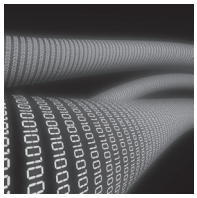

Archives of
Transport System
Telematics

Volume 4

Issue 3

September 2011

Safety related control systems for railway signalling applications with a safety PLC

J. ŽDÁNSKY[a], K. RÁSTOČNÝ[a], J. HRBČEK[a]

[a] Faculty of Electrical Engineering, University of Žilina, Univerzitná 8215/1, 010 26 Žilina, Slovak Republic

EMAIL: juraj.zdansky@fel.uniza.sk

ABSTRACT

Nowadays, there are available on the market not only standard PLCs (Programmable Logic Controller) but also safety PLC´s. These are primarily designed for industrial applications. Their guaranteed safety properties, however, enable to be used in applications, in which the usage of PLC has not been common until now. The aim of this article is to focus on problems related to the usage of safety PLC in railway signalling systems.

KEYWORDS: PLC - Programmable Logic Controller, safety PLC, SIL – Safety Integrity Level

1. Introduction

One of the appropriate devices for an automatic control realisation are programmable logic controllers (PLC - Programmable Logic Controller), to which also their circuit and technological solution responds. Because of its parameters, they are becoming favourite means for solution of different control tasks (e.g. [1]). Nowadays, there is available the large-scaled PLC´s assortment from different producers on the market. Their application possibilities and comfort by their programming and debugging make them tools which cannot be compared with those at the beginning of the PLC development. Producers attempt to constantly innovate the possibilities of the PLC. Main trends in the PLC development can be summed up into the following areas:

- comfort increase by programming and debugging of the PLC – first of all, it concerns automation of some actions by programming and possibilities expansion of their programming by various programmable languages (so the PLC are becoming more available to different user groups); nowadays, almost all programmable environments meet the needs defined in the standard [2];
- increase of the application possibilities of the PLC – it concerns development of new modules which belong to the modular structure of the PLC; as a typical example can be mentioned modules for servo – unit, intelligent sensors, high – speed counters etc.; modules of analogue inputs and outputs are self-evident; this reality relates to the fact that some producers leave the traditional name Programmable Logic Controller and use the name Programmable Controller; also this indicates that traditional, mainly logic character of the PLC, is becoming the past;
- increase of communication possibilities of the PLC – PLC fulfils only some of control functions of the entire control system in modern distributed control systems (generally the PLC are used on the process level) and

must be able to collaborate with parts of the system on other control levels or other control systems (e.g. [3], [4] deals with communication possibilities of modern PLC);
- increase of reliability and safety parameters of the PLC – increase of these parameters relates indirectly to increase of application possibilities of the PLC even in areas where it has not been possible until now (for example control of the safety-critical process).

The main difference in producers approaches to increase of reliability and safety parameters of the PLC is the fact that some producers follow these parameters separately (they offer the PLC with increased reliability or the PLC with increased safety) and some of them offer the PLC with modular structure which enables to follow increase of reliability and safety parameters at the same time.

For PLC, having the property that after occurrence of failure, they will remain in the original condition (if it is not critical in view of control process) or will go to a predefined safe state (usually a setting of outputs to the state log. 0; this feature is necessary to take into consideration by usage of PLC), the name safety PLC is being used. For PLC, having the property to be able to perform its function even in the presence of hardware failures or errors in program, the name fault-tolerant PLC is being used.

Commercially available safety PLCs are principally intended for industrial applications up to the required level of safety integrity SIL3. Dangers that may occur in railway transport are associated with serious human consequences (transport of people) and therefore, the systems for control of train drives have to be usually realised with SIL4.

Usage of commercial safety PLC for such applications is not possible because the increase of SIL to SIL4 would signify intervention to the technical solution of PLC and this is practically impossible for user. Companies solve this problem by developing of special safety PLC, which are certified for SIL4 (for example, system NEXUS from První Signalní, a.s.).

