

Józef KULISZ, Robert CZERWIŃSKI, Jan MOCHA, Mirosław CHMIEL

POLITECHNIKA ŚLĄSKA, INSTYTUT ELEKTRONIKI,
ul. Akademicka 16, 44-100 Gliwice

Komputerowy symulator obiektu przeznaczony do testowania oprogramowania sterowników PLC

Dr inż. Józef KULISZ

Ukończył studia na Wydziale Automatyki, Elektroniki i Informatyki Politechniki Śląskiej w roku 1992. Pracę doktorską obronił w roku 2003. Jest adiunktem w Instytucie Elektroniki Politechniki Śląskiej. Jego zainteresowania naukowe obejmują projektowanie układów cyfrowych ze szczególnym uwzględnieniem układów logiki programowalnej, sterowniki programowalne oraz cyfrowe przetwarzanie sygnałów.



e-mail: jozef.kulisz@polsl.pl

Dr inż. Robert CZERWIŃSKI

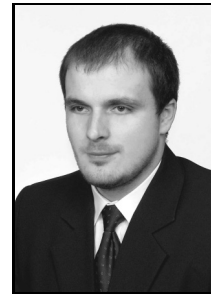
Studia na Wydziale Automatyki, Elektroniki i Informatyki Politechniki Śląskiej ukończył w 2001 roku. Na tymże wydziale obronił pracę doktorską w 2006 roku. Obecnie jest adiunktem w Instytucie Elektroniki Politechniki Śląskiej. Jego zainteresowania naukowe skupiają się wokół projektowania układów cyfrowych z wykorzystaniem układów logiki programowalnej oraz systemów mikroprocesorowych.



e-mail: robert.czerwinski@polsl.pl

Mgr inż. Jan MOCHA

Ukończył studia na Wydziale Automatyki, Elektroniki i Informatyki Politechniki Śląskiej w 2007 roku. Jest doktorantem w Instytucie Elektroniki Politechniki Śląskiej. Jego zainteresowania naukowe koncentrują się wokół układów logiki programowalnej w szczególności dynamicznej częściowej rekonfiguracji układów programowalnych FPGA, praktycznych aspektów wykorzystania układów elektronicznych w aplikacjach przemysłowych i medycznych oraz zagadnień związanych z kompatybilnością elektromagnetyczną.



e-mail: jan.mocha@polsl.pl

Dr inż. Mirosław CHMIEL

Ukończył studia na Wydziale Automatyki, Elektroniki i Informatyki Politechniki Śląskiej w 1992 roku. Na tymże wydziale w roku 2003 obronił pracę doktorską z dziedziny elektronika. Obecnie jest adiunktem w Instytucie Elektroniki Politechniki Śląskiej. Jego zainteresowania naukowe koncentrują się wokół cyfrowych układów sterowania, ze szczególnym uwzględnieniem konstruowania i wykorzystywania systemów opartych o sterowniki programowalne PLC.



e-mail: miroslaw.chmiel@polsl.pl

Streszczenie

W artykule przedstawiono koncepcję programowego symulatora obiektu sterowania przeznaczonego do uruchamiania i testowania oprogramowania dla sterowników PLC. Symulator emuluje zachowanie fizycznego obiektu przemysłowego i komunikuje się ze sterownikiem za pośrednictwem karty wejścia-wyjścia podłączanej do komputera. Pozwala on na przetestowanie tworzonej aplikacji bez udziału fizycznego obiektu, dzięki czemu znacząco przyspiesza proces tworzenia, uruchamiania oraz testowania oprogramowania.

Słowa kluczowe: sterowanie procesami przemysłowymi, programowalne sterowniki logiczne PLC, uruchamianie oprogramowania, testowanie oprogramowania, symulatory procesów, inżynieria oprogramowania.

An object simulator for PLC software testing

Abstract

The paper discusses hardware and software tools used to support program testing and verification of Programmable Logic Controllers (PLC). Three main ideas of tools supporting PLC application development are presented: software PLC simulators (Fig. 1), software PLC simulators with software object simulators (Fig. 2), and software object simulators with a hardware PLC (Fig. 4). The last idea is discussed wider in the paper. The authors propose a new concept of the tool for supporting PLC program testing – an object simulator which is a separate device. The simulator consists of a PC equipped with an appropriate I/O card, and an object simulator program running on the PC. The object simulator program is responsible for emulating behavior of an industrial object, and providing appropriate visualization of its operation, enabling also the PLC programmer to simulate object faults. The PC does not communicate with the PLC using a network interface, but through physical I/Os of the PLC. The simulator is thus capable of testing the most of functionality built in PLC I/O modules, and time-critical functions, e. g. interrupts. The proposed concept of an object simulator can provide a reliable substitute for a physical object, and thus a significant part of software tests can be performed with use of the simulator. This can significantly facilitate and accelerate development of the application.

Keywords: industrial control, programmable logic controllers, program diagnostics, debugging, simulation, process simulators, software engineering, software productivity.

