

Programowe testy spełnienia wymagań czasu rzeczywistego na sterowniku PLC z wykorzystaniem języka LD

Krzysztof Oprędkiewicz

Wydział Elektrotechniki, Automatyki, Informatyki i Elektroniki AGH

Streszczenie: W pracy zaprezentowano nową metodę pomiaru uwarunkowań czasowych podczas pracy sterownika PLC, wykorzystującą język drabinkowy. Konstruując algorytm pomiarowy zastosowano reguły interpretacji języka LD. Do pomiarów czasu wykorzystano standardowe timery. Badania doświadczalne przeprowadzono na sterowniku GE FANUC VersMax Micro. Wyniki testów potwierdziły poprawność proponowanej metody.

Słowa kluczowe: sterowanie cyfrowe, sterowniki PLC, wymagania czasu rzeczywistego, język drabinkowy

Uwagi wstępne

System sterowania PLC jest systemem czasu rzeczywistego. W wielu przypadkach przed systemem stawia się wymagania *hard real time*, co oznacza, że muszą być spełnione określone uwarunkowania czasowe, dotyczące przede wszystkim determinizmu i powtarzalności czasów wykonania określonych operacji. Problemy spełnienia wymagań czasu rzeczywistego podczas pracy systemu sterowania cyfrowego są omówione w licznych pracach [1, 11], a spełnienie wymagań czasu rzeczywistego podczas pracy systemu soft PLC pod nadzorem systemu LINUX – w pracy [12].

Zazwyczaj wymagania *hard real time* muszą być spełnione podczas realizacji funkcji o znacznej złożoności obliczeniowej, których czas wykonania w środowisku PLC jest długi, a jednocześnie jest parametrem krytycznym pracy systemu sterowania. Z kolei sprzęt i oprogramowanie zapewniające większą szybkość działania są znacznie droższe od podstawowych wersji sprzętu i oprogramowania (np. cena oprogramowania soft PLC SIEMENS WinAC RTX, zapewniającego spełnienie wymagań czasu rzeczywistego, jest o ok. 50 % wyższa niż wersji podstawowej BASIS).

Można stwierdzić, że oszacowania czasu realizacji określonych operacji obliczeniowych podawane przez producentów sprzętu i oprogramowania często mogą być użyte jedynie jako „górne oszacowanie” dla wyników rzeczywistych. Ponadto wyniki badań [4] pokazują, że symulatory PLC będące częścią niektórych środowisk konfiguracyjnych nie nadają się do modelowania uwarunkowań czasowych pracy rzeczywistego systemu PLC.

Wymienione czynniki uzasadniają stwierdzenie, że najlepszym sposobem postępowania w razie potrzeby sprawdzenia spełnienia wymagań czasu rzeczywistego przez zbudowane oprogramowanie jest wykonanie testów na rzeczywistym systemie sterowania. Jest oczywiste, że testy te powinny być

proste w realizacji i nie wymagać ingerencji w system operacyjny ani sprzęt sterownika.

Należy zauważyć, że nie można *a priori* podać jednej, uniwersalnej metody testowania, możliwej do zastosowania w każdym systemie PLC. Wynika to z dużego zróżnicowania systemów operacyjnych poszczególnych sterowników PLC, co implikuje różne sposoby „dostępu” do zegara czasu rzeczywistego. W niektórych przypadkach (np. SIEMENS SIMATIC S7 300 i 400), najważniejsze parametry związane ze spełnieniem wymagań czasowych podczas pracy systemu są dostępne jako zmienne systemowe [10].

Istotnym czynnikiem, determinującym metodykę wykonania takich testów jest oprogramowanie dostępne w danym systemie PLC. Prosta i skuteczna metoda pomiaru czasu wykonania złożonych procedur obliczeniowych [6, 10] wymaga zastosowania Grafu Sekwencji, który nie jest dostępny na każdej platformie PLC. Jednocześnie można zauważyć, że najpopularniejszym, „klasycznym” narzędziem programowym dostępnym niemal na każdej platformie PLC jest język drabinkowy. Ma on jasną i jednoznaczną interpretację, pozwala na stosunkowo łatwe warunkowe sterowanie wykonaniem poszczególnych części programu.

