

Student Workplace for PLC Programming Training

Atanas Nikolov Iovev, Peter Ivanov Yakimov, Nikolay Todorov Tuliev, Stefan Jordanov Ovcharov and Emiliya Georgieva Balkanska

Abstract – Programmable logic controllers nowadays are the most common choice used widely in automation problems solution. Thus the teaching and training of the students in the field of Programmable Logic Controllers is very important. This paper describes the student workplace for PLC programming training and demonstrates some software for performing tasks using some typical instructions. The experiments give students knowledge and practice with the basic PLC functions.

Keywords – PLC, student workplace, lab exercises

I. INTRODUCTION

Programmable logic controllers (PLC) are used widely in automation problems solution. Their application allows a definite machine or process functioning to be started, controlled and monitored. 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 [5].

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

A. Iovev is with the Department of Electronics and Electronics Technologies, Faculty of Electronic Engineering and Technologies, Technical University - Sofia, 8 Kliment Ohridski blvd., 1000 Sofia, Bulgaria, e-mail: iovev@tu-sofia.bg

P. Yakimov is with the Department of Electronics and Electronics Technologies, Faculty of Electronic Engineering and Technologies, Technical University - Sofia, 8 Kliment Ohridski blvd., 1000 Sofia, Bulgaria, e-mail: pjj@tu-sofia.bg

N. Tuliev is with the Department of Electronics and Electronics Technologies, Faculty of Electronic Engineering and Technologies, Technical University - Sofia, 8 Kliment Ohridski blvd., 1000 Sofia, Bulgaria, e-mail: ntt@tu-sofia.bg

S. Ovcharov is with the Department of Electronics and Electronics Technologies, Faculty of Electronic Engineering and Technologies, Technical University - Sofia, 8 Kliment Ohridski blvd., 1000 Sofia, Bulgaria, e-mail: sjo@tu-sofia.bg

E. Balkanska is with the Department of Electronics and Electronics Technologies, Faculty of Electronic Engineering and Technologies, Technical University - Sofia, 8 Kliment Ohridski blvd., 1000 Sofia, Bulgaria, e-mail: egb@tu-sofia.bg

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 S7-200, CPU 226.

The aims of this work are:

- 1) To design a student workplace for PLC programming training.
- 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. The student workplace for PLC programming training is ideal tool for this purpose. Designed to imitate real-world environments it can provide one platform for job training and project testing.

The second aim is connected with teaching and testing some instructions most typically applied in PLC programming. Thus the students have the opportunity to gain experience in bit logic instructions (AND, OR, (=), SET and RESET), bit memories and timers. Actually it defines the structure of the lab exercises.

In PLC technology, bit memories are used as outputs and have an effect comparable with auxiliary contactors. A bit memory can be used as often as required at any location as an NC contact or an NO contact. Bit memories are used if the (interim) result of a network is to be further processed in other networks (like subtotals when adding numbers manually). They are also used to store evaluated follow-on states temporarily. The contents of bit memories is immediately available (in the same cycle) for follow-on logic operations.

The PLCs Simatic S7-200, CPU 226 include three timer instructions.

The **On-Delay Timer** and **Retentive On-Delay Timer** instructions count time when the enabling input is ON. When the current value (Txxx) is greater than or equal to the preset time (PT), the timer bit is ON. The On-Delay timer current value is cleared when the enabling input is OFF, while the current value of the Retentive On-Delay Timer is maintained when the input is OFF. You can use the Retentive On-Delay Timer to accumulate time for multiple periods of the input ON. A Reset instruction (R) is used to clear the current value of the Retentive On-Delay Timer. Both the On-Delay Timer and the Retentive On-Delay Timer continue counting after the Preset is reached, and they stop counting at the maximum value of 32767.

The **Off-Delay Timer** is used to delay turning an output OFF for a fixed period of time after the input turns OFF. When the enabling input turns ON, the timer bit turns ON immediately, and the current value is set to 0. When the input turns OFF, the timer counts until the elapsed time

reaches the preset time. When the preset is reached, the timer bit turns OFF and the current value stops counting. If the input is OFF for a time shorter than the preset value, the timer bit remains ON. The TOF instruction must see an ON to OFF transition to begin counting.

