Volume 56, Number 3, 2015

ACTA TECHNICA NAPOCENSIS
Electronics and Telecommunications

USING A PLC TO COMMAND A PLASTIC INJECTION MOLDING MACHINE

Anca APĂTEAN, Carmen MISAROȘ
Technical University of Cluj-Napoca, 26-28 G.Baritiu st.,Cluj-Napoca, Romania, 400027
+4 0264 401 568, anca.apatean@com.utcluj.ro

Abstract: **In the present paper, a control system for a Plastic Injection Mold Machine was proposed by using a Programmable Logic Controller (PLC) of type IMO iSmart ED-RD-20. The PLC is intended to be implemented in an automated industrial system to yield plastic products, such as flowerpots. For the practical implementation of a prototype of the system, besides the PLC, a power source (24 V DC), a temperature sensor, two DC motors, LEDs for signaling the inputs and outputs states, switches and buttons were also used. The project was realized in the Ladder Diagram (LD) language, comprising 8 LD programs and was simulated with the software simulator SMT.**

Keywords: *programmable logic controller, ladder diagram, plastic injection molding machine.*

I. INTRODUCTION

In today's applications, even the simplest of powered equipment needs to be controlled in some way, some even in an intelligent manner or adapting at diverse situations.

Even lighting, which can be controlled by a simple switch, can benefit from additional controls, like being active for a defined period, switch on and off at predetermined times or when an event occur, or switch on when one or more sensors are triggered, etc. Traditionally, such a control system would have been assembled using relays and timers, but today other solutions are also possible. PLCs (Programmable Logic Controllers) or "smart relays" is one possible alternative that offer the advantage of being reprogrammed to modify the existing functions or to enable the smart relay to be reused on a completely different application.

In industrial applications, controlling relays, servos, solenoids, and others is not just a matter of wiring in an Arduino-like platform and plugging in some code, because reliability is the major requirement. A PLC is generally classified as a microprocessor-based application, most experienced engineers or users claiming that PLCs were very reliable and easy to use in their projects.

The PLC, sometimes also called Programmed Logistical Equipment (ELP) is a system which appeared from the need to replace (the logical function made by the) electromagnetic relays with something else cheaper to maintain.

Generally, PLCs are used for conducting sequential processes of reduced complexity, from different branches of activity, like: machines used as tools, injection machines, presses, casting lines or galvanization of medium complexity, transfer lines, manipulators and industrial robots, etc. [1]

The PLC is more or less perceived like a general-use PC: it consists of hardware and software parts, implemented in the program memory. The hardware part is composed of the power supply, the central unit, and different input/output (I/O) modules. The software part is highly dependent on the user preferences, the type of the application but also the available PLC model to be used in the respective application [1], [2].

The aspects making the PLC different from a PC are: first, the PLC is designed to work in industrial environments, at high temperatures and humidity. A well designed PLC is not affected by the electric noise present in most industrial areas. Second, the PLC hardware and software are designed for an easier and specific use, not complicated (for an unexperienced user) and general, like in the case of the PC. The PLC does not have a disk or CD drive, but instead has a case with communication ports, a set of input/output terminals and the program built into its permanent memory. Next, PCs are complex computing machines capable of performing multiple programs simultaneously and in any order, while the PLC run a single program in an orderly and sequential form, from the first to the last. Another essential difference between a PLC and a PC consist in its supported programming language. [2], [3]

Programming a PLC consists of writing instruction sequences based on a phase diagram, a graph of states, Boolean equations, etc. Some PLCs also support graphical languages, thus allowing the visualization on a display of the implemented circuit.

From a structural point of view, models of PLCs can be:

- PLC in open structures, without an encasement;
- PLC with mono-block structure, with an encasement;
- PLC in modular structure (supporting extensions).

The PLC has multiple input ports and output ports, limited as number, but generally this can be increased by the use of additional modules (thus extending the system) called extensions.

Manuscript received May 6, 2015; revised July 5, 2015

Commands delivered by the PLC on the outputs are developed based on the measurement of some process variables, the system being provided with analog inputs and digital inputs (from 4 to 40 inputs, highly correlated with the PLC or extension module acquisition price).

Some analog inputs are used to take signals from the transducers in order to control and protect the system; others are used to implement the control function of some parameters (such as flow or debit, temperature, level, etc.). The input type and the values of the input signal can be set via software, at the PLC initialization stage.

