Sławomir JASZCZAK[1], Krzysztof MAŁECKI[1]

West Pomeranian University of Technology (1)

Hardware and software synthesis of exemplary crossroads in a modular programmable controller

***Abstract**. The aim of this article is to present the achievements in the field of the execution of a road junction model as well as the design of traffic managing algorithms and programs based on a modular programmable controller (S7 – 200 CPU 222), i.e. under conditions of restricted program and data memory along with limited syntax of the programming language. The authors have shown that it is possible to produce a simulation environment (of a structure similar to real conditions), which will allow for experiments with the algorithms controlling traffic lights and observation of vehicles behaviour in the newly adjusted environment.*

***Streszczenie.** Celem niniejszego artykułu jest przedstawienie dokonań w zakresie wykonania makiety skrzyżowania dróg oraz opracowania algorytmów I programów dla potrzeb sterowania ruchem na bazie programowalnego sterownika modułowego (S7 – 200 CPU 222) tj. w warunkach ograniczonej pamięci programu i danych oraz ograniczonej syntaktyki języka programowania. Autorzy wykazali, że możliwe jest opracowanie środowiska symulacyjnego (o strukturze podobnej do rzeczywistych warunków), co pozwoli na prowadzenie badań eksperymentalnych dla algorytmów sterujących sygnalizacją świetlną i obserwacji zachowań pojazdów w nowo wysterowanej rzeczywistości. (**Synteza sprzętowo-programowa przykładowego węzła drogowego w programowalnym sterowniku modułowym**).*

Keywords: hardware and software synthesis, PLC, traffic.
Słowa kluczowe: synteza sprzętowo-programowa, PLC, ruch drogowy.

Introduction

Analysis and modelling of traffic is a difficult task, what with the complexity of the problem and its stochastic nature. Road administration and scientists, not wanting to expose drivers to inconveniences connected with attempts of controlling an area (e.g. a crossroads or a couple of intersections in close vicinity), make use of various simulators. Over the last few years, many publications have covered the attainments in this field. In general, they can be categorised as software, hardware or hardware and software solutions. The most common are probably software solutions, e.g. complex systems, such as MATSim, VISSIM [7], TRANSIMS [18], MITSIM, AIMSUN, SUMO [11] and CORSIM, as well as theories [5, 17], smaller simulations: [2, 3, 13] or those regarding pollution [24]. Some authors focus on solutions close to real-time simulation [8]. As far as hardware is concerned, a wide variety can be observed. Multiple models representing certain road junctions have been created, such as [6], [9], [12, 19], hardware-in-the loop systems [4, 25], real object scale models [14] and GPU utilising solutions [21].

This article presents the achievements in controlling a micro scale real object.

Idiosyncrasy of modular controllers

In automation of processes and large scale machines (i.e. over several dozen points), modular controllers, e.g. S7-200, VersaMax Micro, etc. can function as devices supporting larger scale controllers. This task usually entails logical data acquisition as well as analysis and sending information in direct or processed form to superior controllers. Therefore, they do not exert a direct influence on a control system. Moreover, in case of controlling applications designed for use in the industry, the limitations of micro controllers need to be considered. Those stem directly from the syntax of programming languages of PLC controllers, usually LD (ladder diagram) and/or IL (instruction list) together with limited data and program memory. To reduce the importance of this problem, there are used the optimization methods of logic structures for control network [1, 10, 16], including using graph theory and Petri nets.

However, it turns out that despite the restricted configuration possibilities (program and data memory, additional extending modules, communication interfaces, etc.), the controllers can be utilised as traffic directing devices under laboratory conditions, i.e. as a function of devices verifying real time traffic controlling algorithms verifying devices.

Stages of hardware and software synthesis

The methodics of control system designing describe a process beginning with specifying the need for developing a new system and ending with the creation of a product of appropriate quality. Any control system includes both software and hardware elements. Most of the hardware components can be configured basing on standard elements and only a minority of specialist devices has to be designed separately. Application software usually needs to be drawn up individually, although it can be made up of standard program modules, e.g. a communication module, or classes and functions used multiple times.