TON, TONR, and TOF timers are available in three resolutions – 1ms, 10ms and 100ms.

II. DEVELOPING LAB EXERCISES WITH STUDENT WORKPLACE FOR PLC PROGRAMMING TRAINING

A. The Student Workplace Design

The student workplace is composed of basic S7-200 PLC system, five push buttons, module with sixteen LED indicators, module with relays and one induction motor.

The basic S7-200 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 226 DC/DC/DC. This module has 24 digital inputs and 16 digital outputs. 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 module with sixteen LED indicators imitates one cross-road with traffic lights. The LED indicators are divided in ten groups. Ten digital outputs of the PLC control LED indicators.

Two digital outputs control two relays. Thus, with an appropriate program can be controlled the induction motor.

The five push buttons are connected to the digital inputs of the PLC.

The student workplace for PLC programming training is shown in the figure 1.

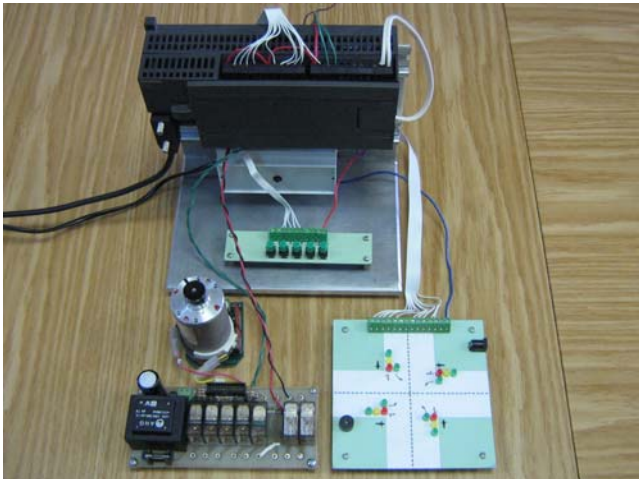


Fig. 1. Student Workplace

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

B. Controlling LED indicators using bit memories and timers.

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

Two push buttons are connected to two digital inputs (I0.0 and I0.1) of the PLC. Six digital outputs (Q0.0, Q0.1, Q0.2, Q0.3, Q0.4, and Q0.5) of the PLC control six LED indicators of the module with sixteen LED indicators using the next algorithm:

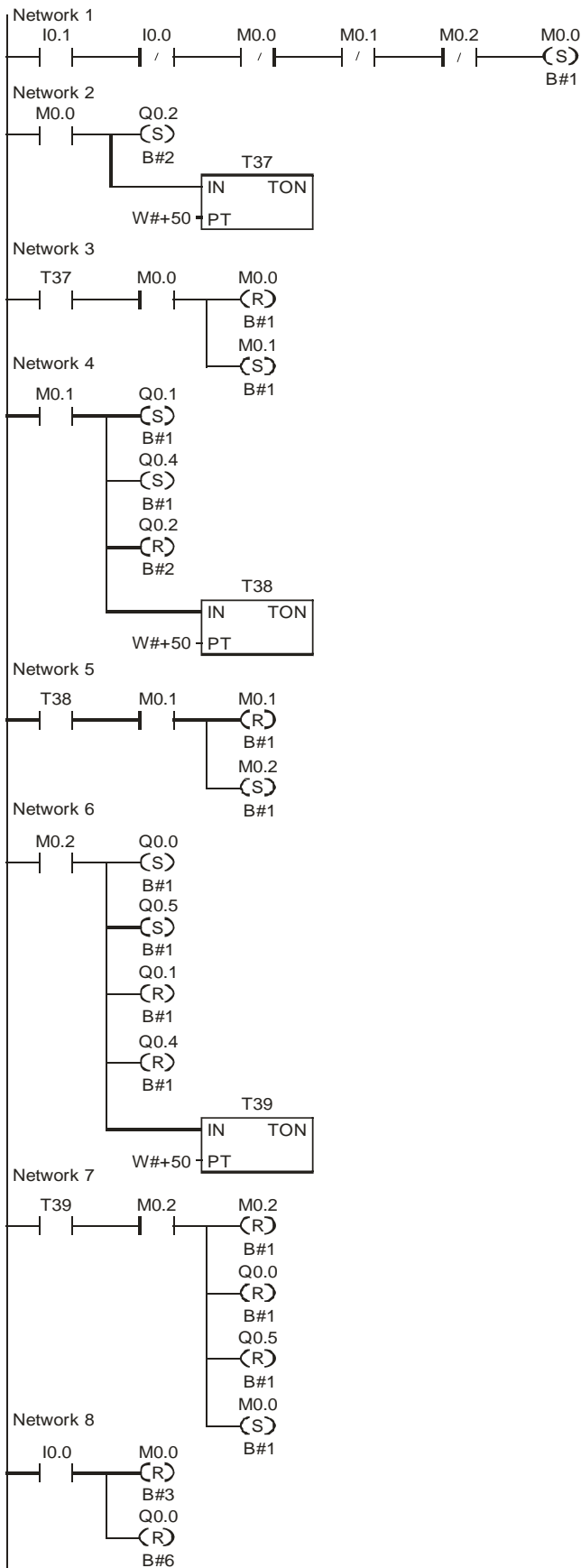
- if the push button B1, connected to input I0.1, is pressed, the program switch three pairs of LED indicators on one after one for 5 sec.

