TECHNICAL TRANSACTIONS | CZASOPISMO TECHNICZNE
ELECTRICAL ENGINEERING | ELEKTROTECHNIKA

1-E/2016

DOI: 10.4467/2353737XCT.16.032.5294

RYSZARD MIELNIK[*]

SYNTHESIS OF CONTROLLER FOR RAILWAY – LEVEL CROSSING DEVICES USING PETRI NETS AND STATE MACHINE

SYNTEZA KONTROLERA URZĄDZEŃ PRZEJAZDOWYCH Z WYKORZYSTANIEM SIECI PETRIEGO I MASZYNY STANÓW

Abstract

The design of modern digital measurement-control systems applied in such fields as nuclear and power, chemical, air and rail transport as well as military requires a special approach to the philosophy of design, manufacture and the use of such systems. Therefore, attention should be focused on safeguarding the required level of reliability and safety of working conditions of such systems. On an example of a controller for an automatic railway crossing devices is presented synthesis of this controller. In this fact were used Petri nets – to modeling this controller and state machine – to make programming application.

Keywords: Petri nets, state machine, discrete-event systems, real-time systems

Streszczenie

Projektowanie cyfrowych systemów pomiarowo-sterujących stosowanych w takich obszarach jak: energetyka jądrowa, przemysł chemiczny, transport powietrzny i kolejowy oraz zastosowania militarne wymaga specjalnego podejścia do filozofii projektowania, produkcji i eksploatacji tego typu. Systemy te wymagają zapewnienia odpowiedniego poziomu niezawodności i bezpieczeństwa. Na przykładzie sterownika samoczynnej sygnalizacji przejazdowej przedstawiono jego syntezę z wykorzystaniem sieci Petriego – do modelowania funkcji kontrolera, oraz maszyny stanów – do wykonania oprogramowania.

Słowa kluczowe: sieci Petriego, maszyna stanów, dyskretne systemy zdarzeń, systemy czasu rzeczywistego

[*] Ph.D. Eng. Ryszard Mielnik, Institute of Electromechanical Energy Conversions, Faculty of Electrical and Computer Engineering, Cracow University of Technology.

1. Introduction

Nowadays, measurement-control digital systems (MCD) are commonly used in industrial processes [12]. These systems are built on the basis of hardware platforms, such as PLC's, industrial computers or embedded systems. Functionality of MCD systems depends on the implemented user software applications. Thus, the quality of the work of these systems is influenced not only by the hardware platform, but also by its user's software. A major problem in the implementation of MCD systems is caused by the design, construction and testing, both the software as well as the system as a whole. The need to ensure high reliability of the software and the reliability of the entire MCD systems is required especially in the so-called critical applications. In these applications, malfunction or damage of the MCD systems during this operation can lead to abnormal functioning of the supervised process, which can result in the loss of life and material waste. Therefore, formal methods for modelling MCD systems are sought. Suitable instruments for this purpose may be the Petri nets. They are graphic and formal tools for modelling, formal analysis and design of discrete event systems. The model represented by such a net allows for the analysis of the characteristics of the system´s behaviour, and allows its evaluation within each phase of the system´s life cycle.

The aim of this paper is to present the possibilities of using Petri nets in modelling MCD systems in critical application, which includes railway transport. The chosen example of such an application can be automatic railway crossing devices [2, 4, 6]. The practical application of this controller was made by using the state machine and statechart tool of graphical software environment – LabView.

2. Petri nets

In 1962, Carl A. Petri developed a tool, based on the graph theory, called Petri nets in order to model discrete systems [9, 11]. As a result of the long-term development of the theory of these nets, a number of classes of Petri nets emerged, for which wide applications have been found [3]. As the graphical tool for modelling complex systems, Petri nets allow, during the development phase of assumptions and design requirements, good and clear communication between the designers and customers, thanks to which a comprehensive requirement specification was created, which does not contain vague rules and inconvenient for the customers' formal record. The resulting graphic model of the system can also be applied in computer graphics simulation environments of Petri nets, resulting in a virtual prototype of the system being designed [1, 3]. This allows the designers to quickly, interactively and comprehensively test the functionality and features at every stage of its life cycle. Petri nets are ideal for system modelling of discrete events and analysis of their properties [1, 3, 11], such as synchronisation of processes, asynchronous events, competing tasks, conflict shared resources, jams etc. Petri nets, as a formal modelling tool, are described by linear algebraic equations or other mathematical formula reflecting the behaviour of the designed system [11]. The ability of formal verification of the model is quite important in the construction of industrial MCD systems.