In railway transport, however, there also exist applications in which for the reduction of risk (arising from the control process) to the tolerable level, it is sufficient to apply technical measures with lower SIL than SIL4. It regards mostly the traffic control on the hump yards, on factory railways and in recent years there are discussions about level crossing systems devices on secondary lines where little ground speed and low traffic intensity are.

So that the safety PLC could be certified for the required safety integrity level, it has to meet the requirements for SIL against systematic failures (especially application software errors) and also against random failures (mostly the failures of hardware components). Meeting these requirements is characterized by certain specifications in application, to which this contribution is dedicated to.

2. Ensuring the safety integrity level against systematic failures

Systematic failures neither occur as a result of system aging nor have a random character, but their presence is linked to a particular situation and state of the system. In case of PLC, systematic failures are associated with software errors caused by system proposal.

When creating a control system based on PLC, hardware part of the control system is built-up on the basis of modules offered by the selected manufacturer (s) of PLC, whereby the interfaces are clearly given and it is not necessary to deal with their definition. After determining the architecture of the control system, the centre of its creation will be resting in creating of application program because this one implements the required safety and control functions of the system.

One of the most important activities in developing of safety critical control system is to define functional requirements. If the specification of functional requirements is made only by an informal specification, it will be a high probability (especially if it is a more complex system) of failures occurrence in software due to its incompleteness and often little lucidity. Specification of functional requirements must be done so as to be clear, understandable, complete, consistent and controllable. Therefore it is recommended that the specification of functional requirements would be carried out on the basis of semi-formal and formal methods. These methods are oriented to minimize systematic errors in software and greatly help to enhance the functional safety of the system.

If the PLC is used to control the discrete-event systems, it can be regarded as a sequential system. The mathematical model of such system is a finite automaton.

The finite automaton M is arranged by:

$$M = (A, S, U, p, v) \tag{1}$$

where A is a set of input vectors, S is a set of states and U is a set of output vectors; p and v are transforms:

$$p : S \times A \to S$$
$$v : S \times A \to U \text{ (eventually } v : S \to U \text{)}, \tag{2}$$

where the transform p is called transfer function and transform v is output function [5].

The finite automaton, whose output function v has the domain range $S \times A$, i.e. it assigns certain output symbol to each pair (state, input), is called Mealy's automaton. If the transform v has its domain range S, i.e. it assigns the

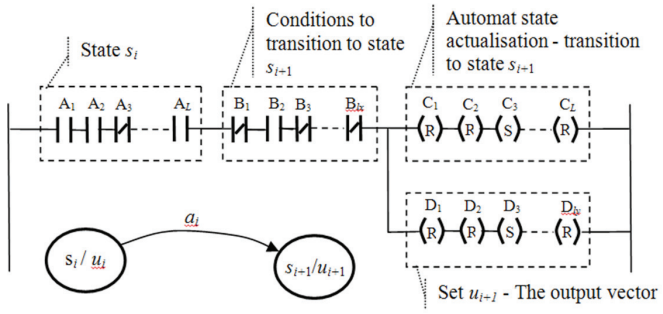

Fig. 1. Structure of program ladders created on the basis of state diagram

output symbol to each state, then it is Moor's automaton. Both of these types of automata can be implemented as synchronous or asynchronous sequential circuit.

Finite automaton can be described as following:

- Mathematical expression – sequential circuit's behaviour in discrete time area can be described by expressions:

$$s_{(t+1)} = s'_{(t)} = p_{(s(t),a(t))}$$
$$u_{(t)} = v_{(s(t),a(t))} \qquad (3)$$

where the symbols a(t) and u(t) symbolize particular input and output vectors of the system in time t; symbols s(t) and s(t+1) symbolize the particular states in time t and t+1. In fact, the main task is the compilation of the Boolean functions, of which the elements of state vector s(t+1) and output vector u(t) can be calculated on the basis of input vector elements a(t) and the state vector in the previous time s(t).

- Table representation – it defines the input words which may cause state transition.
- Graphically – by the state diagram. State diagram is a directed graph whose nodes represent the finite automaton states and directed edges correspond to the transitions between states. The edges are rated by inputs vectors $a_l \in A$, which activate transition of finite automaton from one state to any other. If each state of automaton is assigned the output vector, then it is a Moor's automaton. If the output vector is assigned to transition, then it is a Mealy's automat. In practise, we can see a combined state diagram (the output vectors are assigned to states and transitions, too).