The final product that is the finished controlling program along with device plans, is created through the analysis of the field of use of the device, specification of user's requirements, documentation of the course of the project, hardware, software and usage regulations structure documentation and testing of the components. A cyclic development model of the control system, including the division into stages is depicted in fig. 1.

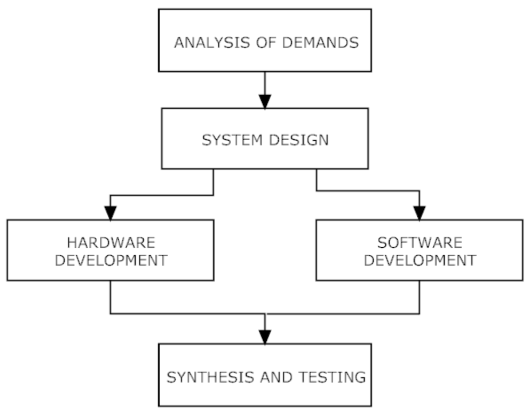

Fig. 1. A cyclic development model of the control system [20]

Requirements analysis (Fig. 1) includes the field of use as well as determination and documentation of the aim of the development of a control system. The result is a system requirements specification, i.e. a document defining the functions of the system, operation accuracy, speed and reliability, applied hardware limitations, project cost estimate and expected effects of its implementation. It needs to be stressed that the analysis must not impose a method of system development – it should only specify its aims and purpose. Before the end of this stage, the specification is assessed by implementation experts. After approval, the specification becomes the basis for a system design planning and assessment. Credible specification assessment reduces the risk of the system's failure to fulfil future user expectations.

During the *system design* stage, the architecture is decided upon by dividing the system into components of purpose stated in the specification and assigning the functions of hardware and software. This stage results in the creation of the following documents:

- system design – specifying system decomposition method and functions of particular components;
- initial specification requirements regarding system components;
- test plan – defining test schedule, required resources and project members' responsibilities;
- software development plan – describing work schedule and cost.

Before the beginning of the next stage, the documents are assessed during the review of system design. When approved, they become the basis for independent elaboration of system components.

Component design includes independent and concurrent development of hardware and software components described in the previous stage. Each component needs to be specified, designed, manufactured and tested individually. The result is a set of fully documented system components – they are evaluated in two revisions:

- functional – confirming all requirements mentioned in the system specification have been met;
- physical – ensuring all components have been delivered with complete documentation.

System integration and testing – a stage in which separately created components are combined and tested – according to the test plan. The results of finishing this stage are a working system and test reports, which, along with corrected documentation undergo a qualifying review checking the completeness and adequacy of conducted tests and deciding on forwarding the system to final evaluation.

Appreciation and acceptance – the final stage of system development in which fulfilment of all requirement described in system specification is checked. It should be done by a special testing team independent from the project team. This stage can also include test exploitation in the system's target environment.

Example of synthesis of an exemplary crossroads

The task of the developed control system is sequential control of traffic lights of a crossroads model of an assumed scale, according to the declared needs.

The system is meant to be used as a teaching aid in laboratory classes and can be the basis for discussing the methodics of design and implementation of traffic lights controlling algorithms and the rules of coupling actors and sensors with programmable controllers. Basing on the initial assumptions, the following functions of the system have been defined:

- checking initial condition necessary for the correct work of the device,
- communication with the operator – receiving commands,
- signalling states of phases,
- signalling alarm states.

The aforementioned functions have been depicted in fig.2.