The commands, as PLC outputs, can also be numerical (such as state of a contact) or analog (voltage or current signal). The analog outputs domain is determined also by program, but in the case of cheap systems it is set to a specific domain (for example 0 … 5 V) by the equipment manufacturer. The outputs can be contact state or voltage levels, adopted in such a way as to have sufficient power to drive an actuator or execution unit. Actuators are controlled elements that can change the flow of matter or the energy flow in a process; the most common are electric motors, hydraulic machines, thermal machines, mechanical devices.

In addition to the inputs and outputs externally accessible (connected directly to the process) inside the PLC, the following are also implemented in most PLCs on the market:y:

- internal relays used for performing logic operations (also called markers, flags);
- timers to allow the initiation of some activities at predetermined moments of time;
- counters (which include e.g. the number of times an entry has changed and generates a signal when the counter has reached a predetermined value), registers and memories;
- oscillators, allowing the timing and synchronization of activities;
- comparators, allowing the comparison of two signals and emphasizing their equality or inequality; systems to interrupt a program and switch to another sequence where a special event (e.g. in case of system failure).

One important PLC function is the one allowing the communication of the PLC with other devices (e.g. a microprocessor, a PC, a system with centralized control). Each PLC manufacturer provides a communication protocol through which two (or more) PLCs communicate in order to exchange data between them. Communication options are generally via standardized interfaces (e.g. Ethernet, RS232, RS485), which allow the PLC integration into an industrial network or even at the Internet (rarely accomplished, due to security problems). Sometimes wireless communication can be assured based on the IEEE 802.3 standard. These specifications generally mention both the physical environment (how binding together the equipments), signal levels, the transfer speed of the communication protocol between devices (which device transmits first, how the transmission of data is accomplished: how much data, the means of the signals from the lines, etc).

Smart relays can be used for different purposes, like signaling, protection, conditioning, control, etc. In applications from the industrial sector, control systems must operate in a continuously manner and in case of failure of the system, the protection systems have to be able to intervene [1]. A list of few possible types of PLCs with their features, but also their implementation price is given in [4].

II. PLC PROGRAMMING ELEMENTS

General purposes PLCs are supplied to the user without any program; this will be designed and written in the PLC program memory by the user, according to the application needs. Most PLCs can be programmed either by means of pushbuttons and small screen (on the PLC front panel) or, via a computer (PC) by writing a program. Still, to program a PLC only using buttons and the little display available on the PLC can be a frustrated experience. Besides, the software is generally very easy to use; the inconvenient is that the required software is not free and also there is the need for a programming cable, sometimes proprietary (very complicated and not advised to be develop by hand as DIY projects).

The program can be achieved by various programming methods, some more intuitive, and others that require programming knowledge. Typically, manufacturers provide the users with PLCs simulators as software that run on PC. The development of a PLC program is generally performed through programming instructions specific to each equipment manufacturer or through PLC programming languages. There are multiple languages for PLCs to be programmed, as defined in the IEC 1131 standard, using an alphanumeric display with Instruction List (IL) or Structured Text (ST), or using a graphical display with Function Block Diagram (FBD), Ladder Diagram (LD) or Sequential Function Chart (SFC). The most common one is the graphic LD language that uses graphic symbols for its elements, such as contacts, coils, etc.

II.1. PLC programming by Ladder Diagrams

PLC programming by the use of LD is easy to accomplish and allows the development of applications without any complex programming knowledge. Ladder Diagrams have its origins in electrical engineering, inheriting thus certain names and representations. The basic elements are contacts and coils [1], as illustrated in Fig.1. The representation of *contacts* is illustrated in Fig.1, generally denoted by I or E and being known as entries. Each input of an input module is recognized by the central unit when using the LD as a contact. The representation of *coils* is illustrated in Fig.1, by a letter Q. These are outputs from the automaton toward the process. Figure 1 illustrates a logical AND circuit with two inputs (Fig.1 a)), with contacts I1 and I2, a logical OR circuit (Fig.1 b)) and a logical NOT circuit (Fig.1 c)). At the output, Q1 can be the contact to a led for example.

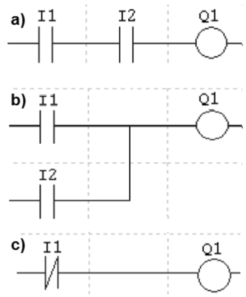

Figure 1. Logical AND, OR and NOT implemented with Ladder Diagrams

In Fig.1, the inputs I1 and I2 need both to be ON to switch ON the Q1 (Fig.1 a)), while I1 or I2 on ON position will switch the Q1 in ON (Fig.1 b)). The led on Q1 will be lighting only if the relay is not power supplied (for Fig.1 c)), the I1 contact remaining in the normal state (closed).