Petri nets extended with time model can be used to model embedded systems [11], real-time systems [1, 10], and in particular real-time safety of critical systems, which include railway traffic control systems [2, 4].

2.1. Low-level Petri nets

One of the essential characteristics of Petri nets is their relatively simple and intuitive graphical representation. These nets, which are derived from the theory of directed bipartite graphs, allow the use of terms specific to these graphs [11]. These graphs contain two types of vertexes called places and transitions. The vertexes represent the states or activity of the modelled system. They are connected with each other by arcs in such a way that arcs only connect vertexes of different types. The resulting graphic notation is characterised by a net structure. Graphic representation and structural properties are common features for all classes of nets [9, 11].

To be able to fully model the system, taking into account its dynamics, a definition of nets was extended by net marking. It represents the state of the modelled system. Net status is represented by token changes as a result of pass-time simulation of the net. Therefore, **Petri marked net** is an ordered four [9, 11]:

$$N = \left(P, T, A, M_0\right) \tag{1}$$

where the following conditions are met:
(1) $N = (P, T, A)$ is the net,
(2) $M_0 : P \to \mathbb{Z}_+$ it is a function defined on the set of places called initial marking of the N net.

In this class of net, places have unlimited capacity for tokens, but arcs can carry only one token. Execution of transition involves the elimination of individual tokens from the input places and adding the individual tokens to the exit places of this performed transition. In the class of **generalised net** [9, 11] it is possible to eliminate or add to one place more tokens in the performance of a single transition by assigning appropriate weights arcs.

By introducing function of limiting capacity of the places for tokens to the generalised net, one of the most common net class in literature [11], called **net of places and transitions** is obtained [9, 11]. This net is an ordered six:

$$N = \left(P, T, A, K, W, M_0\right) \tag{2}$$

where the following conditions are met:
(1) $N = (P, T, A)$ is the net,
(2) $K : P \to \mathbb{N} \cup \{\infty\}$ is a function called **places capacity**, specifying for each place the maximum number of tokens, which place may contain,
(3) $W : A \to \mathbb{N}$ is a function of the arc's weight,
(4) $M_0 : P$ ® $\mathbb{Z}_+$ is a function defined on the set places, called **initial marking** of N net.

The net of places and transitions can easily replace a generalised net, which has the same properties. Accordingly, the term net of places and transitions often refers to generalised net in the literature [9, 11]. The Petri net, as a bipartite graph, can have a hierarchical structure [1, 11], greatly facilitating the modelling of complex systems. Constructing a hierarchical net can be implemented in two ways. The first, from the particular to the general – bottom – up, involves the extraction of separate, small subsystems from the model system and then building simple nets. Then, these simple nets are deposited in growing parts until a whole net of a modelled system is given. The second way, from the general to the particular – a top down, is to build a net for the entire system, without a detailed analysis of the fragments and treating the individual parts of the net in a similar way etc. This procedure is called structure modelling (hierarchical modelling). The net built using one of these methods is called a **hierarchical net** and the net elements of the composite structure are called hierarchical net elements [1, 11]. These elements representing parts of the net are called macroplaces and macrotransitions. The choice of modelling depends largely on the type of system, which is being modelled. If it is completely unknown to the designer of the system, using a second method is more efficient. On the other hand, when past experience can be used while modelling the system, the first method becomes more convenient.

2.2. Simple time Petri nets

In 1983, the first extension of the definition of Petri nets with bounded time was successfully introduced. One such extension is a combination of bounded time with transitions. Thus, extension of the marked net lead to the definition of **simple time net**, and the extension of generalised net to the definition of the interval time net [3, 11]. A simple time net is an ordered five:

$$N = (P, T, A, M_0, \sigma) \tag{3}$$

where the following conditions are met:
(1) $N = (P, T, A)$ is the net,
(2) $M_0 : P \to Z_+$ it is the initial marking of N net,
(3) $\sigma : T \to Q_+$ is a delays function, assigning each transition of the net to a non-negative rational number $\sigma(t)$, called a static delay of transition t.