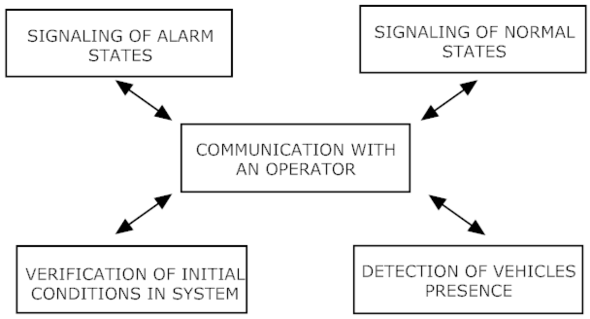

Fig. 2. A functional structure of the control system [own research]

As seen in the diagram (fig. 2), the operator has an ability to interfere with each of the functions, although he does not take direct part in the traffic lights controlling process itself.

Detailed functions of the control system:

system activation - conditions necessary to switch the system on:

- properly working control system (presence of power supply, controller in RUN mode, controller program running),

communication with the operator (operator's actions):

- system activation – powering on sensors and actors,
- lights activation – starts the traffic lights,
- choice of traffic lights mode – colour mode or safe mode,
- warning cancellation – transition from safe to colour mode,
- vehicle detectors activation on set roads.

signalling process state:

- system ready – awaiting traffic lights activation,
- system in colour and safe mode,
- signalling individual phases, i.e. green, yellow and red lights in all controlled directions,
- alert – process stopped – takes place if forbidden states appear, e.g. green phase at the same time in conflicting directions.

Device design begins with breaking down a control system into functional modules, presented in fig. 3.

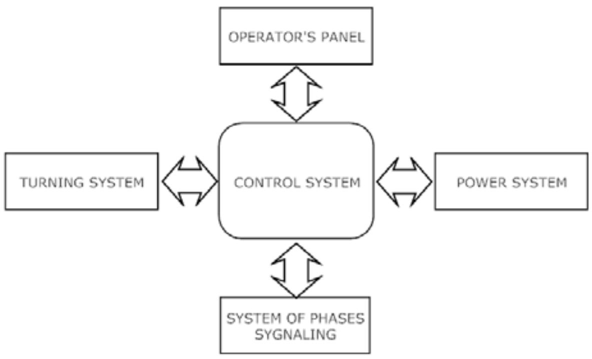

Fig. 3. A functional modules of the traffic lights control system [own research]

Construction of the *control system* (Fig. 3) was based on a modular controller which will be the execution platform of the control software. After balancing all inputs and outputs of system components, it turned out the controller needs at least 8 binary inputs and 14 binary outputs. Because of the type of switching signals, the outputs need to be of transistor type.

System activation circuit – powers on sensors (buttons and switches) and signalling elements (LEDs). Comprises a button and a LED:

- output: „System załączony” LED - 1 bit
- input: „Aktywacja systemu” button - 1 bit

Vehicle detection circuit – its aim is to provide information regarding vehicle’s presence at one of the intersection entrances. The function of presence detectors is fulfilled by buttons:

- input: presence detector - 4 bits

Phase signalling circuit is meant to present the state of current phase in specified traffic directions. It is made up of LEDs controlled with control system outputs – in case of green lights showing up in conflicted directions, the traffic lights are stopped and an alert is displayed.

- output: LED lights control - 12 bits

Operator panel – included buttons and switches for mode selection as well as signal lights informing about the state of process and the mode. The following operations are to be possible to be executed with the use of the panel:

System activation – manner of operation, inputs and outputs described in the System activation circuit paragraph.

Traffic lights activation and stopping

- output: „Traffic lights active” LED - 1 bit
- input: „Start of traffic lights” button - 1 bit

Operating mode selection – colour, safe mode

- output: „The colour mode” LED - 1 bit
  „The safe mode” LED - 1 bit
- input: „The colour mode” button - 1 bit
  „The safe mode” switch - 1 bit

Alert state handling

- output: „Alarm” LED - 1 bit
- input: „Alarm reset” button - 1 bit

Power supply – should provide the controller and signalling elements with standard voltage 24VDC. Current efficiency needs to be greater than the sum of currents of all devices powered by the supply.