Z tego względu uzasadnione jest opracowanie metody pomiaru uwarunkowań czasowych podczas pracy systemu PLC bazującej na języku drabinkowym i wykorzystującej jego typowe cechy oraz elementy oprogramowania, takie jak timery.

W pracy zostanie zaprezentowany algorytm pomiaru czasu cyklu oraz czasu wykonania złożonej procedury obliczeniowej, możliwy do realizacji z wykorzystaniem najprostszych narzędzi programistycznych, dostępnych w każdym systemie PLC. W pracy zostanie też zaprezentowana przykładowa funkcja testowa, implementacja algorytmu z użyciem języka drabinkowego, wyniki badań doświadczalnych i uwagi końcowe.

Algorytm pomiaru uwarunkowań czasowych

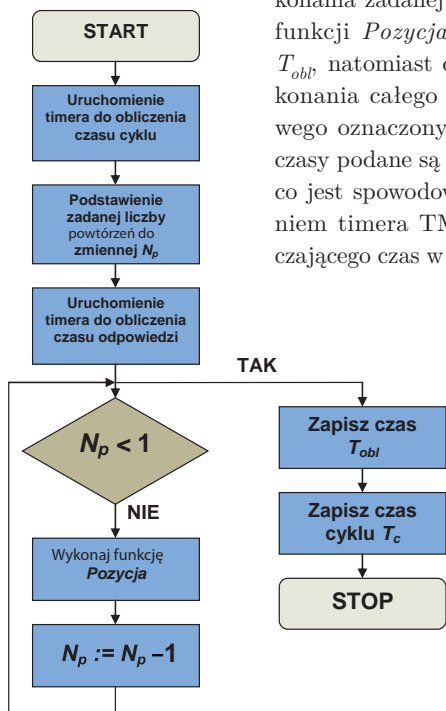
Algorytm pomiaru uwarunkowań czasowych [3] możliwy do implementacji z użyciem języka drabinkowego pokazano na rys. 1. Główna idea jego działania jest analogiczna jak w metodach podanych w pracach [5, 9, 10] i polega na tym, że uwarunkowania czasowe są mierzone podczas realizacji funkcji o znacznej złożoności obliczeniowej, przy czym funkcja ta (w celu dodatkowego zwiększenia obciążenia obliczeniowego CPU sterownika) jest wykonywana w pętli. Na schemacie blokowym (rys. 1) funkcja ta jest oznaczona jako *Pozycja*. W przypadku użycia języka drabinkowego pętla nie może być

zrealizowana, tak jak np. z zastosowaniem języka ST, lecz jest ona realizowana w głównej pętli programowej podczas pracy sterownika, a warunkowe wykonywanie odpowiednich fragmentów algorytmu jest sterowane stykami.

Podczas wykonania algorytmu następuje jednoczesny pomiar czasu całego cyklu oraz czasu wykonania funkcji *Pozycja*. Pomiary są wykonywane z użyciem dwóch timerów. Działanie algorytmu można w uproszczeniu opisać następująco:

- **START**: ustawienie zmiennej *Start* na TRUE,
- **Uruchomienie timera do obliczenia czasu cyklu**: załączenie timera do pomiaru czasu trwania cyklu programu,
- **Podstawienie zadanej liczby powtórzeń do zmiennej N_p** : wartość znajdujaca się w zmiennej *Ile_powtorzen_zadane* jest przepisywana do zmiennej N_p opisujacej liczbę powtórzeń pętli,
- **Uruchomienie timera do obliczenia czasu odpowiedzi**: załączenie timera do pomiaru czasu odpowiedzi programu,
- porównanie $N_p < 1$: sprawdzenie, czy została wykonana zadana liczba wywołań funkcji *Pozycja*, jeśli tak – zapisywane są poszczególne czasy, jeśli nie – kolejne wywołanie funkcji *Pozycja*,
- **Wykonaj funkcję *Pozycja***: wywołana zostaje funkcja *Pozycja* za pomocą operacji CALL,
- $N_p := N_p - 1$: następuje dekrementacja zmiennej N_p ,
- **Zapisz czas obliczania funkcji**: wartość z rejestru timera odpowiedzialnego za pomiar czasu odpowiedzi przeniesiona zostaje do zmiennej T_{obl}
- **Zapisz czas cyklu**: wartość z rejestru timera odpowiedzialnego za pomiar czasu trwania cyklu programu przeniesiona zostaje do zmiennej T_c ,
- etap **STOP**: zmienna *Start* ustawiona zostaje na FALSE.

Podczas wykonania algorytmu, zgodnie ze schematem, wyznaczone zostają dwa czasy, pierwszy z nich to czas wykonania zadanej liczby powtórzeń funkcji *Pozycja* oznaczony jako T_{obl} , natomiast drugi to czas wykonania całego cyklu programowego oznaczony jako T_c . Oba te czasy podane są w milisekundach, co jest spowodowane wykorzystaniem timera TMR_THOUS zliczającego czas w setnych częściach



Rys. 1. Algorytm pomiaru czasu cyklu i czasu odpowiedzi

Fig. 1. An algorithm of measure a cycle time and a response time

sekundy. Na podstawie pomiaru czasu T_{obl} możliwe jest obliczenie czasu pojedynczego wykonania funkcji *Pozycja*, który będzie oznaczony przez T_{poz} . Związek między tymi czasami jest następujący:

$$T_{obl} = N_p T_{poz} + T_{to} \tag{1}$$

gdzie T_{to} oznacza czas uruchomienia i wykonania timera użytego do pomiaru czasu T_{obl} . Na podstawie (1) można natychmiast wyznaczyć czas pojedynczego wykonania funkcji *Pozycja*:

$$T_{poz} = \frac{T_{obl} - T_{to}}{N_p} \tag{2}$$

Z kolei czas całego cyklu programowego podczas realizacji testu może być wyrażony następująco:

$$T_c = T_{obl} + T_{tc} + T_d \tag{3}$$

przy czym T_{tc} – czas uruchomienia i wykonania się timera użytego do pomiaru czasu cyklu, a T_d – czas realizacji wszystkich pozostałych zadań związanych z wykonaniem programu w danym cyklu, np. komunikacji z systemem SCADA, diagnostyki itp.

Zmierzone czasy T_{obl} i T_c są po każdym eksperymencie odczytywane przez aplikację SCADA i zapisywane w bazie danych systemu w formacie MS EXCEL. Komunikacja z systemem SCADA odbywa się w końcowej fazie każdego cyklu programowego (po zapisie wyjść PLC a przed wykonaniem diagnostyki). Uruchomienie i zatrzymanie algorytmu również jest realizowane z poziomu aplikacji SCADA.

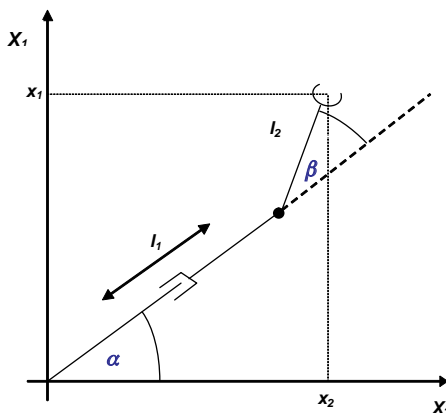
Funkcja testowa o dużej złożoności obliczeniowej

