METHODOLOGY FOR TEACHING BASIC SKILLS IN PLC PROGRAMMING DEVELOPMENT

Atanas Nikolov Iovev, Peter Ivanov Yakimov, Nikolay Todorov Tuliev, Stefan Jordanov Ovcharov, Emiliya Georgieva Balkanska, Eftim Ivanov Stoyanov[*]

Department of Electronics, Technical University of Sofia, 8 Kliment Ohridski Str., 1000 Sofia, Bulgaria, phone: +359 2 965 3265, e-mail: iovev@tu-sofia.bg
[*] Department of Computer Information Technologies, University "Prof. Dr. Asen Zlatarov", "Prof. Iakimov" str., 1, 8000 Bourgas, Bulgaria, eftim55@abv.bg

The teaching and training of the students in the field of Programmable Logic Controllers (PLC) is very important, because it determines the safety and the quality of control. This paper describes the various parts of the PLC and demonstrates some software for performing tasks using some of the next typical instructions: Bit Logic Instructions, Move and Conversion Instructions, Timer Instructions and Counter Instructions. The experiments give students knowledge and practice with the basic PLC functions.

Keywords: PLC, Timer, Counter, lab exercises

1. INTRODUCTION

The main advantage that Programmable Logic Controllers (PLCs) provide is flexibility. Behaviour of the system can be easily changed via program without any other alterations. Special devices for example make any changes in control algorithm very hard to implement. Flexibility makes PLCs well suitable for frequently changed applications. In PLCs the relations between inputs and outputs are determined by user program. By using advanced programming technologies it is much easier to implement complex control algorithms than in any hard-wired solutions. It makes PLCs very competitive for complex tasks [3].

Lab experience is an important factor in teaching PLCs and their subsystems. PLCs span a vast range but the fundamental ideas behind all of these units are about the same. Thus, by learning one type PLC, the students will be able to move to other units with relative ease. The tool used in these experiments is PLC Simatic S7-300.

The aims of the educational strategy are:

1) To offer basic knowledge of PLC's digital inputs and outputs
2) To develop and demonstrate software using some typical instructions.

The first aim is connected with PLC's digital inputs and outputs, which are necessary to monitor and control a process. Digital inputs come from sensors that translate physical phenomena into electrical digital signals. Digital outputs to actuators allow a PLC to cause something to happen in a process. For the tool used in the experiments, levels of the digital signal applied to the digital inputs are 0V and 24V, and the digital outputs are DC outputs with transistors.

The second aim is connected in the beginning with teaching of the students in the possibilities of the three Editors, used in PLC programming.

The first Editor is Ladder Logic Editor (LAD), using to build programs that resemble the equivalent of an electrical wiring diagram. Basically, the ladder programs allow the CPU to emulate the flow of electric current from a power source, through a series of logical input conditions that in turn enable logical output conditions. The logic is usually separated into small, easy-to-understand pieces that are often called "rungs" or "networks." The program is executed one network at a time, from left to right and then top to bottom as dictated by the program. Once the CPU has reached the end of the program, it starts over again at the top of the program.

The second Editor is Statement List Editor (STL), which allows creating control programs by entering the instruction mnemonics. The CPU executes each instruction in the order dictated by the program, from top to bottom, and then restarts at the top.

The third Editor is Function Block Diagram Editor (FBD), which allows viewing the instructions as logic boxes that resemble common logic gate diagrams. There are no contacts and coils as found in the LAD editor, but there are equivalent instructions that appear as box instructions. The program logic is derived from the connections between these box instructions.

The main part of the second aim is connected with teaching some instructions most typically applied in PLC programming. Actually they define the structure of the lab exercises. Beneath are examined these instructions:

- **bit logic instructions (BLI)** – they interpret signal states of 1 and 0 and combine them according to Boolean logic. These combinations produce a result of 1 or 0 that is called the "result of logic operation" (RLO). The logic operations that are triggered by the bit logic instructions perform a variety of functions. Basic BLI, which are including in the lab exercises perform the next Binary Logic Operations: AND, OR, XOR, (=), SET and RESET;

- **load and transfer instructions** – they enable to program an interchange of information between input or output modules and memory areas, or between memory areas;

- **conversion instructions –** they convert binary coded decimal numbers and integers to other types of numbers;

- **timer instructions** – a timer is a function element of the programming language that implements and monitors timed sequences. The timer instructions enable the program to perform the following functions:

1) provide waiting times;
2) provide monitoring times;
3) generate a pulse or sequence of pulses with programmable parameters;
4) measure time.

