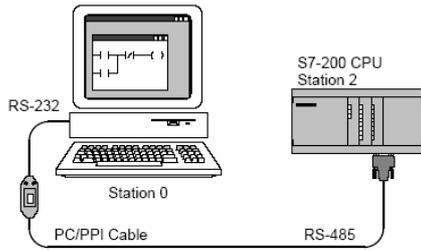


Fig 8. Using a PC/PPI Cable for Communicating with an S7-200 CPU

3.4 The Demo Box

The purpose of the demo box (fig.9) was to test the communication protocol in real life and second to emulate a slow system behavior (a mechanical system). A bulb operated with different voltages proved a good approach – in closed loop it showed a time constant response of about 100 ms.



Fig. 9 Demo Box

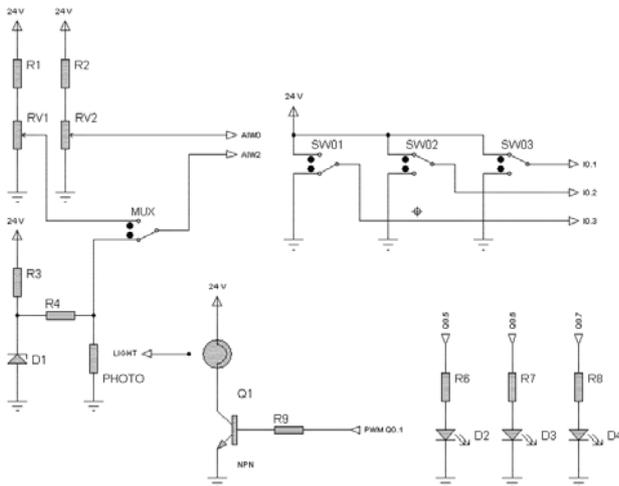


Fig. 10 Demo Box schematics

3.5 LabView application

For PC (Labview) – PLC communication a dedicated protocol had to be established. The PLC and PC must be synchronized; the protocol must recover if the link is temporally down.

The demo box contains (fig. 10):

- 3 LEDs wired to the Q0.5, Q0.6 and Q0.7 outputs of the PLC;
- 3 switches wired to the PLC inputs I0.1; I0.2; I0.3;
- 2 Potentiometers delivering a 0 ... 5 V analog voltage, wired to the analog inputs of the PLC EM231 analog expansion module (AIW0 and AIW2);
- A bulb controlled by the PWM output of the PLC (Q0.1) via a npn transistor.
- A photoresistor converts the bulb light into voltage (in the 0 ... 5V range).
- A switch selects which demo box output is used as an analog input for the PLC: a Potentiometer or the Photoresistor.

Using the LabView software a virtual instrument was developed. First, the communication was tested in an open loop.

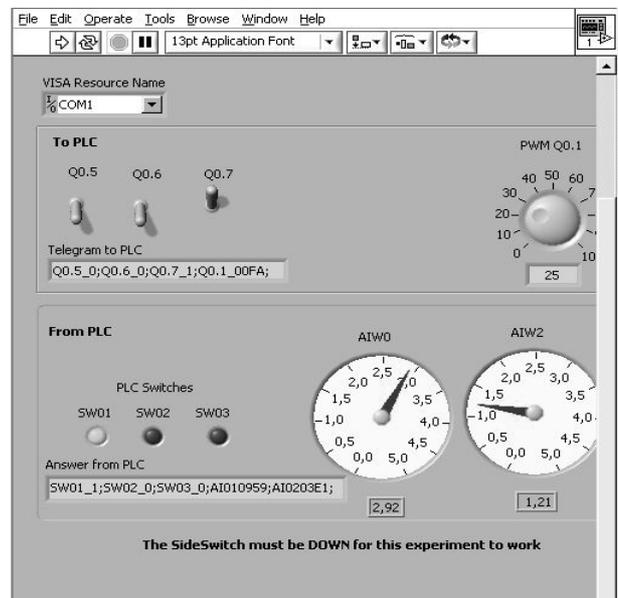


Fig. 11 VI for testing the PC – PLC protocol

There are two sections in the VI presented in figure 11. One section is dedicated to the PLC output. Virtual switches Q0.5, Q0.6, Q0.7 correspond to the similar outputs of the PLC; the 3 LEDs correspond to LEDs on the demo box. The PWM knob controls the PWM output; controlling the bulb wired on the demo box. It is also possible to see the telegram sent to PLC.