W funkcji testowej *Pozycja* rozwiązano przykładowe zadanie kinematyki prostej dla robota przemysłowego o trzech stopniach swobody [13]. Równanie kinematyki prostej dla łańcucha kinematycznego (rys. 2) ma następującą postać:

$$\begin{cases} x_1(t) = l_1(t) \sin(\alpha(t)) + l_2(t) \sin(\alpha(t) + \beta(t)) \\ x_2(t) = l_1(t) \cos(\alpha(t)) + l_2(t) \cos(\alpha(t) + \beta(t)) \end{cases} \tag{4}$$

gdzie: $x_1(t)$, $x_2(t)$ – położenie odpowiednio na osi X_1 i X_2 w dwuwymiarowym układzie odniesienia, $l_1(t)$ – długość pierwszego ramienia robota, l_2 – długość drugiego ramienia robota (wartość stała), $\alpha(t)$ – kąt (rad) między pierwszym ramieniem i osią X_1 , $\beta(t)$ – kąt (rad) między drugim ramieniem i prostą powstałą w wyniku przedłużenia ramieniem l_1 .

Równanie (4) zostało zapisane w podprogramie *Pozycja*, wywoływanym warunkowo z poziomu programu głównego. Do zapisu działań (4) wykorzystano język drabinkowy. Parametry równań były zadawane z poziomu aplikacji SCADA.



