The Method of Reducing the Cycle of Programmable Logic Controller (PLC) Vulnerable "to Avalanche of Events"

Andrzej Kwiecień[1] and Marcin Sidzina[2]

[1] Silesian University of Technology, Institute of Informatics
[2] University of Bielsko-Biala, Department of Mechanical Engineering Fundamentals
andrzej.kwiecen@polsl.pl, msidzina@ath.bielsko.pl

Abstract. Reducing the cycle duration of PLC (Programmable Logic Controller) is one of the methods for increasing the access frequency of nodes in the real time distributed system into the transmission medium. The authors presents in this paper one of the methods developed by themselves for a rather specific use of a cases of avalanche of events.

Keywords: PLC, distributed real time system, industrial computer network, time cycle of exchange data, PLC programming, avalanche of events.

1 Introdution

The basic parameter of the design of industrial real time systems is the maximum time (T_G) defined [1] as the maximum amount of time which elapses from the moment of detection of an event or sequence of events, until the appropriate system response to these events. It is obvious, [1] that the shorter time the better properties expressed even by the quality of controlling or monitoring and further reporting of events. Through the analysis of the node work, which is usually the PLC, it is clear (Fig. 1) that the duration of the user application executed by the processor has a huge impact on the frequency of access to the transmission medium.

From the principles of PLC work result (Fig. 1), that values such as T_{IC}, T_{WE}, T_{WY}, T_K are constant. The only one value, variable in time, is the duration of application realization T_{AP}. The sum of all listed values consists of the so-called machine cycle time (time of basic cycle PLC). Therefore, through manipulation of the value of the cycle duration, it is possible to influence on the access frequency of the system node to the communication network coprocessor, and thus gain the frequent access to the medium.

The increase in the frequency of access to the network, influences basically on such parameters as the efficiency of communication system (n) and its practical throughput (P), and thus on the time T_G. [1,2] Therefore, the very high attention is given to the appropriate software development, in accordance with the rules of software engineering, and also new ways and methods designed for reducing

A. Kwiecień, P. Gaj, and P. Stera (Eds.): CN 2011, CCIS 160, pp. 379–385, 2011.
© Springer-Verlag Berlin Heidelberg 2011

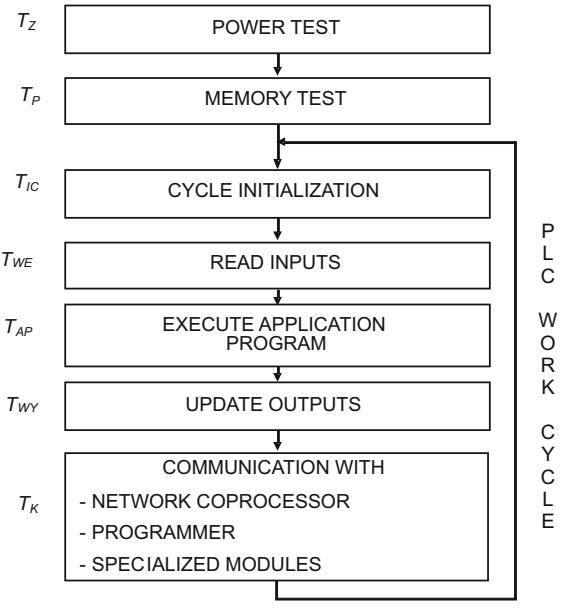

Fig. 1. The basic processes realized during PLC cycle [2]

the duration of the basic cycle of user application are explored. Several methods prepared by the authors have already been published [3,4]. Another one is the subject of consideration in this paper.