From the mentioned ways of finite automaton notations, the state diagram can be considered as the most suitable from the view of its usage for program creation. This is because the fact that it is easily understood by the people involved in system specification and can be used for direct creation/generating of software for control system.

The basic idea of the state diagram using for the program creation consists of assigning a code to diagram states ($s_1, s_2, \ldots, s_l$). So as the control system could fulfil its function according to state diagram, the code of actual state must be kept in its mind constantly and the conditions for transition from this state to another state must be evaluated. The transition can be initialized by input word / words from the set A. In case of fulfilment of the conditions, the place of the actual state is replaced with a code of a new state in program memory and the actions connected with the new state, eventually with the transition, are executed (the setting of output word / words from the set U). In the Fig. 1 the structure of ladder of the program is shown, created according to state diagram in principle.

Program shown in the Fig. 1 uses ladder logic. It is a graphic method of programming, based on techniques used for relay circuits. Considering its lucidity it can be used advantageously for programming of the safety critical control systems. Such a built-up program can be directly implemented into the safety PLC. For example, programming language F-LAD (Fail-safe ladder logic) can be used. This language differs from standard language LAD mostly by limited instruction file and accurate defining of individual subprograms callings.

In the Fig. 1 there is applied a binary code for coding of states. Using the binary code is not a condition. However, it seems to be advantageous in regard to its simplicity, lucidity and instructions applicable in F-LAD language. In case of usage this methodology for programming of standard PLC e.g. decadic code can be used, eventually code consisting of alphanumerical symbols.

From the perspective of the PLC functionality it does not matter which way of states coding is being used. The coding of states, however, will have an impact especially on the speed of program running. The influence of coding on the program speed will be more noticeable with increasing complexity of the program. This is because of the fact that by evaluating the program created according to Fig. 1, the number of comparison actions (the evaluation, in which state the currently control system is) is adequate to the number of automaton states.

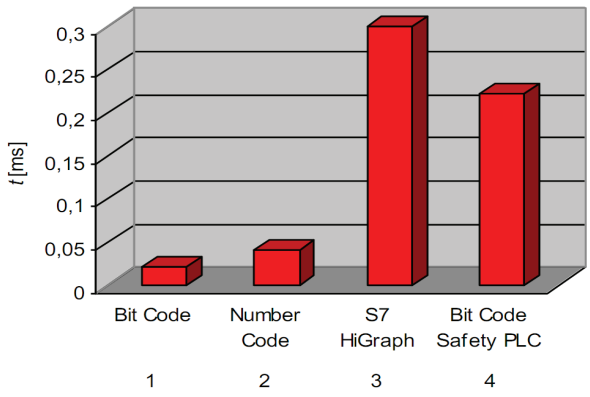

Fig. 2. Comparison of execution times of the programs created by various coding of states

The influence of the selected code to the execution speed of the program is shown in the Fig. 2. The Fig. 2 compares the execution time of one cycle of the program for standard and safety PLC. For standard PLC, various codes of states are being used. The measurements were done for control system SIMATIC S7-300 in standard and safety version. A very simple program was implemented. This example assesses pushing the button. After its pushing and releasing, the output is activated. After repeated pushing the button it comes to deactivation of output.

The first and the fourth column show the program execution time under the same conditions for standard and safety PLC. Significantly longer program execution time of safety PLC is caused by a producer defined functions. These functions relate to the required level of safety and the user can not influence them in any way.

3. Ensuring the safety integrity level against random failures

The system safety integrity level against random failures is mostly influenced by:

- Structure of the system;
- Intensity of system elements failures;
- Diagnostic features of the system
- Mutual independence of the system channels, eventually common cause failures (CCFs), in case of multi-channel system.

In railway applications, safety-relevant control systems with safety PLC are being used mainly on process level and therefore the part of the control system are, apart from safety PLC, sensors and actuators, too (Fig. 3; SRCS-R is Safety Related Control System for Railway).