II.2. PLC programming by timers

The elements most used after contacts and coils for PLCs programming are *timers*. Still, the memory capacity limits the number of timers that may be software implemented in a PLC. The types of applications that require the use of timers are quite diverse and consist of the need to introduce time delays in exchanging certain outputs of the PLC.

There are 2 types of timers: with a 10 seconds *On–delay timer*, considering the same led lighting example, by pressing the contact button the led will be ON *for* 10 seconds; with the second type of timer, the *Off-delay* one, the led will be ON *after* 10 sec. one has pushed the button.

II.3. PLC programming using counters

Counters used by PLCs have the same function as those known (integrated). Control applications using counters can be achieved using the increment count, up to a certain value; after that, they cause an event occurring or causes the event maintaining until the counter reaches a preset value. A counter is set or reset depending on the signals present at its inputs and the counting frequency depends on the processing time of the program.

In order to program the PLC, first the type of the inputs and their evolution has to be established. The entries of a PLC may be of a digital type "0" or "1" or analog type, like a voltage level (usual range is between 0 and 10V DC). In the case of digital inputs, the logic level of "1" is determined by the supply voltage of the PLC. If the supply voltage of the PLC is 24 V DC, then the value for a logical "1" on inputs (i.e. the voltage for activating inputs) will be 24V DC. Generally, the sensors that activate a PLC inputs have the supply voltage range from 10 V to 30 V DC. In applications, cases in which only digital inputs or only analog inputs are used may be encountered, but generally, applications including PLCs are based on combinations of inputs, both digital and analog.

In production lines, a PLC can be used to control simple successive operations, in which case the commands delivered by the PLC are based on the information received from the sensors attached to its inputs. In this case, the program on which basis the automation process is realized, can be written as sequence of lines, generally implemented with LD or FBD elements.

The program execution begins with the first line from upper left, then follows from left to right, line by line until it reaches the last line where it will update the output (three example lines implemented with LDs are presented in Fig.1). It runs the program from the first line to the last one and repeat the cycle if necessary. This cycle is called *scanning* and start with reading the entries and ends by changing the implemented outputs. The program may contain interrupts and subroutines, which consists in testing certain conditions at certain times. The time in which a PLC goes through all the lines of the program depends on the processor frequency, the cycle time (including possible timers), and the number of lines from the program.

Each manufacturer has its own simulator for PLC programming, as the following examples:
- IMO PLCs use the SMT simulators;
- Siemens PLCs use Simatic Step 7;
- Omron PLCs use CX simulators;
- LG PLCs use GMWIN simulators;
- Allen Bradley PLCs - use RXLogix Studio 5000 or 5000 Logix Designer.

III. PLC DETAILS

To implement a possible application, the IMO iSmart SMT-ED-R20 PLC was considered. This can be programmed with the SMT Version 3.49 software by using LD. Having 12 inputs and outputs (but with possibility to extend the number of analog I/O with extensions), PWM (Pulse Width Modulation) function and possibility to monitor its functioning from a PC just to mention few of them, it is proper for a large spectrum of applications. From the most important features, the ones from Table 1 are generally mentioned in applications:

Table 1. Features of the IMO iSmart SMT-ED-R20 PLC

Feature	Values
Power supply:	100-240V AC or 24V DC
Programming languages:	LD and FBD
*i*Smart Memory Type:	32Kbyte Flash (EEPROM)
Execution Speed:	10ms/cycle LD, 6ms/cycle FBD
Program Size:	1200 Steps (300 Lines LD), (260 FBD Blocks)
LCD Display:	4 lines x 16 characters
Operating Temperature:	-10 to 60 °C
Humidity:	5 - 90% RH no frost
Vibration:	IEC60068-2-6 (0.075mm Amplitude/1g acceleration)
Impact Resistance:	IEC60068-2-28 (15g peak, 11ms duration)
Timers:	a max. no. of: 31 (LD), 250 (FBD)
Timing ranges:	0.01s – 9999 min
Counters:	a max. no. of: 31 (LD), 250 (FBD)
Highest count:	999999, with a resolution of 1
RTC:	no. available: 31 (LD), 250 (FBD), with a resolution of 1min.