Compared with the marked net [11], the definition of the simple time net has been enriched with the delay function $\sigma(t)$, defined on the set transitions. This delay is a non-negative rational number specifying the execution time of the transition, counted from the time at which it was activated. If the transition t is enabled (i.e. all the input places have tokens) that it must be done after exactly $\sigma(t)$ units of time. This delay is a static delay [11], i.e. independent of the global clock. In contrast, a dynamic delay is defined for any of the transition and is dependent on the current value of the clock. Therefore, if the data transition is active, dynamic delay determines the amount of time remaining to complete this transition.

The state of simple time Petri net [11] is a pair (M, δ), wherein M is a marking, and δ is a vector of dynamic delays.

The transition $t \in T$ is enforceable in state (M, δ) [11], if it is enabled by marking M and has minimal dynamic delay among all active transitions at the same marking. If the transition is not active when the marking is, it is written in the $\delta(t)$ =_ form. If the transition t is enforceable in the state (M, δ) and as a result of its execution is replaced by a new state (M', δ′), such an operation can be written as [11]:

$$(M, \delta) \xrightarrow{t} (M', \delta') \tag{4}$$

Given a string of transitions α = t_1, t_2, …, t_n conducted from the state (M_i, δ_i), such that [11]:

$$(M_1,\delta_1)\overset{t1}{\rightarrow}(M_2,\delta_2)\overset{t2}{\rightarrow}\ldots\overset{tn}{\rightarrow}(M_{n+1},\delta_{n+1}) \tag{5}$$

this time performance of strig α is called the sum of dynamic transitions t_i in the states (M_i, δ_i), which is denoted by δ(α) [11], ie.:

$$\delta(\alpha)=\sum_{i=1}^{n}\delta_i(t_i) \tag{6}$$

3. Steps of synthesis of the digital controller of railway crossing devices

A controller of railway traffic control systems implemented on digital platforms can be classified into class of real-time discrete measurement - control systems (from the point of view of automation and real-time control systems). Systems of this type are reactive systems, which require an assumption about the immediate response to events as they happen [1, 3]. Designing such systems requires a behavioural approach to modelling of the system. Such modelling is enabled by efficient tools based on graphs, such as Petri nets or state diagrams [15]. An important advantage of these tools is the ability to manipulate the concept of the state, in order to support the construction of hierarchical structures and concurrent modelling.

Figure 1 shows the steps in the synthesis of discrete controller of railway crossing devices. Based on the verbal description, a functional model of the controller using the Petri

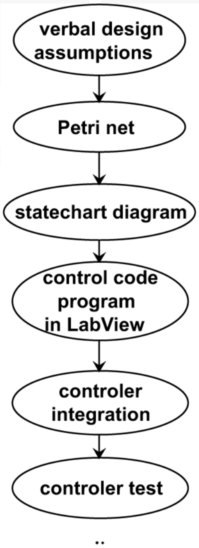


Fig. 1. Steps of synthesis of the digital controller of automatic crossing devices

net is developed. On the basis of this net and the use of the statechart tools from LabVIEW package implements a graphical diagram of a controller's state machine. Statecharts tool´s functions allow the introduction of the program code to the structures obtained in diagram states. LabVIEW generates the code control program with the as-prepared diagram. The next step of the realisation of the controller is integrating the software with the hardware platform. After integration and a positive run of the controller, its tests and verification in accordance with the previously developed program of tests and trials are carried out.

3.1. Warning users of roads

Warning road users is done from the moment of entering of front of rail vehicle on the approach section, until the moment when the whole rail vehicle crosses the danger zone. This process consists of the following phases [7, 8]:

the process of closing the crossing:

- rail vehicle approaching the crossing, crashing into sensors track approach section, will launch controller of MCD system, resulting in the inclusion of red light on the road signals and turning on sirens sound signal,
- after 8 seconds´ delay, electric drives that leave the bars dams are activated,
- the deviation of bars dams from the vertical position results in turning on the lights of lighthouse on bars,
- positive verification: closure of the dams in the horizontal position, shining lights on the road signalling and lighthouse on bars will turn on the signal OSP2 on the signalling ToP,
- if during the warning process a rail vehicle on the second track is detected, the closure process will be continued.
-

the process of opening of the crossing:

- after max. 6 seconds from the exit of the rail vehicle from the sensor track of the crossing danger zone, lights on the road signalling are switched off and lifting of bars dams begins,
- turning off light lanterns on the drams takes place when they reach the vertical position,
- a change is also performed to the signal to the ToP signalling,
- positive verification: the state of dams in the vertical position, turning off the lights on the road signalling and lighthouse on drams, causes turning on the signal OSP2 on the signalling ToP.