Table 1. A power consumption analysis of elements installed in the control system [own research]

Device	Quantity	Sum of currents
S7-224 controller	1	280mA
EM223 extending module	1	125mA
Relay	1	60mA
LEDs	16	320 mA
Sum of currents		**785 mA**

Basing on the analysis, the maximal current consumption has been found to amount to 785 mA (tab.1). However, because of the fact all of the output devices are never on at the same time, the actual current during the operation of the device is significantly lower. By reason of safety, though, it has been assumed that the power supply needs current efficiency of 2.5A.

A real crossroads model, created accordingly to the presumptions is presented in fig. 4.

For this article, two fixed-time algorithms and two algorithms of a fixed initial state – red light and movement detection - have been implemented. Due to the limited number of controller inputs, one fixed-time algorithm displaying the same signals on opposite indicators. Another algorithm displays a signal from the basic sequence (red –> red and yellow –> green –> yellow –> red) on only one indicator, whereas the others are showing red lights at the time. The order of the light basic sequence changes clockwise.

In nighttime simulating algorithms, all indicators are set to display red lights. The change of light ensues after detection of vehicles at specific entrances. For this case, two algorithms have been designed: the first one changes the signal for the vehicle’s entrance and its opposite; the other flashes green only for the entrance of vehicle detection, the other three lights remain red.

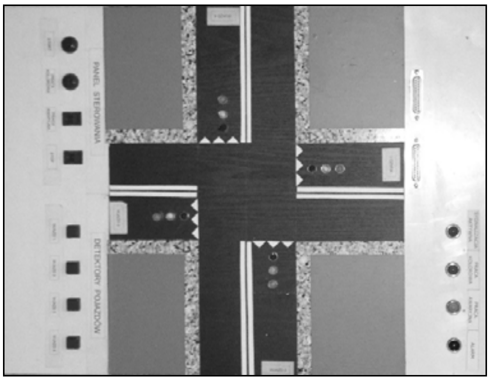

Fig. 4. A physical model of the real crossroads [own research]

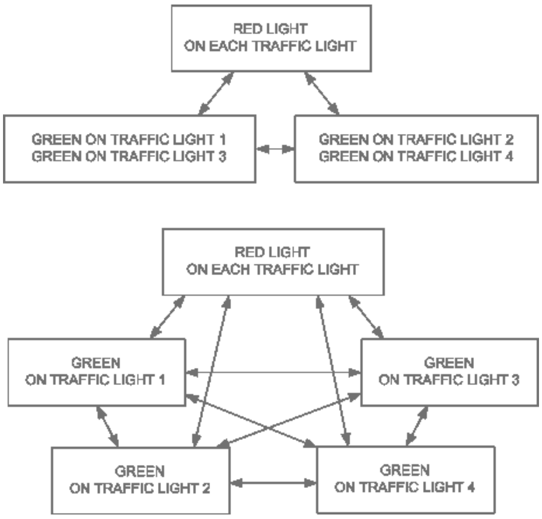

Fig. 5. System lights on traffic lights [own research]

There is one more possible initial state. Two opposite lights are green (main road). The remaining lights, located by the side roads, show red signal. After the arrival of a vehicle, green light is displayed at its entrance or both side roads. The main road indicators remain red then.

The authors have prepared various solutions which should facilitate the observation of differences in traffic control with the use of distinct algorithms.

In the picture 6 a general diagram representing the operation of a fixed-time algorithm – the more important elements have been marked with numbers:

1. after switching the controller on and starting the program with the „START” button (I0.0), yellow lights blink at all entrances. This is an information for drivers, meaning the traffic lights are about to switch into colour mode. Next, red lights are displayed at all entrances (safe traffic stop). At this point safe mode can be induced or the controller can be switched off.
2. controller power off sequence, safe for the traffic. First, red lights are shown at all entrances to stop the traffic, next, they are replaced by yellow blinking lights. From that moment the traffic is managed with the use of road signs. This sequence is executed with the „STOP” (I0.3) switch.
3. safe mode activation, takes place after turning on with the „the safe mode” (I0.2) switch. It is possible only when all signals are red.
4. controller power off, takes place after toggling the

„STOP” (I0.3) switch. Possible only when all red lights are displayed. The controller power off sequence, described in point 2 ensues then.
5. another controller power off method is used only when the controller is in safe mode. After pressing the power button, the controller shuts down, skipping the power off sequence.
6. after starting the safe mode, all traffic lights show red signal for an extended period of time, followed by yellow blinking signal.
7. return to colour mode from safe mode ensues after pressing the „the color mode” (I0.1) button.