For communication, some available functions are for remote I/O: 1 Master *i*Smart with program, 1 Slave used as I/O, and can be linked with up to 8 *i*Smarts in a local network. Available communication options include Modbus RTU, DeviceNet, Profibus, Ethernet.

IV. SYSTEM DESCRIPTION

In order to use the PLC in a possible automation application, a Plastic Injection Molding Machine (PIMM) has been designed. The PLC is intended to be implemented in an automated system to yield plastic products, such as flowerpots and to be used in industry. The use of the PLC in the frame of the PIMM is illustrated in Fig. 2, where the PIMM functioning process is briefly presented.

IV.1. Application description

The raw material, from which the desired product will be developed, has to be introduced in the PIMM through a hopper, as it can be noticed in Fig.2.

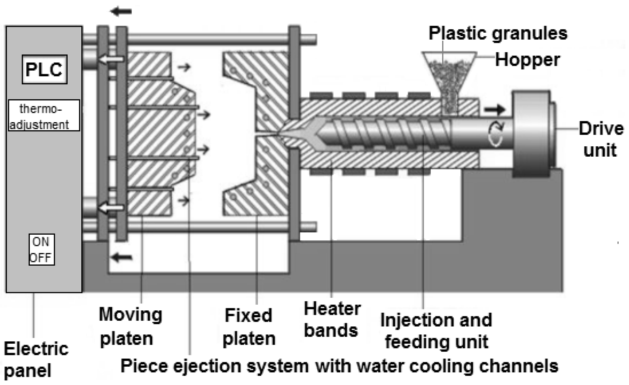

Figure 2. PIMM functioning process, inspired by [5].

The plastic granules will be melted (using the *heater bands*) and they will be monitored to reach a desired temperature, which is to about 165°C for real production. In the PIMM test-bench we designed, this temperature is measured with a LM335 temperature sensor and for the sake of simulation, the ambient temperature was considered. The finished product will result in the forming mold, which consists of two parts: one fixed (*fixed platen*) and one mobile (*moving platen*).

The process will be controlled by two DC motors which move in both directions. The moving platen and the fixed platen form the so-called *mold package* or simply *mold*. The injection molding process involves first heating and injecting plastic material under pressure into the closed metal mold. Second, the molten plastic cools (with the help of water showered by some channels in the piece ejecting system) and hardens into the desired shape inside the mold. Finally, the mold opens and allows the piece to be ejected for inspection, delivery or other secondary operations. The system is automatically controlled using an electrical control panel, inside which the PLC is mounted.

Being designed for the industrial sector of applications, the PIMM real functioning implies resources difficult to acquire. Also, considering the totality and complexity of manufacturing elements (involving knowledge from mechatronics, mechanical, electrical engineering), but also the need for a space large enough for proper operation, a compromise solution has been chosen: to simulate the PIMM using a prototype. Thus, in the prototype testbench for PIMM, the real sensors and execution elements have been replaced with ones at a smaller scale. Buttons, switches and LEDs have also been added, as inspired by the material found in the bibliography. The obtained testbench to simulate the industrial process is shown in Fig.3.

IV.2. Resources needed to achieve the prototype

To obtain the prototype from Fig.3 the following components were used: IMO iSmart SMT ED-R20 PLC with SMT 3.49 software, a stabilized power supply to 24V DC, one LM335 temperature sensor, 2 DC motors, a testbench with LEDs, buttons, switches to achieve the layout that connects all the other components.

On the layout, there are 5 LEDs for identifying active inputs (I1, I2, I3, I4, I8), 4 LEDs for identifying active outputs (Q1, Q2, Q3, Q4) and 1 led for Q7 to illustrate the alarm issue. To control the inputs, 4 buttons (for the first 4 contact entries), a button ON/OFF and another one for Reset were mounted on the panel.

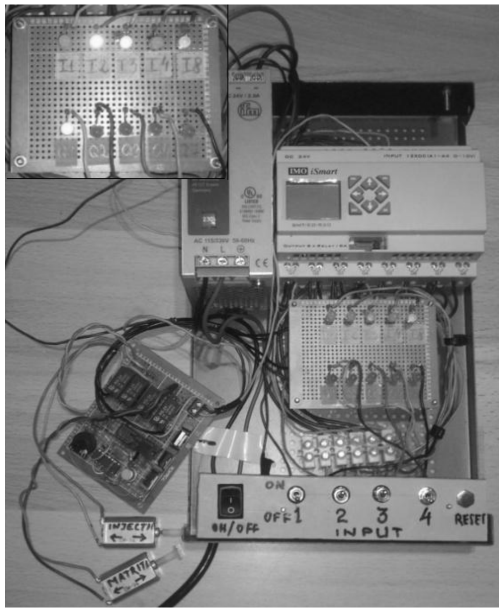

Figure 3. The prototype of the PIMM control system.