From the point of view of ensuring the safety of road users, starting the warning process and closing the crossing must be performed with the appropriate lead time. This time is defined as the pre-warning time [2]. When calculating this time, it is assumed that the rail vehicle is running at the maximum permissible speed of force on the route of the railway and road vehicles leaving the danger zone passing move at the minimum speed. In [2], it is assumed that the pre-warning time t_0 is:

$$t_0 = t_n + t_{zp} + t_{0p} \tag{7}$$

where:

t_n [s] – the time classes of danger zone of railway crossing,

t_{0p} [s] – the delay time to be paid by theMCD controller for electronic devices shall be 1 sec,

t_{zp} [s] – the constant value of inventories time shall be 10 sec.

This t_0 time for the station`s conditions, according to the decree of the Ministry of Communications [14], cannot be less than 30 s and because of practical reasons, no more than 90 seconds. Thus, the time should be included in the range:

$$30 \text{ s} < t_0 .< 90 \text{ s} \qquad (8)$$

The general structure of the block diagram under consideration and the automatic railway crossing devices are shown in Fig. 2.

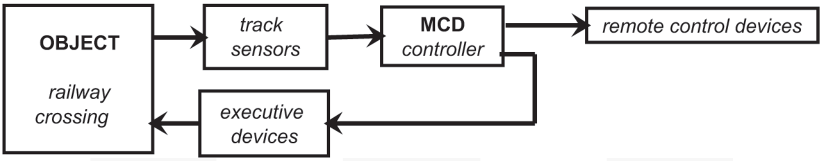


Fig. 2. The block diagram of automatic railway crossing devices

4. Marked Petri net of automation crossing devices

On the basis of the rules of operating an MCD controller of automatic railway crossing devices presented in Fig. 3, a marked Petri net of these devices was developed [5, 6, 7]. In this net, it is assumed that the places comply with the conditions placed after the word "**IF ...**" and transition activities (shares) followed by the word "**Then...**". Using the principle of reduction of Petri nets [9] has led to the development of reduced net for automatic railway crossing devices, which is shown in Fig. 4. The identified places are:

(1) **if** *the rail vehicle is on the approach section* **then** *warn.*
(2) **If** *warn* **then (***turn on the road signalling device*) and (*measure delay of 8 seconds*).
(3) **If** *measured delay of 8 seconds* **then** (*turn on the siren signal*) and (*turn on the electric drive of rods*).
(4) **If** *rods are deviation from the vertical* **then** *turn on the rod's electric torch*.
(5) **If** (*turn on the road signalling device*)
and **(***turn on the siren signal*) **then** *view signal Osp 2 on Top signalling* device.
and (*rod's turnpike closed*)
and (*turn on the rod's electric torch*)

(6) **if** *the rail vehicle drove into the danger zone* **then** *measure delay of 8 seconds*
(7) **if** *the rail vehicle lives the danger zone* **then** *measure delay of 6 seconds*
(8) **If** *measured delay of 6 seconds* **then** (*turn on the electric drive of rods*)
and (*turn off the road signalling device*)
(9) **If** *rods are in the vertical* **then** *turn off the rod's electric torch .*
(10) **If (***turn off the road signalling device*)
and (*turn off the siren signal*) **then** *view signal Osp 1 on Top signalling* device.
and (*turn off the rod's electric torch*)
and (*rod's turnp ike opened*)
(11) **if** *viewed signal Osp 1 on Top signalling* ***then*** *the railway crossing is opened.*

Fig. 3. The rules of operating an MCD controller of automatic railway crossing devices

- P1 – end of one of the sub processes: closing, holding up, lifting, control or decision and transition to information receipt event (if there is one),
- P2 – end of information receipt,
- P3 – end of information transmission,
- P4, P5 – conditions allowing control of approach section states and transmission of information on the state.

It can be seen that places P4 and P5 are token generators; places P1 and P2 are token sinks, place P3 transfers tokens from information processes (places P4 and P5) to the decision – making and executive processes (places P1 and P2), where they are absorbed. This means that each piece of information on the approach sections state reaches the decision-making and execution processes and then interpreted and executed. The graph of attainable marking – Fig. 4 – allows a statement that the net is lively, active, stable, 3-side constrained, safe and conservative – with one exception, when followed by the transfer tokens with the information process to other processes. The illustrated net, as a result of the analysis, shows that, during dynamic operation (the graph of attainable marking), there is no pinching it, which means that the system will pass freely to each of the states.