The second section is dedicated to PLC inputs. In this section the 3 switches on the demo box are read and the corresponding virtual LEDs are turned ON. The values on the PLC analog inputs (AIW0, AIW2) are read and displayed (the voltage range is 0 ... 5V). The voltage is derived by a potentiometer and a

potentiometer or a photo resistor. It is also possible to see the telegram received from PLC.

4 Results

The closed loop control virtual instrument VI presented in figure 12 demonstrates the behavior of

the control algorithm. The Preset value is set by the user and the system reaches this value in several steps varying the PWM value; when in the correct range, a LED is on (“tracked”).

During the control process, asynchronous commands may be performed on both sides (PC and PLC).

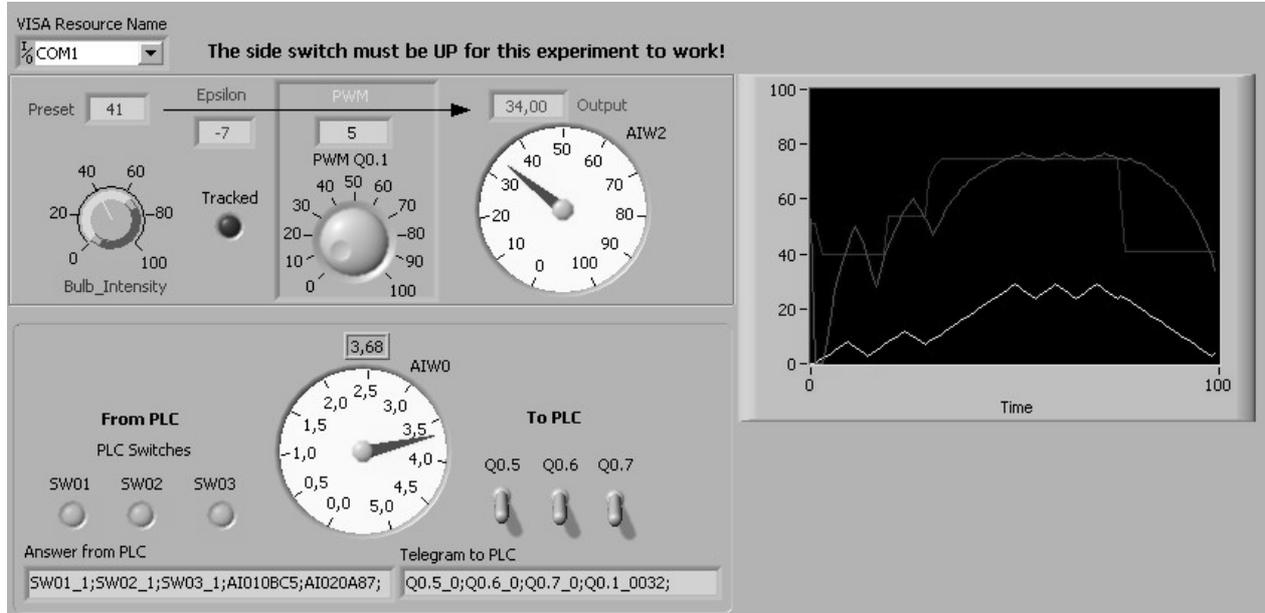


Fig. 12 Closed Loop application

5 Conclusion

This paper presented a working solution for measuring data from a sensor array via a PLC and simultaneously controlling a set of actuators by the means of a Labview application. A demo box was used in order to simulate a closed loop control for a slow system.

Acknowledgements:

This work was supported by Siemens V.D.O. and the authors are especially grateful to Mr. Andreas Henning Head of RS Department..

References:

[1]. E.A. Parr, *Programmable Controllers. An engineer’s guide*, Newnes ISBN 0 7506 5757 X, 2003
 [2] * * * SIEMENS SIMATIC S7-200 Programmable Controller System Manual, Edition 08/2005

[3] * * * SIEMENS STEP 7-Micro/WIN32 Documentation
 [4] Greg, J. Hammer, Input/Output Control of Asynchronous Sequential Machine, *IEEE Trans. on Automatic Control*, Vol.50, No.12, 2005, pp. 1956 - 1970
 [5] M. Fu, L.Xie, The Sector Bound Approach to Quantized Feedback Control, *IEEE Trans. on Automatic Control*, Vol.50, No.11, 2005, pp 1698 - 1711
 [6] G.C. Calafiore, M.C. Campi, The Scenario Approach to Robust Control Design, *IEEE Trans. on Automatic Control*, Vol.51, No.5, 2006, pp 742 - 753