Rys. 2. Przykładowy łańcuch kinematyczny
Fig. 2. An exemplary cinematic chain

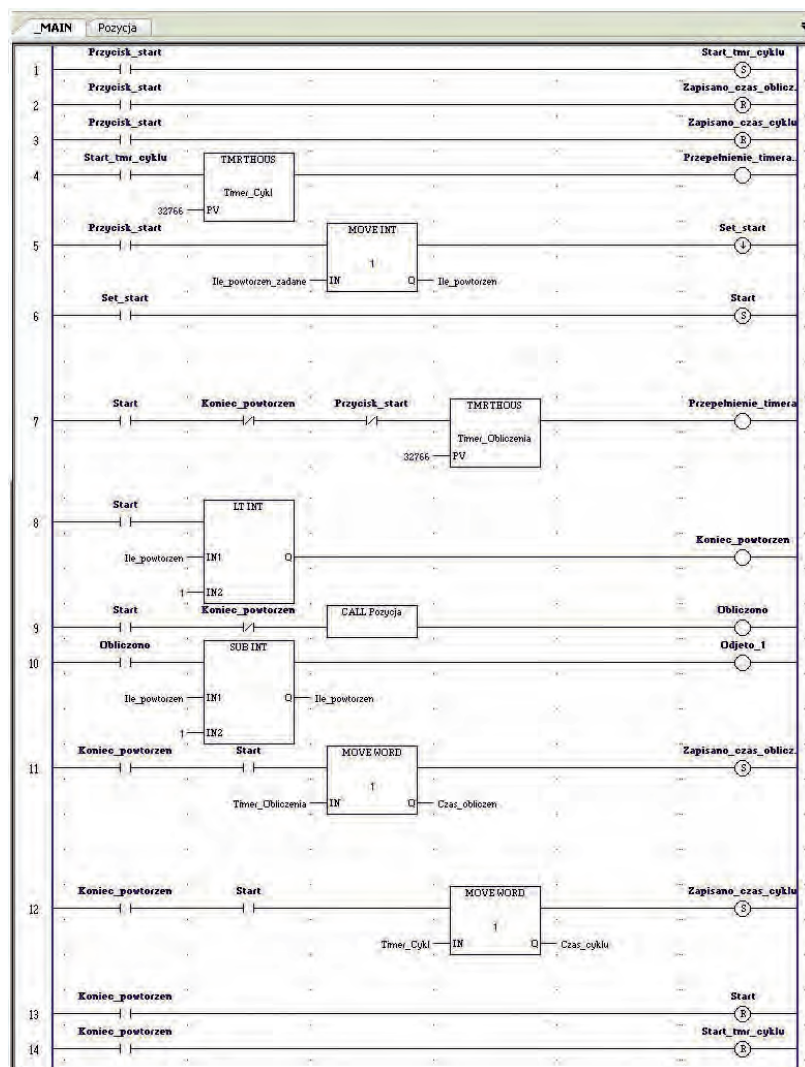
Implementacja drabinkowa algorytmu

Program w języku drabinkowym realizujący algorytm pokazany na rys. 1 [3] został zbudowany z użyciem narzędzia programowego PROFICY MACHINE EDITION, dedykowanego do konfiguracji sterowników GE FANUC. Zawiera on program główny oraz podprogram testowy *Pozycja*, realizujący równania (4). Program główny w języku drabinkowym pokazano na rys. 3.

Podprogram testowy *Pozycja* również został zrealizowany w języku drabinkowym. Do uruchamiania aplikacji oraz zbierania wartości czasów cyklu i czasów obliczeń wykorzystano współpracującą z programem na PLC aplikację SCADA, zbudowaną z wykorzystaniem środowiska InTouch.



Rys. 4. Uproszczony schemat układu doświadczalnego
Fig. 4. A simple diagram of experimental system



Rys. 3. Implementacja drabinkowa programu testowego
Fig. 3. A ladder implementation of testing program

Badania doświadczalne

Badania doświadczalne zrealizowano na sterowniku PLC GE FANUC VersaMax Micro wersja 23-punktowa. Aplikacja SCADA oraz środowisko konfiguracyjne były zlokalizowane na komputerze klasy PC, który komunikował się z PLC za pośrednictwem portu RS-232 oraz konwertera RS-232 na USB (rys. 4).

Podczas realizacji eksperymentów przyjęto następujące założenia:

- w trakcie eksperymentów sterownik nie korzysta z żadnych wejść ani wyjść procesowych, uruchomienie i zatrzymanie zbierania danych oraz zadawanie parametrów wejściowych (np. liczby powtórzeń pętli N_p lub liczby zbieranych próbek) odbywa się wyłącznie z poziomu aplikacji SCADA,
- wielkość czasu cyklu w sterowniku ustawiono bez ograniczeń (parametr SWEEP MODE: NORMAL),
- zapis wyników pomiarów jest dokonywany również z poziomu systemu SCADA w formacie obsługiwanym przez MS EXCEL,
- liczbę wykonań funkcji testowej *Pozycja* przyjęto równą $N_p = 1000$,
- liczbę zebranych próbek przyjęto równą 1000.

Wyniki eksperymentów są opisane przez histogramy (rys. 5 i 6) oraz tabele 1 i 2. Na rys. 5 oraz w tabeli 1 opisano wyniki pomiarów czasu obliczeń T_{obb} , natomiast na rys. 6 oraz w tabeli 2 podano wyniki pomiarów całego czasu cyklu podczas wykonywania eksperymentów. Na podstawie analizy tych histogramów można zauważyć, że mogą one być opisane rozkładem normalnym:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (5)$$

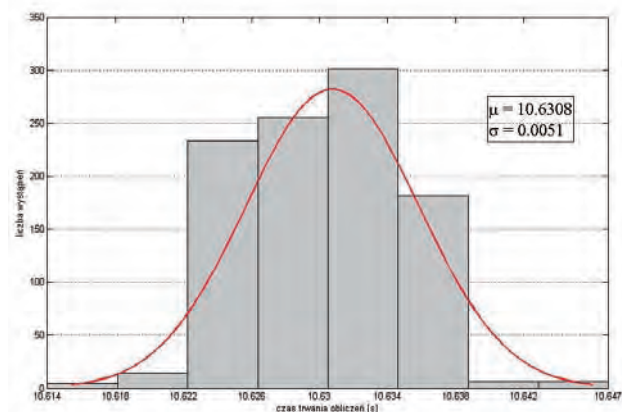
gdzie σ – odchylenie standardowe, μ – mediana. Przebieg funkcji (5) jest naniesiony na histogramy.