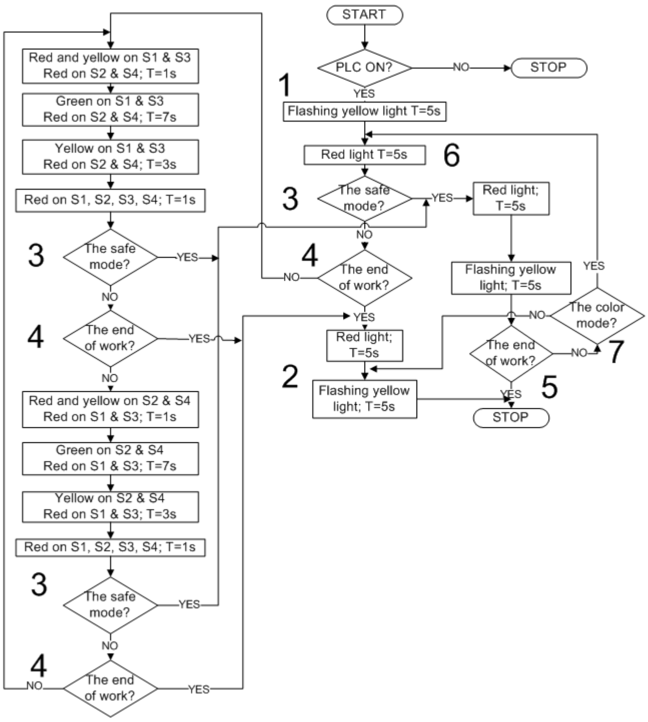

Fig. 6. The PLC operation algorithm for fixed-time mode [own research]

Conclusion

The main aim of this work was to prove that with the use of micro controllers it is possible to prepare a simulation environment of an exemplary crossroads. For this article, the following has been prepared:

- a crossroads model with traffic lights and a set of buttons simulating detectors of vehicles entering the intersection and control panel managing the traffic lights,
- traffic lights controlling programs, created with Siemens Step-7 MicroWIN software.

The model can be used for educational purposes (used controller does not fulfil the criteria of the Minister of Infrastructure regarding traffic lights controllers). It allows for learning about the essence of design and execution of the software controlling traffic lights at intersections.