The final level of the system safety integrity SRCS-R is dependent not only on safety features of safety PLC, but also on reliable and safety features of sensors and actuators and the way of their connection to the safety PLC and their reliability and safety parameters. Therefore it is necessary to pay attention to sensors and actuators selection, the way of their connection and setting of input/output circuits of the PLC (to which the sensors and actuators are connected).

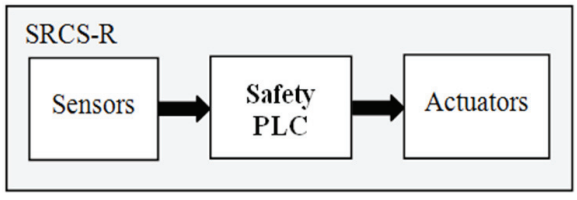

Fig. 3. Block scheme SRCS-R with safety PLC

According to producers data (for example, in the document [6]) 90% of dangerous failures is caused by sensors and actuators failures and only 10% of dangerous failures is caused by the safety PLC failures. Setting of appropriate parameters of input / output circuits needs to be done not only with respect to the required level of safety integrity, but it is also necessary to take into consideration the parameters of the connected sensors, respectively actuators (for example, by some actuators there is not acceptable pulse testing).

The basic building elements of safety PLC are modules (processor module, input / output module, ...). Each module is defined by the intensity of dangerous failures. Knowledge of the intensity of dangerous failures is a necessary prerequisite for quantitative assessment of the assembled SRCS-R. The most commonly used model to evaluate the safety SRCS-R with safety PLC is a serial model. Such a model is recommended by the PLC producers themselves. The reason for using of the serial model is its simplicity. In the view of safety, such a model is acceptable because it comes from assumption that a dangerous failure of any module causes a dangerous failure of the whole control system. For example, for the assembly of the safety PLC is valid:

$$\lambda_{PLC}^{N} = \sum_{i=1}^{n} \lambda_{M_i}^{N} \leq \sum_{i=1}^{n} \lambda_{M_i} \qquad (4)$$

where λ_{PLC}^{N} is intensity of dangerous failures of the PLC, $\lambda_{M_i}^{N}$ is the intensity of dangerous failures of i-th module of the PLC and λ_{M_i} is intensity of failures of i-th module of the PLC and n is the number of PLC modules.

Likewise, we can determine SIL for SRCS-R. This means that:

$$\lambda_{SRCS}^{N} = \sum_{i=1}^{n} \lambda_{S_i}^{N} + \sum_{i=1}^{m} \lambda_{A_i}^{N} + \lambda_{PLC}^{N} \qquad (5)$$

where λ_{SRCS}^{N} is intensity of dangerous failures SRCS-R, $\lambda_{S_i}^{N}$ is intensity of dangerous failures of i-th sensor, $\lambda_{A_i}^{N}$ is intensity of dangerous failures of i-th actuator, n is number of sensors, m is number of actuators and λ_{PLC}^{N} is intensity of dangerous failures of PLC.

The standard [7] does not define SIL for the system, but for the safety function of the system. It means that for

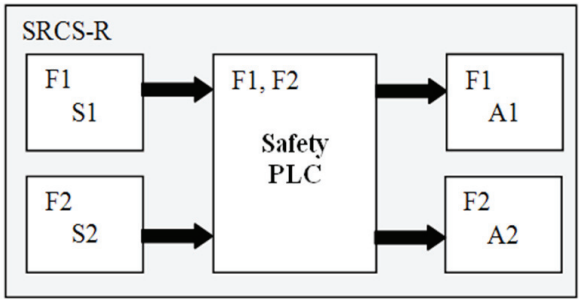

Fig. 4. Example of a safety function realisation through SRCS-R

the calculation of intensity of dangerous failures of a given safety function are relevant those system elements, which are involved in its implementation.

Let us define two safety functions F1, F2.