- **counter instructions** – they are the only functions that have access to the counter memory area reserved for the counters in the memory of the CPU.

2. DEVELOPING LAB EXERCISES WITH PLC TRAINING SYSTEM

2.1. PLC training system

The PLC Training System for teaching and training of the students is composed of basic S7-300 PLC system and simulator for programming testing [1].

The basic S7-300 PLC system includes a CPU module, a power supply module, a personal computer, STEP 7 programming software, and a communications cable. The CPU module used in the experiments is type 313C. This module has 24 digital inputs and 16 digital outputs.

A simulator for programming testing provides digital outputs and digital inputs. The digital outputs (push button switches, digital BCD switches and a digital – controlled oscillator) are connected to the digital inputs of the PLC. The digital outputs of the PLC control LED indicators and LCD display, which are included in the simulator.

The programs in the exercises will be developed and demonstrated with the STL programming package, which is an integral part of the STEP 7 Standard Software.

2.2. Experiments for PLC training system

1) Controlling LED indicators using bit logic instructions

The purpose of this experiment is to develop and train programs including normal opened contacts and normal closed contacts.

Six push button switches of the simulator are connected to six digital inputs of the PLC. Three digital outputs of the PLC control three LED indicators of the simulator using the next algorithm:

- LED 1 is "on" when the push button switches S1 and S2 are both switched on, in another case LED 1 is "off";
- LED 2 is set "on" when the push button switches S1 and S2 are both switched on, and LED 2 is set "off" when the push button switch S3 is pressed or the normally closed push button switch S4 is opened;
- LED 3 is "on" when only one of the push button switches S5 and S6 is pressed – function XOR.

Table 1 lists the components of the system and their corresponding absolute addresses.

System Component	Absolute Address
Push Button Switch S1	I 124.0
Push Button Switch S2	I 124.1
Push Button Switch S3	I 124.2
Push Button Switch S4	I 124.3
Push Button Switch S5	I 124.4
Push Button Switch S6	I 124.5
LED 1	Q124.0
LED 2	Q124.1
LED 3	Q124.2

Table 1

The statement list program is:

```
STL                                   Explanation
Network 1
A     I  124.0
A     I  124.1
=     Q  124.0             // pressing switches S1 and S2 light LED1 and LED2 on
S     Q  124.1
Network 2
O     I  124.2
ON    I  124.3             // pressing switch S3 or opening the normally closed switch S4
R     Q  124.1             // light LED2 off
Network 3
X     I  124.4
X     I  124.5
=     Q  124.2             // pressing one of the switches S5 and S6 light LED3 on
```

2) Displaying a BCD number after simple math instruction.

This experiment demonstrates loading of one BCD number, its conversing in integer and after one simple math instruction the result in BCD format is displaying on LCD display.

Four digital BCD switches of the simulator are connected to 16 digital inputs of the PLC and 13 digital outputs of the PLC control LCD display of the simulator. A BCD number set by using the four digital BCD switches in the range (+/-499) is multiplying by 2 and the result is shown on LCD display. Table 2 lists the components and their corresponding absolute addresses used in this experiment. An explanation of the program follows the table 2.

System Component	Absolute Address
digital BCD switch for ones	I 125.0 to I 125.3
digital BCD switch for tens	I 125.4 to I 125.7
digital BCD switch for hundreds	I 124.0 to I 124.3
digital BCD switch for a sign	I 124.4 to I 124.7
LCD display for ones	Q 125.0 to Q 125.3
LCD display for tens	Q 125.4 to Q 125.7
LCD display for hundreds	Q 124.0 to Q 124.3
LCD display for a sign	Q 124.4

Table 2

```
STL                                   Explanation

Network 1
L     IW   124                 // load accumulator 1 low  with a BCD number, set by
                               //digital BCD switches
BTI                            // convert BCD to Integer
L     2
*I                             // multiply by 2
ITB                            // convert Integer to BCD
T     QW   124                 // transfer the result to the LCD display
```

3) Pulse generator

This experiment illustrates the use of timer functions to realize a pulse generator with pulse duty factor 1:1. The width of one pulse is setting in the range (1-9) s.