REFERENCES

[1] Adamski M., Kołopieńczyk M., Mielcarek K., Doskonała sieć Petriego w projektowaniu współbieżnych układów sterujących, *Pomiary Automatyka Kontrola*, (2011), nr 6, s. 656-660
[2] Barcelo J., Codina E., Casas J., Ferrer JL., Microscopic traffic simulation: A tool for the design, analysis and evaluation of intelligent transport systems, *Springer Journal of Intelligent and Robotic Systems*, 41 (2005), Issue 2-3, pp. 173-203
[3] Bloomberg L., Dale J., A Comparison of the VISSIM and CORSIM Traffic Simulation Models On A Congested Network, *Journal of Transportation Research*, (2000)
[4] Bullock D., Urbanik T., Hardware-in-the-loop evaluation of traffic signal systems, *Int. Conf. on Road Transport Information and Control*, (2000), p. 177–181
[5] Chowdhury D., Santen L. and Schadschneider A., Statistical physics of vehicular traffic and some related systems, *Physics Reports*, 329 (2000), pp. 199-329
[6] Dzierżek K., Ostaszewski M., Intersection model with configurable control system, *Zeszyty Naukowe AM w Szczecinie*, (2010), s.36-39
[7] Fellendorf M., VISSIM: A microscopic simulation tool to evaluate actuated signal control including bus priority, in 64th Institute Transportation Engineers (ITE) Annu. Meeting, Dallas, TX, USA, (1994), pp. 1-9
[8] Franzese O., Joshi S., Traffic simulation application to plan real-time distribution routes, *Proc. of the Winter Simulation Conference*, (2002)
[9] Han X., Sun H., The Implementation of Traffic signal Light Controlled by PLC, *Journal of Changchun Institute of Optics and Fine Mechanics*, (2003)
[10] Kołopieńczyk M., Application of adress converter for decreasing memory size of compositional microprogram control unit with code sharing, *Lecture Notes in Control and Computer Science*, Vol. 12 (2008)
[11] Krajzewicz D., Hertkorn G., Rössel C., Wagner P., SUMO (Simulation of Urban MObility) - An open-source traffic simulation, *MESM2002 Proc.*, (2002)
[12] Kwon E., Kim S., Kwon TM., Pseudo real-time evaluation of adaptive traffic control strategies using hardware-in-loop simulation, *IECON'01*, (2001)
[13] Małecki K., Iwan S., Development of Cellular Automata for Simulation of the Crossroads Model with a Traffic Detection System, *Communications in Computer and Information Science, Springer*, no. 0329 (2012), p. 276-283
[14] Małecki K., Jaszczak S., Sokołowski R., Synteza sprzętowa i programowa symulatora sterowania ruchem pojazdów na określonym obszarze, *Pomiary Automatyka Kontrola*, (2012), No. 7, p. 608-610
[15] Miao Q., Zhu F., Lv Y., Cheng C.-J., Chen C. and Qiu X., A game-engine-based platform for modeling and computing of artificial transportation systems, *IEEE Transactions on Intelligent Transportation Systems*, 12 (2011), no. 2, p. 343-353
[16] Mielcarek K., Adamski M., Zając W., Czy prawidłowa sieć Petriego jest siecią doskonałą?, *Metody Informatyki Stosowanej*, (2010), nr 3, s.169-176
[17] Nagel K. and Michael S., A cellular automaton model for freeway traffic, *J. de Physique I*, (1992), vol.2, pp. 2221-2229
[18] Nagel K. and Rickert M., Parallel implementation of the TRANSIMS micro-simulation, *Parallel Computing*, 27 (2001), pp. 1611-1639
[19] Popescu M.C., Ranea C., Grigoriu M., Solutions for Traffic Lights Intersections Control, *Proceedings of the 10th WSEAS*, (2010)
[20] Sacha K.: Projektowanie oprogramowania systemów sterujących, *Oficyna Wyd. Politechniki Warszawskiej*, (1999)
[21] Shen Z., Wang K., Zhu F., Agent-based traffic simulation and traffic signal timing optimization with GPU, *Intelligent Transportation Systems*, (2011)
[22] Strippgen D. and Nagel K., Multi-agent traffic simulation with CUDA, in International *Conference on High Performance Computing & Simulation*, (2009), pp. 106-114
[23] Wang F.-Y., Parallel control and management for intelligent transportation systems: concepts, architectures, and applications, *IEEE Trans. on Intelligent Transportation Systems*, 11 (2010), no. 3, pp. 630-638
[24] Taosheng J., Lixin F., Application of GIS to modified models of vehicle emission dispersion, Elsevier, (2005)
[25] Wells R.B., Fisher J. et al., Hardware and software considerations for implementing hardware-in-the loop traffic simulation, *The 27th Annual Conference of the IEEE, IECON*, 3 (2001), p. 1915–1919

***Authors**: dr inż. Sławomir Jaszczak, Zachodniopomorski Uniwersytet Technologiczny w Szczecinie, ul. Żołnierska 49, 71-210 Szczecin, E-mail: sjaszczak@wi.zut.edu.pl; dr inż. Krzysztof Małecki, Zachodniopomorski Uniwersytet Technologiczny w Szczecinie, ul. Żołnierska 49, 71-210 Szczecin, E-mail: kmalecki@wi.zut.edu.pl.*