If the function F1 is realised by sensor S1, safety PLC and actuator A1 (Fig. 4), then

$$\lambda_{F1}^{N} = \lambda_{S1}^{N} + \lambda_{PLC}^{N} + \lambda_{A1}^{N} \qquad (6)$$

where λ_{F1}^{N} is intensity of dangerous failures of function F1, λ_{S1}^{N} is intensity of dangerous failures of sensor S1, λ_{PLC}^{N} is intensity of dangerous failures of PLC and λ_{A1}^{N} is intensity of dangerous failures of actuator A1.

If the function F2 is realised by sensor S2, safety PLC and actuator A2 (Fig. 4), then

$$\lambda_{F2}^{N} = \lambda_{S2}^{N} + \lambda_{PLC}^{N} + \lambda_{A2}^{N} \qquad (7)$$

where λ_{F2}^{N} is intensity of dangerous failures of function F2, λ_{S2}^{N} is intensity of dangerous failures of sensor S2, λ_{PLC}^{N} is intensity of dangerous failures of PLC and λ_{A2}^{N} is intensity of dangerous failures of actuator A2.

For the intensity of dangerous failures SRCS-R according to Fig. 4 is valid that

$$\lambda_{SRCS}^{N} = \lambda_{S1}^{N} + \lambda_{S2}^{N} + \lambda_{PLC}^{N} + \lambda_{A1}^{N} + \lambda_{A2}^{N} \qquad (8)$$

From expressions (6), (7) and (8) it is evident that

$$\lambda_{SRCS}^{N} \neq \lambda_{F1}^{N} + \lambda_{F2}^{N} \qquad (9)$$

The way of sensors connecting depends on sensors features and requirements for safety of SRCS-R. For example, in railway applications, there is very often required the evaluation of the state of the contact button while by pushing the button it is necessary to execute the required safety function. And there are several options of button connecting to safety PLC. In Fig. 5 two of these options are shown.

The connection according to Fig. 5 a) can be used when the contact button is closed in a basic state and there is excluded contact failure - short circuit - with such a probability which corresponds to the required SIL of the given safety function. The connection according to Fig. 5 b) does not impose any special requirements for safety features of the button. The intensity of dangerous failures, with which the button circuit contributes to the overall intensity of dangerous failures of the required safety function, can be calculated from the expression:

$$\lambda_{TL}^{N} \cong 2.\lambda_{K1}.\lambda_{K2}.t_{CH} \qquad (10)$$

where λ_{K1} is intensity of contact failures K1, λ_{K2} is intensity of contact failures K2 and t_{CH} is the maximum value of time between two pushing of button. Contacts K1 and K2 are controlled by one button.

Analogic approach can be used for connection of actuators on the output of safety PLC. Output modules of safety PLC, like the input modules, enable single-channel or dual-channel connection of actuators.

Failure detection (detection of fault) and the subsequent negation of failure (negation of fault) are crucial for ensuring the required level of system safety. SIL of the system is influenced by two features of diagnostic:

- Fault detection time;
- Diagnostic coverage.

Measures for the negation of fault can only be effective if the fault is identified. Therefore SCRS-R contains, apart from functional diagnostic, test diagnostic, too. Diagnostic system implements specific (testing) signals to the object of diagnosis and analyses the responses. In case that such a diagnostic system is also being used when the object is in use (operating diagnostic), test signals may not interfere with normal operation of the object. Diagnostic test is being used in operation to detect faults which do not appear immediately in the object operation, but by the change in the system or in combination with other fault they can lead to a critical state. In this case, test procedures must be analysed for safety, because they can be a source of faults themselves.