One digital BCD switch of the simulator is used to provide the width of the pulse in seconds. A digital output of the PLC controls one LED indicator of the simulator and realizes a pulse generator.

Pulse generator Components and Corresponding Absolute Addresses are shown in Table 3.

System Component	Absolute Address
digital BCD switch for seconds	I 124.0 to I 124.3
LED indicator	Q124.0

Table 3

The following STL program uses ON Delay (SD) timer and realizes a pulse generator.

```
STL                                  Explanation
Network 1
L     IB    124                      // load accumulator 1 low  with the width of one pulse in
                                     // seconds – time value in BCD of timer T1
L     2#1111
AW                                   // null the other bits of accumulator 1 low
L     2#10000000000000
OW                                   // load accumulator 1 low  with a time base 1s of timer T1
T     MW   100                       // transfer the result to memory byte MB100
Network 2
AN    M     0.0
L     MW  100                        // load accumulator 1 low  with a time base and time value
SD    T     1                        // start timer T 1 as an on-delay timer
Network 3
A     T    1                         // check the signal state of timer T 1
=     M    0.0                       // the state of the timer is saved temporarily in an auxiliary
                                     // memory marker M0.0
Network 4
X     M    0.0
X     M    0.1                       // the result of logical function XOR is saved temporarily in
=     M    0.1                       // an auxiliary marker M0.1
Network 5
A     M    0.1
=     Q    124.0                     // control LED indicator
```

By learning and training one of the five types of timers, the students will be able to move to other types with relative ease.

4) Counter

This program uses Counter C1 to count the number of pulses in one pulse package and shows the result in BCD format on LCD display. A push button switch is providing to reset the counter.

Two digital BCD switches from the simulator set the number of pulses in one pulse package in the range (1 – 99). Pressing Start button, the digital–controlled oscillator generates a pulse package with frequency 100 Hz. The output of the oscillator and one push button switch of the simulator are connected to two inputs of the PLC. Eight digital outputs of the PLC control LCD display of the simulator.

Table 4 lists the components and their corresponding absolute addresses used in this experiment. An explanation of the program follows the table.

System Component	Absolute Address
Counter C1	I 125.0
Push Button Switch Reset	I 124.0
LCD display for ones	Q 125.0 to Q 125.3
LCD display for tens	Q 125.4 to Q 125.7

Table 4

STL			Explanation
A	I	124.0	// pressing push button switch Reset
R	C	1	// reset counter C1
A	I	125.0	// each pulse of the pulse package
CU	C	1	// increases the count value of counter C 1 by one
LC	C	1	// load accumulator 1 low directly with the count value of // counter C 1 in binary coded decimal format.
T	QW	124	// transfer the number of the pulses to the LCD display

3. Results

Here are reported several experiments to provide practice with PLC Simatic S7-300, CPU 313C. The programs are tested using the simulator for programming testing. The experiments with PLC Simatic S7-300 are good teaching tools and they offer knowledge that cannot be learned at classical lectures. The students have the opportunity to gain experience in basic concepts of PLC programming. The experiments could be upgraded with adding some components and the area of applications could be extended.

4. Acknowledgment

The investigations and the paper preparation have been supported by Technical University of Sofia within the framework of the project №08010 НИ-3/2008.

5. References

[1] Yakimov P. I., N. T. Tuliev, A. N. Iovev, S. J. Ovcharov, E. G. Balkanska, G. S. Mihov, *Investigation and Development of Electronic Instruments for PLC Programming Training*, in this issue

[2] Jack H., *Automating Manufacturing Systems with PLC-s*, 828 p.

[3] Mikkor A., L. Roosimölder, *Programmable Logic Controllers in Process Automation*, 4th International DAAAM Conference "Industrial Engineering – Innovation as Competitive Edge for SME", 29 - 30th April 2004, Tallinn, Estonia, pp 55-57.

[4] Berger H., Automating with SIMATIC, Siemens AG, 2003, 214 p.

[5] SIMATIC Programming with STEP 7 V 5.2: Manual, Siemens AG, 2002, 610 p.

[6] SIMATIC HMI WinCC Configuration Manual, Volume 1, 2, 3, Siemens AG 2000, 468 p.

[7] http://www2.sea.siemens.com/Training/Simulator+Systems/simsys_automation.htm