2 The Idea of Shortening the Duration of Controller Cycle

The idea of shortening the controller cycle depends on sharing of the application program in such a way, so that in each cycle of program loop (Fig. 1) which is executed in controller (PLC) only a part of it was realized. Thus, a fragmentation (division) of a program into parts is achieved, of which only some are executed in a particular cycle. The realisation of parts in a particular cycle depends on many elements. In general, the division results from the fact whether a particular part (function) is supposed to be executed in each cycle (fixed functions), or whether its execution can be delayed or postponed without any negative effects for the controlled process (reducible functions) . Duration of a machine cycle for the structure of application fragmentation from the (Fig. 2) is described by the dependence (1). The entire execution of cycle is the realization of control algorithms, placed in the fixed and reducible functions, but in the shortened cycle only the algorithms placed in fixed functions are executed,

$$T_{\mathrm{AP}} = T_{\mathrm{S1}} + T_{\mathrm{R}} + T_{\mathrm{S2}} \quad or \quad T_{\mathrm{AP}} = T_{\mathrm{S1}} + T_{\mathrm{S2}} \tag{1}$$

where:

T_{AP} – time for program realization;
T_{S1}, T_{S2} – time for fixed functions realization;
T_{R} – time for reducible functions realization.

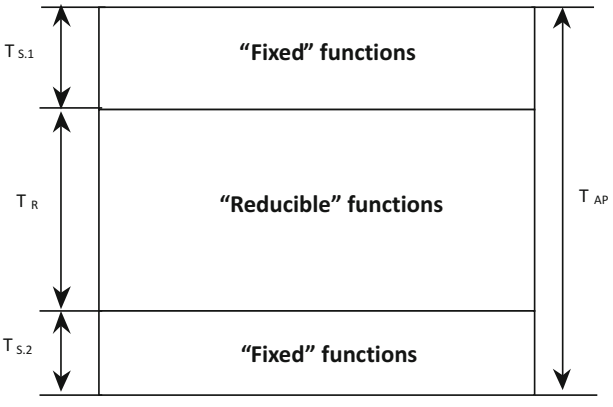

Fig. 2. Fragmentation of the application [2]

The constant part of a program is divided into two segments. The first segment is at the beginning of application and it consists of some functions used for taking a decisions that refer to the shortening the duration of control application, these are counters of exchanges, conditions for finishing the shortened cycle. The shortened cycle or idle cycle refers to the cycle which does not execute any control algorithms placed in the part of reducible functions, that is the application omits the reducing part, and realizes only the part of fixed function. The simple logical issues decide about activating and deactivating of shortened cycle [4]. The way of activation of a shortened cycle can be as follows:

- the cycle of PLC controller is executed;
- at some point an exchange data is required (which can be the result of the action of control algorithm, or this request will be called from outside of the system);
- a request is sent to the network as a result of program response;
- the program sets a state 1 for condition CS a shortened cycle, if the state of this condition is 1 it skips the part of the reduced application;
- after the exchange, the state of variable CS is reset. The execution of control algorithm can be continued.

In the constant part of application the following elements should be included:

- queue of cyclic exchanges;
- queue of acyclic exchanges;

– the condition for the beginning of shortened cycle (if the condition is realised, the cycle is permanently shortened until further notice).

3 The Method Description

One of the rules that prevailed during the development of a method, was the concept of shortening the duration of the controller cycle, while ensuring the proper realization of control algorithm. If it is possible that in the control system many states which trigger the shortening the controller cycle occur, it was then necessary to prevent the starvation of certain parts of the program by introducing the rank order of the transitions. The authors introduced the principle, that an item omitted from the last time is directed at the end of the list of skipped items.

Using a mechanism for temporary cessation of control algorithm realization it is possible to shorten significantly the machine cycle, thus enhancing the frequency of coprocessor access to the memory of central unit. This operation will shorten the duration of individual exchanges in the network.

It should be noticed that the method being the subject of this paper, is characterized by another feature, which accepts dynamic rather than static division of applications. It means, that the range of application executed in a particular cycle depends on the fact whether the network service request appears or not. Obviously, the fixed functions are executed always. Another fundamental feature of this method is that it includes the emergence of an avalanche of events. This phenomenon will certainly cause the delays in realization of reducible functions as a result of increased frequency of transmission request. When many events occur then the segmentation can be presented as in the Fig. 3.