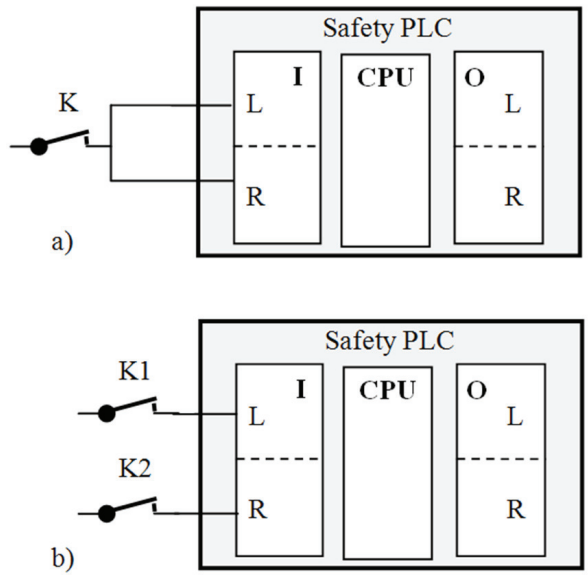

Fig. 5. Example of a safety function realisation through SRCS-R

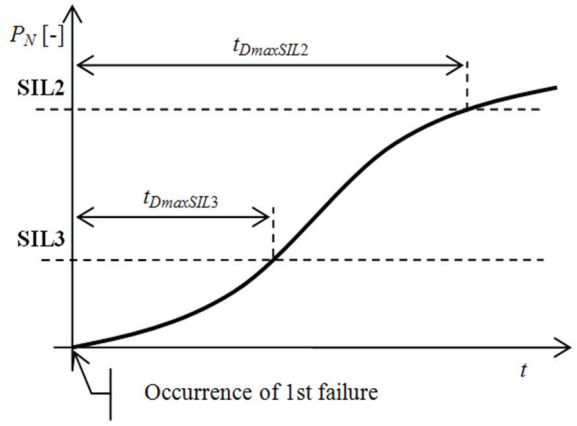

Fig. 6. The probability of dangerous failure occurrence SRCS-R with safety PLC

The influence of failure detection time on probability of dangerous failure occurrence SCRS-R with the safety PLC is shown in Fig. 6 (Note: it would also be right to consider the time needed for fault negation; this time is usually not considered because it is usually negligible in comparison with the time of failure detection).

The graph (Fig. 6.) shows that with increasing time of detection, the probability of dangerous failure occurrence of the system is increasing, too. The required level of safety SRCS-R can be achieved either by continuous diagnostics (on-line tests), or by regular controls (off-line). The graph shows the maximum time allowed for the detection and negation of failure for the required SIL ($t_{DmaxSIL2}$, $t_{DmaxSIL3}$). In order to control the interface between the safety PLC and controlled objects (COs), feedback must be used. A typical example of connection of safety PLC outputs with feedback, that allows early detection of failure, is illustrated in Figure 7 b). In this case, the application program must include testing procedures to control the functionality of switches S11, S21 and also the mechanism of failure negation.

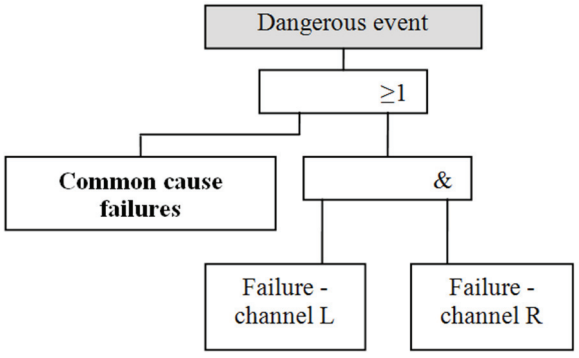

Fig. 8. The influence of CCFs on safety of SRCS-R

SCRS-R with a safety PLC does not have to include feedback with aim to detect a failure of output switches (Fig. 7 a)) if:

- The time required to detect failure is greater than or equal to the required time of system life, i.e. SIL achieved without feedback is sufficient for the given application;
- Possible failures are detected during regular inspections; the time between these checks must conform to the required SIL.

If a continuous diagnostic (on-line testing) is not able to detect all potentially dangerous failures (diagnostic coverage c < 1), then it is necessary to combine the continuous diagnostic with regular checks.

Safety of multichannel systems can be even threatened by a common cause failure. If we consider two-channel system, then the impact of common cause failures to safety SCRS-R can be illustrated by a simple tree in Fig. 8.

It should be noted that the mutual independence of the channels is not only related to the technical solution, but also to the independence of persons (organizational measure) involved in the developing of the system.