The corresponding LEDs were mounted on a PCB to indicate the status of the inputs and the outputs ordered, and another PCB is used to control the movement direction of the motors, respectively for the operation with the temperature sensor.

IV.3. The technological cycle

The entire PIMM system is based on a sequence of commands. To start the injection cycle, the mold has to be prepared for the injection process. As inputs, 5 different stimulus are used. For example, I1 indicates the state of the mold: in a NC state, the mold is closed and in a NO state the mold is opened. The I8 input shows that the material is at the proper temperature to be processed or injected. Different combinations of I2, I3 and I4 control the entire PIMM system functioning.

As outputs, Q1 is the command to close the mold, Q2 is a command to start the injection, Q3 is for plastic material feeding and Q4 is used to open the mold and eject the piece. With the command on the Q8 output, the alarm is issued.

In the system simulation phase, at the PLC outputs a sequence of commands that contain continuous voltage to power actuators (in our case the DC motor to control the mold and the DC motor to control the plunger injection mechanism) were obtained. System operation is accomplished by a temperature condition, using a temperature sensor LM335.

For the automation of the proposed application, the PLC has been programmed by the programming language LD. The whole PIMM system is based on a sequence of 8 LD commands. In order to start the injection cycle, the injection mold has to be prepared. These LD commands are specified in the following, as it is illustrated in Fig. 4 and Fig.5.

In Fig.4 the first stage of the PIMM functioning is presented, providing the first command of the process (through Q1). The counter C01 (visualized in the middle of the Fig.4) represent the start condition after a lot of pieces (in this case 4 pieces) was produced; the M02 is the start contact.

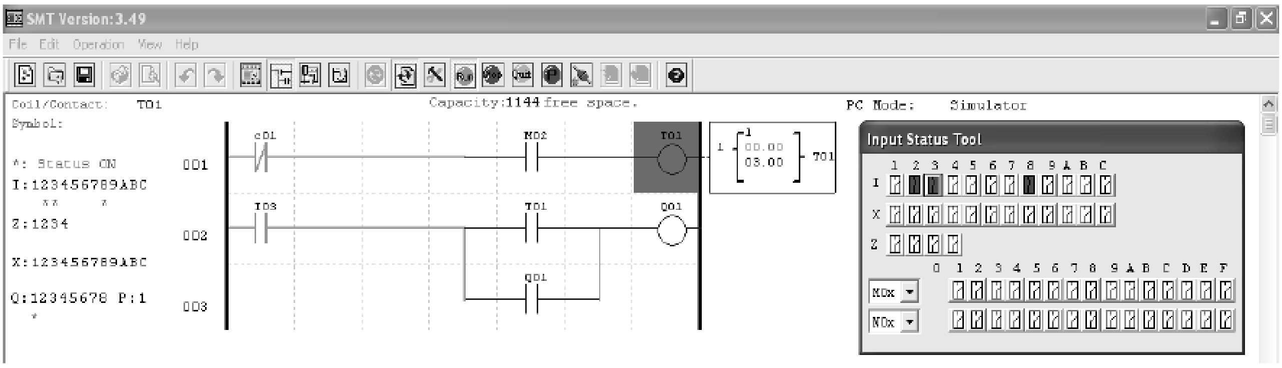

Figure 4. The first stage of the PIMM functioning process, providing the first command in the SMT simulator.

A timer T01, on-delay type, indicates three seconds; during this amount of time, the I3 input will reach a NC state. At the end of this LD command, the output (via Q1) will command the closure of the mold using the MG1 motor. Fig. 4 shows on the Input Status Tool window that I2, I3 and I8 are activated (in a NC state); at the output, Q1 will be selected, given the command to close the mold.