- if the push button B2, connected to input I0.0, is pressed, the program switch the six LED indicators off.

The statement list program is:

STL	Explanation
NETWORK 1	
LD I0.1	// push button "on"
AN I0.0	// push button "reset"
AN M0.0	// marker "step 1"
AN M0.1	// marker "step 2"
AN M0.2	// marker "step 3"
S M0.0, B#1	// marker "step 1" is 1
NETWORK 2	
LD M0.0	// step 1
S Q0.2, B#2	// LED 2 and LED 3 are on
TON T37, W#+50	// timer 5 sec.
NETWORK 3	
LD T37	// after 5 sec.
A M0.0	// marker "step 1"
R M0.0, B#1	// marker "step 1" is 0
S M0.1, B#1	// marker "step 2" is 1
NETWORK 4	
LD M0.1	// step 2
S Q0.1, B#1	// LED 1 and LED 4 are on
S Q0.4, B#1	
R Q0.2, B#2	// LED 2 and LED 3 are off
TON T38, W#+50	// timer 5 sec.
NETWORK 5	
LD T38	// after 5 sec.
A M0.1	// marker "step 2"
R M0.1, B#1	// marker "step 2" is 0
S M0.2, B#1	// marker "step 3" is 1
NETWORK 6	
LD M0.2	// step 3
S Q0.0, B#1	// LED 0 and LED 5 are on
S Q0.5, B#1	
R Q0.1, B#1	// LED 1 and LED 4 are off
R Q0.4, B#1	
TON T39, W#+50	// timer 5 sec.
NETWORK 7	
LD T39	// after 5 sec.
A M0.2	// marker "step 3"
R M0.2, B#1	// marker "step 3" is 0
R Q0.0, B#1	// LED 0 and LED 5 are off
R Q0.5, B#1	
S M0.0, B#1	// marker "step 1" is set
NETWORK 8	
LD I0.0	// push button "reset" is on
R M0.0, B#3	// all markers are 0
R Q0.0, B#6	// all LEDs are off

The Ladder diagram is:



This experiment could be upgraded with adding some components to imitate the control of one cross-road with traffic lights.

C. Controlling an induction motor

This programming example shows how to control an induction motor with two possible rotational directions. The push button "on" at input I0.2 allows the rotation. The push button "left" at input I0.0 is used to start the motor on the left. The push button "right" at input I0.1 is used to start the motor on the right. The rotational direction cannot be changed until a waiting period of 5 s is not elapsed. If the push button "right" and the push button "left" are actuated simultaneously, the motor stops and will not start. The outputs Q0.0 and Q0.3 control respectively the relay "right" and the relay "left". In addition, four LED indicators display the operating state. The LED indicator for rotation on the left is connected to output Q0.4, the LED indicator for rotation on the right is connected to output Q0.1, the LED indicator for the off state (no rotation) is connected to output Q1.0 and during the waiting period of 5 s, a LED indicator flashes at output Q1.1.