Analiza uzyskanych wyników pozwala na sformułowanie kilku wniosków dotyczących prezentowanej w pracy metody pomiaru uwarunkowań czasowych bazującej na języku drabinkowym:

- Duża liczba obliczeń funkcji trygonometrycznych powoduje bardzo znaczne obciążenie obliczeniowe CPU sterownika PLC, i wynik ten jest analogiczny do rezultatów otrzymanych przy użyciu innych metod pomiarowych i na innym sprzęcie [10].

Tab. 1. Parametry statystyczne czasu obliczeń T_{obl} **Tab. 1.** Statistical parameters of the calculation time

Parametr	Wartość
Średnia	10,6308 s
Wartość maksymalna	10,647 s
Wartość minimalna	10,614 s
Zakres	0,033 s
Mediana m	10,63 s
Odchylenie standardowe s	0,000051 s
Wariancja próbki	0,000026 s ²

**Rys. 5.** Rozkład wartości czasu obliczeń T_{obl} **Fig. 5.** The distribution of the calculation time

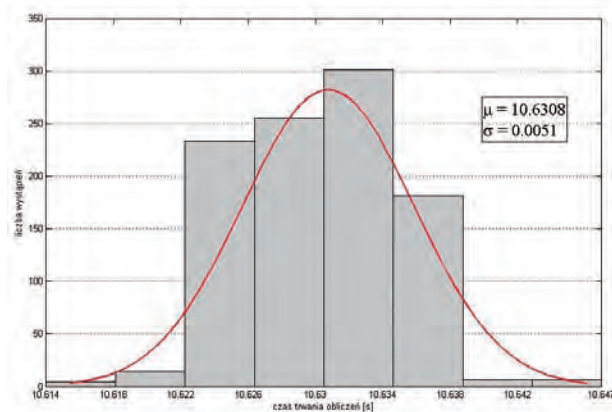
- Wykonanie funkcji testowej stanowi główną część obciążenia CPU podczas wykonywania programu. Wskazuje na to porównanie wartości czasów: T_{obl} oraz T_c , gdzie czas trwania całego cyklu jest tylko nieznacznie dłuższy od czasu wykonania funkcji testowej.
- Histogramy pokazujące rozkład wartości czasów obliczeń i cyklu również potwierdzają poprawność wykonanych doświadczeń, gdyż ilości wystąpień poszczególnych wartości mogą być z dobrym przybliżeniem opisane rozkładem normalnym.

Uwagi końcowe

- Wyniki testów na rzeczywistym systemie PLC w każdym aspekcie są zgodne z oczekiwaniami: funkcje trygonometryczne powodują znaczne obciążenie CPU, histogramy wartości czasów mogą być opisane rozkładem normalnym, co wskazuje na poprawność zaproponowanej w pracy metody.
- Zaproponowana w pracy metoda wykorzystuje podstawowe narzędzia programowe sterowników PLC: język drabinkowy i timery, które są dostępne na każdej platformie sprzętowo-programowej PLC. Z tego względu może ona być zastosowana do badań uwarunkowań czasowych na każdej platformie PLC.
- Podczas realizacji eksperymentów dane doświadczalne były odczytywane przez system SCADA po każdym cyklu programowym, co należy traktować jako zakłócenie pomiaru. Ten sposób odczytu wynika jednak ze specyfiki badanej platformy PLC – testowany sterownik nie oferu-

Tab. 2. Parametry statystyczne czasu cyklu T_c **Tab. 2.** Statistical parameters of the cycle time