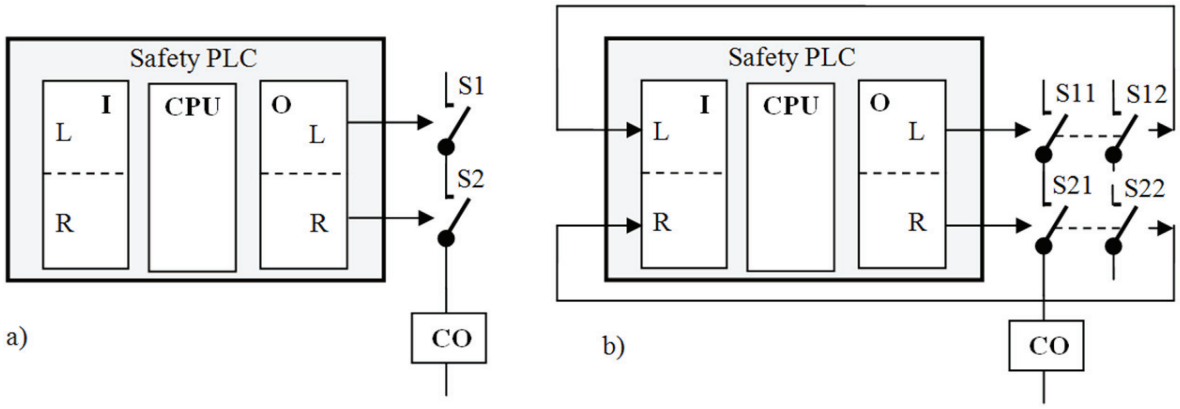

Fig. 7. Connecting of controlled object

4. Conclusion

The usage of safety PLC in safety systems restricts the maximum attainable safety integrity level SIL3 (without additional hardware and software components). For applications, in which the safety integrity level is sufficient, safety PLC can considerably simplify the proposal and implementation of safety system. Then the greatest emphasis should be given on connecting of sensors and actuators and the parameters influencing the way of their assessment.

Acknowledgement

This publication is the result of the project implementation: Centre of excellence for systems and services of intelligent transport, ITMS 26220120028 supported by the Research & Development Operational Programme funded by the ERDF.

Bibliography

[1] MACEK, P. - ŠIMÁK, V. - ROFÁR, J. - MIČIETA, B.: Application of new automation technologies in laboratory of automation and simulation of processes in production.. In: Acta Mechanica Slovaca. - ISSN 1335-2393. - Roč. 12, No. 1-A (2008), p. 317-320.

[2] STN IEC 61131-3: Programovateľné regulátory. Časť 3: Programovacie jazyky. 2003

[3] BÉLAI, I. - DRAHOŠ, P.: The Industrial Communication Systems PROFIBUS and PROFInet. In: Applied Natural Sciences 2009 : International Conference. Trnava, Slovak Republic, 7.-9.10.2009. - Trnava : Univerzita sv. Cyrila a Metoda v Trnave, 2009. - ISBN 978-80-8105-127-2. - p. 329-336

[4] BEZÁK, T. - STRÉMY, M. - HUSÁROVÁ, B.: Distributed control systems modelling using PROFInet CBA. In: Annals of DAAAM and Proceedings of DAAAM Symposium. - ISSN 1726-9679. - Vol. 21, No 1. Annals of DAAAM for 2010 & Proceedings of the 21st International DAAAM Symposium "Intelligent Manufacturing & Automation: Focus on Interdisciplinary Solutions" 20-23rd October 2010, Zadar, Croatia. - Vienna : DAAAM International Vienna, 2010. - ISBN 978-3-901509-73-5, p. 559-560

[5] FRIŠTACKÝ, N. - KOLESÁR, M. - KOLENIČKA, J. - HLAVATÝ, J.: Logické systémy. Publisher Alfa, Bratislava, 1986

[6] Publication: 1756-RM001E-EN-P - November 2006, Available at www.ab.com/manuals

[7] EN 50 129: Railway applications: Safety related electronic systems. 2003