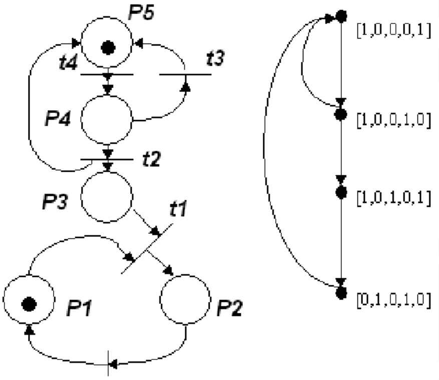
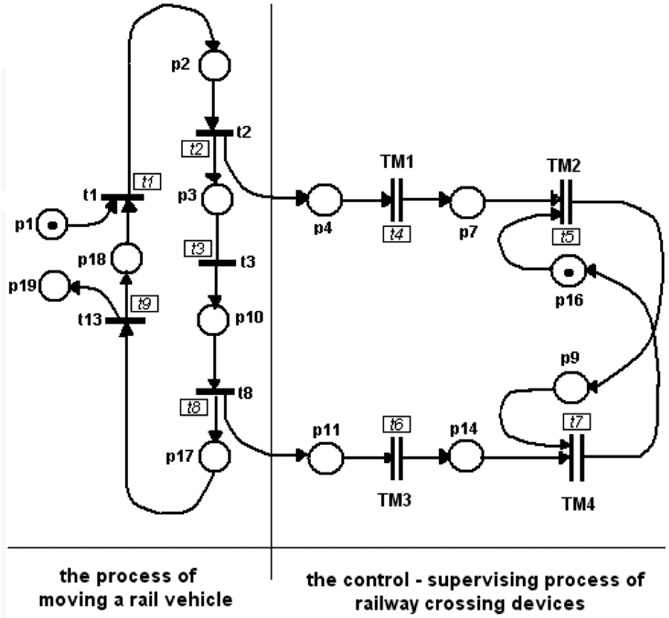


Fig. 4. Reduced Petri net and its graph of attainable marking

5. Simple time Petri net of automation crossing devices

Modelling of the dynamic behaviour of a discrete event system requires the use of tools for modelling time constraints. One class of Petri nets allowing such modelling is the class of simple net time, which is discussed briefly in p. 2.2. of this work. Using the knowledge and experience of the development of the reduced net, automatic railway crossing devices, using the methodology of the construction of hierarchical nets – from the particular to the general (bottom – up) – a simple time net for these devices is developed. This net (Fig. 5) highlights two fundamental processes: the process of moving the rail vehicle and the control – supervising process of automation railway crossing devices.

The process of movement of the rail vehicle corresponds to the place P1, P2, P3 and P10, P17, P18 of the simple net time (Fig. 5), which is associated with the entry of the vehicle to the approaching section from trial, entry and exit from the danger zone of crossing and travel and leaving trail. Places P4, P16 and P9, P11 and P14 correspond to the control - executive process of an automation crossing devices. Macrotransitions TM11 and TM13 model subprocesses of the acquisition of measurement signals from the sensor's track section approaching and passing the danger zone. Macrotransition TM12 is responsible for closing the passage sub-process and macrotransition TM14 for sub-process of opening run. For transition and macrotransition of this simple net, appropriate static times are assigned. After these times, transition and macrotransitions are performed. These times are modelling delay`s times associated with the processing of signals and data.



Places:
p1 – away from the station of rail vehicle on the track,
p2 – the rail vehicle is before approaching section,
p3 – the rail vehicle is on approaching section,
p4 – the rail vehicle is on approaching section,
p7 – message "close" crossing devices,
p9 – status of crossing devices is "close",
p10 – the rail vehicle is before the danger zone of crossig devices,
p11 – the rail vehicle exit from the danger zone of crossing devices,
p14 – message "open" crossing devices,
p16 – status of crossing devices is "open",
p17 – the rail vehicle on the track,
p18 – track is free (no rail vehicle on the track),
p19 – entry rail vehicle from the track to the station.

Transition:
t1 – ride a rail vehicle by track to approaching section,
t2 – moving the front of rail vehicle approaching section,
t3 – drove a rail vehicle through a approaching section,
t8 – drove a rail vehicle through the danger zone of crossig devices,
t13 – leaving the track by a rail vehicle.