The discussed method work in the following way:

– In case of avalanche of transaction requests, transition to the end of shortening part should be performed in such a way in order to ensure to all segments the execution of control algorithm for a some impassable time;
– To avoid the "starvation" of the control algorithm of some reducible functions, it is necessary, in case of many requests to implement a flag informing that a particular segment of the control algorithm cannot be omitted;
– This mechanism should ensure the execution of each program segment in case of avalanche of events. It will guarantee the increase of minimum frequency of access to the network. $\dot{f}_{\mathrm{ACS}}$

The minimum frequency of the coprocessor access to the network during a complete cycle of the controller is:

$$f_{\mathrm{ACS}} = \frac{1}{T_{\mathrm{A}}} = \frac{1}{T_{\mathrm{APS}} + T_{\mathrm{APR}} + T_{\mathrm{INSTR}}} \tag{2}$$

where:

T_{A} – duration of basic PLC cycle;
T_{APS}, T_{S2} – realization time of fixed functions;

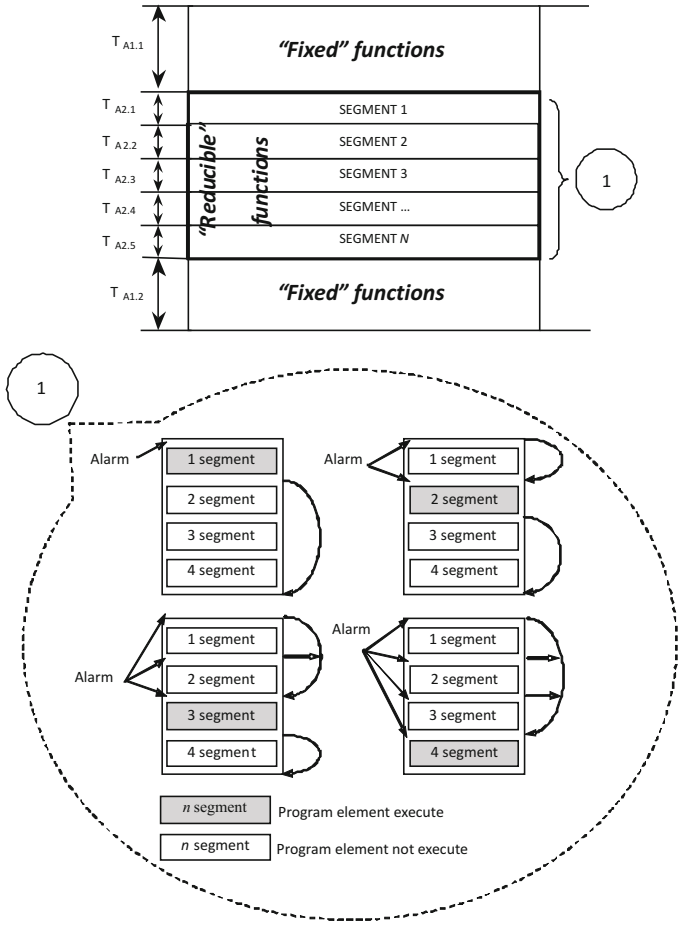

Fig. 3. The mechanism preventing the "starvation" of application segments, in case of many exchange data requests [3]

T_{APR} – realization time of reducible functions;
T_{INSTR} – realization time of all constant cycle elements such a:

- Time of inputs data acquisition – physical states of PLC inputs;
- Time of coprocessor operation;
- Time of updating the physical states of PLC outputs.