Parametr	Wartość
Średnia	10,7965 s
Wartość maksymalna	10,876 s
Wartość minimalna	10,774 s
Zakres	0,102 s
Mediana m	10,797 s
Odchylenie standardowe s	0,0098 s
Wariancja próbki	0,000096 s ²

**Rys. 6.** Rozkład wartości czasu cyklu T_c **Fig. 6.** The distribution of the cycle time during experiments

- je możliwości zdefiniowania, zapisu i późniejszego odczytu przez system SCADA zmiennej złożonej typu tablica, która byłaby najlepszym typem zmiennej do gromadzenia wyników eksperymentów.
- Przyjęcie stosunkowo dużej liczby powtórzeń wykonania funkcji testowej „Pozycja” wynika z konieczności zapewnienia odpowiedniej dokładności pomiarów czasu, które są realizowane z użyciem timerów dostępnych z poziomu programu. Ten sposób realizacji testów implikuje, że mierzony czas powinien być znacznie dłuższy od najkrótszej podstawy czasu timera, która w przypadku rozważanego sterownika wynosi 10 ms.

Bibliografia

1. Benett S.: *Real-time computer control: an introduction*. Prentice Hall New York 1994.
2. Craig J.J.: *Wprowadzenie do robotyki. Mechanika i sterowanie*. WNT Warszawa, 1993.
3. Mukawa M.: *Analiza własności czasu rzeczywistego sterownika PLC GE Fanuc*. Praca dyplomowa magisterska zrealizowana na Wydziale EAIIE AGH pod opieką K. Oprzędkiewicza, 2010.
4. Niedźwiedz A.: *Ocena przydatności symulatora PLC do badań testowych czasu rzeczywistego*. Praca dyplomowa magisterska zrealizowana na Wydziale EAIIE AGH pod opieką K. Oprzędkiewicza, 2009.
5. Oprzędkiewicz K.: *Uwarunkowania czasowe realizacji specjalnych algorytmów sterowania w systemach PLC*. PAR 4/2003, s. 48-52.

6. Oprędkiewicz K.: *Programowy pomiar czasu realizacji złożonych procedur obliczeniowych w środowisku PLC*. PAR, 2/2006, s. 22–25,
7. Oprędkiewicz K.: *Porównanie języków programowania sterowników PLC pod kątem spełnienia wymagań czasu rzeczywistego*. PAR, 12/2006, s. 5–9,
8. Oprędkiewicz K.: *Spełnienie wymagań czasu rzeczywistego na platformie sprzętowo-programowej „soft PLC” zrealizowanej na komputerze klasy PC*. Automatyza – nowości i perspektywy: XI konferencja naukowo-techniczna : 14–16 marca 2007, Warszawa, materiały konferencyjne. PIAP, Warszawa, 2007, s. 1–8,
9. Oprędkiewicz K.: *Praktyczne sterowanie systemami dynamicznymi z widmem punktowym i parametrami przedziałowymi*. Wyd. AGH, seria monografie, nr 186, 2008.
10. Oprędkiewicz K.: *Programowe metody diagnostyki spełnienia wymagań czasu rzeczywistego w systemach PLC*. [w:] Trybus L., Samolej S. (red.): *Metody wytwarzania i zastosowania systemów czasu rzeczywistego*. WKŁ, Warszawa 2010, s. 163–172.
11. Pilat A., Grega W.: *Hardware and software architectures for reconfigurable time-critical control tasks*. Computer Science, vol 8, 2007, s. 69–81,
12. Wurmsdobler P.: *Linux for Real-Time PLC Control? Slower is Easier*. ISA, 2001.
13. Zaczek M.: *Materiały pomocnicze do ćwiczeń laboratoryjnych z przedmiotu „Konstrukcja i programowanie robotów”* (niepublikowane).
14. GE Fanuc Automation. *Sterowniki programowalne. Seria 90-30/VersaMax/Micro. Opis funkcji*. GE Fanuc Automation North America, Kraków 1999.