Microtransition:
TM11 – detected by measuring system entry rail vehicle to the approaching section,
TM12 – making process of closing,
TM13 – detected by measuring system exit rail vehiclefrom the danger zone of crossing,
TM14 – making process of opening.

Fig. 5. Simple time Petri net of automation crossing devices

6. State machine diagram of the controller of automation railway crossing devices

State machine diagrams have been proposed by D. Harel as a visual formalism to describe complex reactive systems [10]. This methodology is a notation that uses the concept of the state and is an extension of the concept of finite automaton. A graphical representation of this methodology is a diagram of states and transitions. In the basic diagram, there is a possibility of assigning complex states of subordinate diagrams. Orthogonality in the diagram means that two or more sequential machines can model concurrent processes. In turn, a broadcast mechanism is implemented by generating events that by feeding back affect functioning of the diagram [16]. The statechart module of graphical programming environment – LabVIEW, is a tool for creating diagrams of state machines. The features of this module include, for instance, defining states and transitions, events, hierarchy, and substates. With the introduction of the corresponding program code to set up the structure diagram after translation and compilation, the program code for the designed controller is obtained. The state diagram, in accordance with a net shown in Fig. 5, takes the following four states:

- waiting,
- closing,
- maintenance,
- opening.

The main loop of the controller program in LabView applications (Fig. 6b) through block **Run Statechart**, starts the state machine diagram of the controller (Fig. 6a). This loop also calculates the execution time of one iteration of the loop with an accuracy of 1 ms.

Figure 7 is an exemplary block diagram of the initialisation of system input signals' (sensor signals) acquisition.