The statement list program and the Ladder diagram are:

```

NETWORK 1
LD    I0.1           // push button "right"
A     I0.0           // push button "left"
O     M2.3           // or marker "delay 5s"
S     M2.0, B#1     // marker "stop" is set
NETWORK 2
LDN   I0.0           // push button "left"
AN   I0.1           // push button "right"
AN   M2.3           // delay 5 sec. is passed
R     M0.0, B#20    // all markers are 0
NETWORK 3
LD    I0.2           // push button "on"
AN   M1.1           // marker "rotation right is on"
=     M2.1           // marker "rotation left"
NETWORK 4
LD    I0.2           // push button "on"
AN   M1.0           // marker "rotation left is on"
=     M2.2           // marker "rotation right"
NETWORK 5
LD    I0.0           // push button "left"
O     M1.0           // marker "rotation left is on"
AN   M2.0           // marker "stop" is not set
A     M2.1           // marker "rotation left"
=     M1.0           // marker "rotation left is on"
=     Q0.3           // relay "left" is on
=     Q0.4           // LED "left" is on
NETWORK 6
LD    I0.1           // push button "right"
O     M1.1           // marker "rotation right is on"
AN   M2.0           // marker "stop" is not set
A     M2.2           // marker "rotation right"
=     M1.1           // marker "rotation right is on"
=     Q0.0           // relay "right" is on
=     Q0.1           // LED "right" is on
NETWORK 7
LDN   M1.0           // marker "rotation left is on"
AN   M1.1           // marker "rotation right is on"
=     Q1.0           // LED "motor off" is on
NETWORK 8
LD    Q1.0           // when the motor is stopped
EU

```

```

S      M2.3, B#1      // marker "delay 5s" is set
NETWORK 9
LD     M2.3           // marker "delay 5s"
TON   T33, W#+500    // start timer "delay 5 s - motor
                        // is off "

A      T33
R      M2.3, B#1     // marker "delay 5s" is 0
NETWORK 10
LD     M2.3           // marker "delay 5s"
A      SM0.5         // flash 1 sec.
=      Q1.1          // LED "delay 5 sec." is flashing
    
```

II. CONCLUSIONS

The new student workplace for PLC training was designed and implemented. The design of the student workplace is briefly mentioned in this paper. Here are reported two experiments to provide practice with PLC Simatic S7-200, CPU 226DC/DC/DC. The programs are tested using the student workplace. The experiments with PLC Simatic S7-200 are good teaching tools and they offer knowledge that cannot be learned at classical lectures. The experiments could be upgraded with adding some components and the area of applications could be extended. Thus the students have the opportunity to gain experience in basic concepts of PLC programming.

ACKNOWLEDGEMENT

The investigations and the paper preparation have been supported by Technical University of Sofia within the framework of the project №091uc014-03/2009.

REFERENCES

- [1] P.Yakimov, N.Tuliev, A.Iovev, S.Ovcharov, E.Balkanska, G. Mihov, *Investigation and Development of Electronic Instruments for PLC Programming Training*, Proceedings of The Seventh Int. Conference ELECTRONICS'2008, Book 3, pp.133-138.
- [2] A.Iovev, P.Yakimov, N.Tuliev, S.Ovcharov, E.Balkanska, E.Stoyanov, *Methodology for Teaching Basic Skills in PLC Programming Development*, Proceedings of The Seventh Int. Conference ELECTRONICS'2008, Book 3, pp.139-144
- [3] S7-200 Programmable Controller System Manual, Siemens AG, 2000, p. 9-15.
- [4] Berger H., *Automating with SIMATIC*, Siemens AG, 2003.
- [5] 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.