In the division of the n segments of program in case of avalanche of events, the average frequency of access to the network is:

$$\dot{f}_{\mathrm{ACS}} = \frac{1}{\frac{T_{\mathrm{APR}}}{n-1} + T_{\mathrm{INSTR}}} \quad . \tag{3}$$

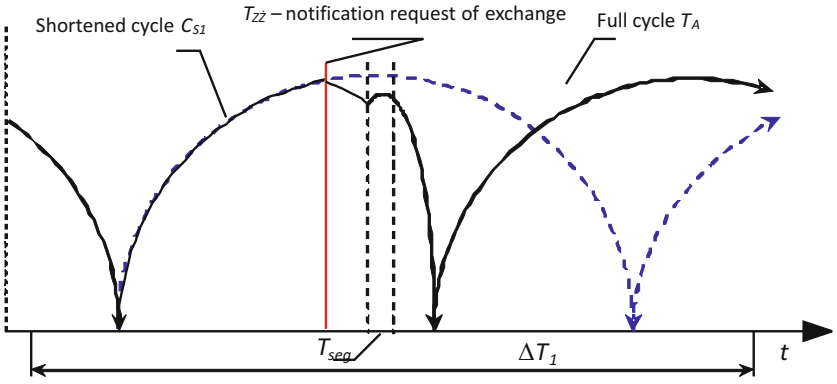

Fig. 4. Illustration of time delay ΔT_1 during the realization of reducible functions for the discussed method

Delay ΔT during the avalanche of events for the entire application program realization is:

$$\Delta T = \dot{T}_{\mathrm{A}} - T_{\mathrm{A}} = n\left(\frac{T_{\mathrm{AP}}}{n} + T_{\mathrm{INSTR}}\right) - T_{\mathrm{AP}} - T_{\mathrm{INSTR}} = (n-1)\,T_{\mathrm{INSTR}} \quad (4)$$

where: n refers to the number of reported events.

This value may deteriorate realization of control process, as well as parameters of real time system. However, it can prevent the starvation of control algorithm through execution of following part of algorithm one after another. Figure 4 presents the time course with forcing the realization of one segment of program.

The method, as already mentioned, is quite specific in comparison with other already presented methods [4]. The main goal of the method is to reduce the duration of data exchange transaction through reduction of basic control cycle (PLC) duration, so that in case of avalanche o events not to cause the control algorithm realization to be blocked. Thus, the essential condition, which must be implemented is the guarantee of continuous realization at least of a part of control algorithm.

4 Conclusions

The presented method requires further research, which the authors are going to conduct in the Genius [5] network. Based on the single tests it can be presume that this method will be effective in shortening the cycle duration, and thus increasing the frequency of access to the transmission medium. The essential issue is that the method presented in the paper allows in a dynamic way, depending on the network load and application to reduce the cycle duration. Thus, the time of realization of a program will be change depending on requirements of control algorithm and the network load. The authors hope that thanks to the presented

method the PLC cycle time reduction on the one hand, improves the parameters of transmission and data exchange, and on the other will not deteriorate the control quality.

References

1. Kwiecień, A.: Analiza przepływu informacji w komputerowych sieciach przemysłowych. Studia Informatica, Gliwice (2002) (in Polish)
2. Kwiecień, A., Sidzina, M.: Metody skracania wymian wyzwalanych w sieciach przemysłowych typu Master-Slave. In: Współczesne Problemy Systemów Czasu Rzeczywistego. WKŁ, Warszawa (2004) (in Polish)
3. Sidzina, M.: Dynamiczne modyfikacje programu aplikacji sterownika swobodnie programowalnego celem zwiększenia częstości wymian komunikatów w przemysłowych systemach rozproszonych czasu rzeczywistego. Doctor thesis, Silesian University of Technology, Gliwice (2008)
4. Kwiecień, A., Sidzina, M.: Metody skracania czasu trwania cyklu sterownika swobodnie programowalnego i ich podstawowe badania. In: Systemy Czasu Rzeczywistego. Projektowanie i aplikacje, vol. 2. WKŁ, Warszawa (2005) (in Polish)
5. Kwiecień, A., Sidzina, M., Rysiński, J.: Analiza wymiany danych w sieci Ethernet dla sterowników swobodnie programowalnych. In: Systemy Informatyczne z Ograniczeniami Czasowymi. WK, Warszawa (2006) (in Polish)