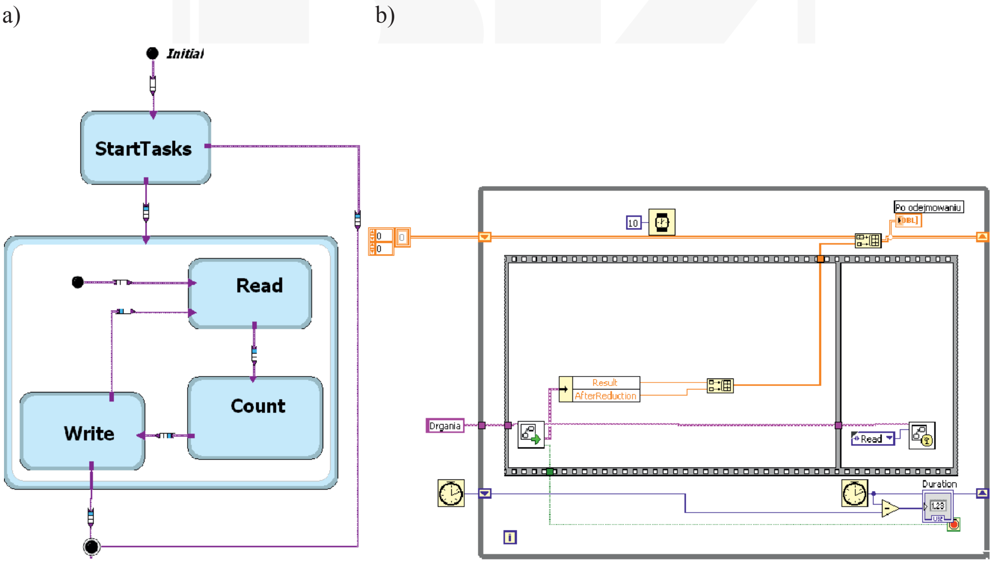


Fig. 6. a) The state diagram controller, b) The main program loop

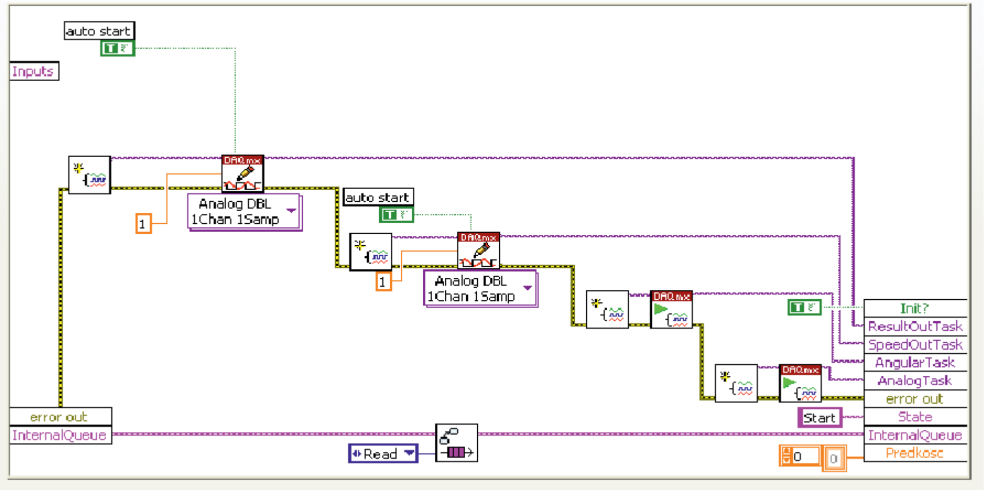


Fig. 7. View of the Block diagram with the settings of the signal acquisition system

7. Conclusions

Using Petri nets for modelling of discrete systems, one can dynamically model (in time) the behaviour of the system. The simple time Petri net for automation railway crossing devices presented in this work makes it possible to calculate the response time process control and - a control device for driving into of an automation tailway crossing devices front section of the vehicle to the approach. Also, based on this net, a state machine diagram of automation railway crossing devices Modern LabView graphical programming environment gives Statechart tool lets through practical application made state machine. So made software is clear and easy to maintain. National Instruments, whose product is the LabView environment, is also a manufacturer of digital hardware platforms for industrial applications. Implementing a comprehensive device driver spp is therefore possible.

References

[1] David R., Alla H., *Discrete, Continuons, and Hybrid Petri Nets*, Springer 2005.

[2] Dąbrowa-Bajon M., *Podstawy sterowania ruchem kolejowym*, Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa 2002.

[3] Hruz B., Zhou M.C., *Modeling and Control of Discrete-event Dynamic Systems with Petri Nets and other tool*, Springer 2007.

[4] Lewiński A., Perzyński T., *Nowe rozwiązania komputerów sterujących w systemach sterowania ruchem kolejowym na przykładzie SSP*, Materiały konferencyjne „Transport w XXI wieku", Warszawa 2001.

[5] Mielnik R., *Petrii net as a tool for modelling a microprocessor measurement-control system using in critical applications*, Proceedings of the IMEKO-TC7 Symposium, "Measurement Science of the Information Era", Kraków 2002.

[6] Mielnik R., *Sieć Petriego w syntezie mikroprocesorowej sygnalizacji przejazdowe*, Materiały konferencji SEMTRAK, Zakopane 2002.

[7] Mielnik R., *Modelowanie przepływu informacji w urządzeniach sterowania ruchem kolejowym*, Materiały konferencji SEMTRAK, Zakopane 2008.

[8] Mielnik R., *Sieć Petriego w projektowaniu systemów pomiarowo-sterujących zdarzeń dyskretnych w zastosowaniach krytycznych*, [in:] *Elektrotechnika w zastosowaniach trakcyjnych*, edited by A.S. Jagiełło, Monografia No. 450, Wydawnictwo Politechniki Krakowskiej, Kraków 2014.

[9] Starke P.H., Żurek J., *Sieci Petrii*, Warszawa 1987.

[10] Szmuc T., Szpyrka M., *Metody formalne w inżynierii oprogramowania systemów czasu rzeczywistego*, WNT, Warszawa 2010.

[11] Szpyrka M., *Sieci Petriego w modelowaniu i analizie systemów współbieżnych*, WNT, Warszawa 2008.

[12] Winiecki W., *Organizacja komputerowych systemów pomiarowych*, Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa 2006.

[13] *Instrukcja o budowie, przeglądach i konserwacji urządzeń zabezpieczenia ruchu na przejazdach kolejowych E-4*, Warszawa 1996.

[14] Rozporządzenie Ministra Transportu i Gospodarki Morskiej z dnia 26 lutego 1996 r. w sprawie warunków technicznych, jakimi powinny odpowiadać skrzyżowania linii kolejowych z drogami publicznymi i ich usytuowanie, Dz. U. Nr 33, poz. 144.

[15] Hennie F. C., *Finite-State Models for Logical Machines*, Wiley, New York 1968.

[16] *LabVIEW Statechart Module Tutorial*, http://www.ni.com/tutorial/7425/en [online: 9.08.2014].