IV.4. The PIMM operation

The whole system will work only when the temperature condition is accomplished. In a real actual production of plastic products (e.g. polyethylene), the material will require a temperature of about 165°C. Lacking the necessary conditions to handle such a level of temperature, we used the ambient temperature, i.e. about 30°C. Thus, if the LED showing that the temperature condition is accomplished is lit, the PIMM system will work. This condition may allow or restrict the operation of the DC motors, i.e. the one which controls the mold (MG1) and the one that controls the injection (MG2). Still, if Q1, Q2, Q3 and Q4 are all open, the DC motors MG1 and MG2 will not function.

If the temperature condition is reached, then the motors operating condition is accomplished, i.e. the system will enter in one of the LD commands and will function accordingly. When the temperature will drop during the operation of a cycle, it will not affect the completion of that cycle, but in the next cycle the system will stop functioning. The condition for the system to start functioning is given by I8, which represent the temperature required to melt the plastic granules. On the LD lines from Fig.4, when a contact is entering a NC state, it will be viewed as follows: the Ix contact line turns green and slightly thickened, and on the contact simulator (Input Status Tool window) the button corresponding to Ix will be colored in red.

The transfer of the program in the PLC memory is realized through the standard RS232 interface on a serial port. After PLC connected with the PC, the RUN to option will be selected to start the simulation mode.

After executing all lines from a LD command program, the issue command (from Q1 to Q4 or combination of them) will be activated and the state of the inputs will be as illustrated in Fig. 5.

*LD command 1***:** Initially, to start running an operating cycle, the mold package must be opened. The counter C01 is the start condition after a preset lot of pieces (in this case 4 pieces) was produced, and the M02 is the start contact. The I3 entry, displayed on counters table, goes to a NC state and commands the output Q1: "Move the moving platen to close the mold!" or simpler "Close the mold".

After 3 seconds (indicated by the timer T1) the mold will be operated and will close by activating the motor MG1 to push the mold for closure. After starting off, the entry contact I2 will be issued and will reach a NC state.

LD command 2: Once the mold is totally closed (state illustrated by I1 in NC), the input I2 will be activated. When I2 will switch to NO status, the engine MG2 (representing the drive part of the injection unit where the material is melted) will start to function. The molten material will be introduced with pressure into the mold. In the program, an off-delay timer introduces a delay of three seconds: the contact line appears green, which means that the command was issued; when the amount of time was reached, the contact line will be disconnected. At the output, the commands Q1 (the mold should be kept closed, under pressure, or it may open) and Q2 (the injection) will appear.

LD command 3: Immediately after the start of injection, the I4 will be released and will go to a NC state. The output commands Q1 and Q2 still keep the mold under pressure (mold closed) and finish the injection. The left image in Figure 6 shows the transition from state 2 to state 3. The capture has been realized just before the I4 being activated. The outputs Q1 and Q2 are on, indicating the injection.

LD command 4: To complete the injection, the I3 entry will be operated, passing in NO state. This will order a delay of 3 seconds in starting the raw material supply system.

LD command 5: After the injection is finished, the material feed is performed. To issue this command, the I3 will switch to a NC state. The active output Q3 will start the motor MG2 in a reverse sense, making the supply material for a new piece to be inserted in the PIMM.

LD command 6: The transition to a new state will be controlled by input I4 which will pass into a NO state. This, in turn, will act to delay the opening of the mold, indicated by a 3 seconds timer (required for cooling the produced piece). The Q4 output will be active, to command the mold to open and eject the piece.

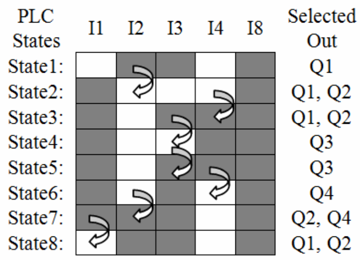

Figure 5. PLC functioning states and commands for PIMM, as defined with LD commands.

Figure 6. PLC functioning in 3 different states: passing from state 2 to state 3 (left), state 3(middle), state 4 (right)

LD command 7: After the delay time (representing the time needed for the piece to cool), the PLC will command the mobile part of the mold by a I2 in NC condition to open. This will activate the motor MG1 in a reverse sense, thus opening the mold. Meanwhile, the finished product is carried out and ejected. At the output, Q2 and Q4 will be activated, to command the injection part which is closed and the mold to open for throwing the finished product.

LD command 8: The I1 entry will pass in a NO condition and this will start next cycle command. It will return directly to state 1, then following injection for the second product. Thus, Q1 and Q2 outputs will be activated, carrying hence a new injection material for the second item.