15. GE Fanuc Automation. *Sterowniki programowalne. VersaMax Micro i Nano. Podręcznik użytkownika*. GE Fanuc Automation North America, Kraków 2002. ■

The software-based tests of real-time requirements meeting at PLC with the use of LD

Abstract: In a paper a new, LD-based method of real-time requirements meeting at PLC is presented. During algorithm construction basic interpretation rules for LD language are applied. To time tests standard timers were applied. Experimental tests were run with the use of GE FANUC VersMax Micro. The results of tests confirmed the correctness of proposed method.

Keywords: digital control, PLC, real-time requirements, ladder diagram

dr hab. inż. Krzysztof Oprędkiewicz

Obecnie zatrudniony w Katedrze Automatyki na Wydziale Elektrotechniki, Automatyki, Informatyki i Elektroniki AGH na stanowisku adiunkta oraz w Instytucie Politechnicznym w Państwowej Wyższej Szkole Zawodowej w Tarnowie na stanowisku profesora nadzwyczajnego. Zainteresowania naukowe: problemy sterowania systemami dynamicznymi o niepewnych parametrach, sterowanie cyfrowe, automatyka przemysłowa.

e-mail: kop@uci.agh.edu.pl



REKLAMA

ROBOTYKA.com

wortal branżowy dla profesjonalistów i pasjonatów

Firmy
Nowości
Produkty
Publikacje
Wydarzenia
150 stron teorii
Forum dyskusyjne
Informacje o targach i konferencjach

Jedyny wortal w Polsce poświęcony w pełni tematyce związanej z robotyką

TWÓJE WROTA DO ŚWIATA NAJNOWSZYCH TECHNOLOGII

WIADOMOŚCI | FIRMY | PRODUKTY | WYDARZENIA | TEORIA

O NAS | OFERTA | REKLAMA | LINKI | KONTAKT

www.robotyka.com

SZRZEGIWA | OFERTY I SPRZEDAŻ | ZAPISZ SIĘ NA OFERTY | FORUM

Wortal Robotyka.com został stworzony z myślą o połączeniu użytkowników, producentów oraz pasjonatów robotyki w Polsce i ma on na celu zjednoczenie działań oraz usystematyzowanie wiadomości z tego zakresu. Informacje o produktach pozwolą na szybkie odnalezienie i dobranie odpowiedniej technologii dla Państwa przedsiębiorstwa a informacje o firmach ułatwią odnalezienie odpowiedniego kontrahenta lub partnera handlowego. W dziale wiadomości pragniemy Państwu na bieżąco przekazywać najnowsze informacje z dziedziny robotyki z całego świata. Dział wydarzenia to miejsce, w którym znajdą Państwo informacje na temat imprez targowych, seminariów, konferencji naukowo-technicznych oraz innych wydarzeń związanych z robotyką w Polsce i na świecie. Dział teoria to źródło usystematyzowanej wiedzy o teorii manipulatorów i robotów.

info@robotyka.com, tel.: (+48) 12 410 02 87, fax: (+48) 12 410 87 50

www.robotyka.com

Nie trać czasu na długie poszukiwania!

www.energoelektronika.pl
marketing@energoelektronika.pl
www.energoelektronika.pl

DARMOWY WPIS PODSTAWOWY

- nowości z branży
- oferty kupna i sprzedaży
- artykuły naukowe
- przegląd prasy branżowej
- aktualności i nowinki techniczne
- szeroka baza firm
- szczegółowe informacje techniczne na temat urządzeń i podzespołów
- kalendarium wydarzeń
- źródło wiedzy i informacji naukowych
- słownik techniczny angielsko-polski, polsko-angielski

Wortal jest doskonałym i skutecznym sposobem na reklamę Państwa firmy. Energoelektronika.pl była odwiedzana średnio 66 000 razy miesięcznie w 2008 roku. Posiada bazę klientów gromadzącą ponad 10 000 firm.