After completing a number of pieces, the counter C01 will transmit a warning (alarm) which could be switched off by the Reset button, i.e. I7. In Figure 4 the Z variable shows a number of 4 pieces that was set for the first batch. As soon as this number of pieces was produced, C01 becomes active until it will be reset by the I7.

IV.5. Final considerations

The system presented in this paper is a testbench, aimed to incorporate a PLC and demonstrate its functioning into real industrial equipment. Like a trainer panel, it is aimed to prove that using PLCs could be a viable solution to develop projects in which some successive, repetitive operations are required. By the experiments we developed, the testbench has function in a proper manner, executing each command as it was programmed, according to the LD programs. Leds where turn on and off according to the state of the inputs and outputs they were related to, and such as Fig.5 and Fig. 6 shows. A mechanical faulty will not have a huge impact for the system due to the fact that it is functioning in successive operations: i.e. in each step one or more conditions have to be accomplished in order that something happens. If the inputs are not in the proper state, the PLC will not issue the next command, so a human operator should intervene and repair a possible malfunction problem.

For the real implementation, there are a number of blends of different plastic-like materials, designed to achieve different levels of performance, depending on the desired properties: chemical resistance, impact resistance, UV resistance, flexible or rigid, durable, gloss, texture, transparency or not, cost, etc. Additional strength may be gained through the addition of fillers (e.g. talc, glass) or additives (to improve UV stability or flame reduction or provide antimicrobial properties among others). [6]

For a control system implemented in the industry, generally either a PC or a PLC is chosen. The main aspects to be considered in such a project are: security, reliability, performance, functionality and cost. These all should ensure a long-term process with also the ease of re-programming the device.

Numerous different models are available, from suppliers of automation products: Allen-Bradley, Hitachi, IMO, LG Industrial Systems, Mitsubishi, Motorola, Omron, Rockwell Automation, Schneider Automation, Siemens, Toshiba, Yokogawa [7]. Their marketing strategy is based on using newest technologies, aiming thus to improve their price/performance curves with every new product they develop.

PLCs can be met in different applications: used for a variety of lifts, disabled access systems, home-mobility lifts, operating doors and other gadgets, controlling pumps, etc. Due to its compact size, easy programming, and communication options, they are also integrated into heating and ventilation systems, to control irrigation systems, animal feed systems, water tank levels, etc. The strength of using a PLC instead of PC in industrial applications is the ease of reprogramming its operations and of maintaining it, its size, but also higher reliability and security [2],[3].

Besides, the developed PLC testbench we proposed was also aimed to provide a low-cost trainer panel for students who want to accommodate with the PLCs actually found in manufacturing lines. Thus, they could build their own LD program in an controlled environment, without the risk to cause damage in a larger machine industrial type.

V. CONCLUSION

In the present paper, a control system for a plastic injection mold machine was proposed by using a PLC of type IMO iSmart ED-RD-20. For the practical implementation of a prototype of the system, besides the PLC, the following were necessary: a power source, stabilized at 24 V DC, a temperature sensor LM335, two DC motors, LEDs for signaling the inputs and outputs states, and the software simulator SMT Version 3.49. The project was realized in the Ladder Diagram language. After creating the 8 LD programs, on a PC a simulation was performed using the simulator to verify the accuracy and avoid possible implementation errors.

As their large-scale implementation in production sector demonstrates it, the error rate is lower when using PLCs to command industrial processes. Generally, such systems are claimed to be more reliable than PC-based ones or platforms Arduino-like, especially as concern their functioning condition. PLCs are professional designed hardware and software, more robust, adopted by the industries on large scale internationally, especially by their ease of use by plant electricians and technicians.

REFERENCES

[1] Hackworth, J.R., and Hackworth, F.D.. *Programmable Logic Controllers: Programming Methods and Applications*. Pearson, 2004.

[2] Frank D. Petruzella, Programmable Logic Controllers, 3rd ed,. 2005

[3] PC vs. PLC: key factors in comparing control options, Rexroth Bosch Group, Accessed 30Aug.2015: http://www.automation.com /pdf_articles/Rexroth_PLCvsPC_L.pdf

[4] http://www.entertron.com/smartrelay.htm

[5] http://www.sgi-tooling.co.uk/processmaterials/

[6]http://www.rutlandplastics.co.uk/advice/thermoplastic_materi als.html

[7] List of PLCs from a Romanian provider http://electroblue.ro/images/1368102196rt.pdf