1. Wprowadzenie

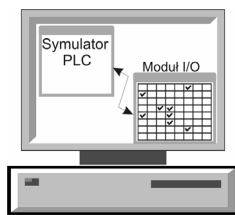
Uruchamianie i testowanie oprogramowania dla sterowników programowalnych PLC (ang. *Programmable Logic Controllers*), jest procesem skomplikowanym i stwarzającym często sporo problemów. Twarde warunki konkurencji na rynku wymuszają pracę pod presją czasu. Niejednokrotnie zdarza się, że podczas uruchamiania i testowania aplikacji dostęp do sterowanej instalacji jest ograniczony, np. ze względu na równoległe prowadzone prace montażowe lub mechaniczne. Może to utrudnić lub wręcz uniemożliwić swobodne prowadzenie testów. Problemem wymagającym specjalnej uwagi przy testowaniu aplikacji bezpośrednio na obiekcie są też kwestie bezpieczeństwa ludzi i niejednokrotnie bardzo kosztownego sprzętu.

Na wstępnym etapie uruchamiania oprogramowania dla sterowników PLC tradycyjnie wykorzystuje się zadajniki sygnałów binarnych w postaci przełączników oraz wizualizację sygnałów wyjściowych sterownika w postaci lampek lub diod LED. Rozwiązanie to jest pomocne przy uruchamianiu i testowaniu małych fragmentów programu. Jednak w przypadku setek wejść oraz wyjść „powiązanych” ze sobą skomplikowanymi algorytmami sterowania metoda ta staje się bardzo uciążliwa. Wymaga ona „ręcznego” symulowania w czasie rzeczywistym zachowania się wielu sygnałów sprzężeń zwrotnych ze sterowanego obiektu. Nadto przy ręcznym pobudzeniu wejść sterownika należyte odwzorowanie zależności czasowych pomiędzy sygnałami w rzeczywistym obiekcie może okazać się niemożliwe. Problem testowania aplikacji dla sterowników PLC zdaniem autorów nie został dotychczas w satysfakcjonujący sposób rozwiązany.

W niniejszym artykule zaproponowano alternatywne podejście do testowania oprogramowania sterowników PLC – symulator obiektu, który jest oddzielnym urządzeniem zbudowanym w oparciu o komputer PC. Zanim przedstawione zostanie proponowane rozwiązanie, przedyskutowane zostaną w skrócie najpopularniejsze dotychczas stosowane metody testowania oprogramowania dla sterowników PLC.

2. Metody testowania oprogramowania dla sterowników PLC

Najpowszechniej dostępnymi narzędziami ułatwiającymi testowanie programów dla sterowników PLC są programowe symulatory sterownika. Programowy symulator sterownika PLC jest uruchamiany na komputerze PC. Interpretuje on program sterowania napisany w języku programowania właściwym dla danego sterownika i symuluje jego działanie. Zadawanie stanów wejść oraz wizualizacja wyjść odbywa się z wykorzystaniem odpowiedniego interfejsu graficznego. Idea programowego symulatora sterownika została przedstawiona na rys. 1.

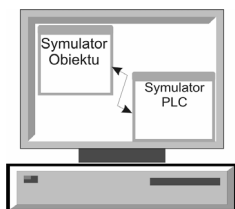


Rys. 1. Koncepcja programowego symulatora sterownika PLC
Fig. 1. The concept of a software PLC simulator

Programowe symulatory sterowników mogą być udostępniane przez producentów sterowników programowalnych jako jeden z modułów w oprogramowaniu służącym do programowania danego sterownika [1 - 4], jak również mogą być dostarczane przez firmy niezależne [5 - 9].

Uruchamianie oraz testowanie oprogramowania z wykorzystaniem programowego symulatora sterownika jest w istocie podobne do podejścia wykorzystującego zadajniki binarne, choć ze względu na wyeliminowanie sprzętowych zadajników sposób ten można uznać za tańszy. Metoda ma podobne ograniczenia, z których najważniejszym jest uciążliwość „ręcznego” symulowania sygnałów w czasie rzeczywistym.

Aby uwolnić programistę od konieczności „ręcznej” emulacji sygnałów sprzężeń zwrotnych z obiektu, zamiast symulatora samego sterownika bardziej pożądanym byłby program symulujący działanie sterowanego obiektu. Idea ta została przedstawiona na rys. 2. Oba programy, tj. symulator sterownika i symulator obiektu, pracują na tym samym komputerze PC i wymieniają między sobą stany sygnałów wejściowych i wyjściowych wirtualnego sterownika. Niestety narzędzia tego typu są znacznie mniej powszechne. Symulację sterowanego obiektu umożliwia oprogramowanie RSLogix Emulate firmy Rockwell Automation. Działanie obiektu opisuje się w nim w sposób podobny, jak program dla sterownika i za pomocą tego samego języka programowania, np. LAD [2, 3]. Graficzny edytor sterowanego obiektu wbudowany jest w program TrySim [9]. Obiekt można „zbudować” jako instalację składającą się z pewnych prostych elementów (np. dźwieg, taśmociąg, zbiornik), a następnie obserwować jego symulowane działanie.

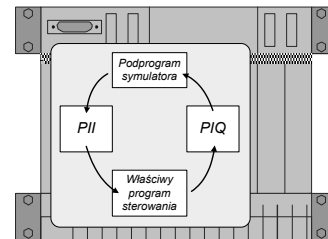


Rys. 2. Symulator sterownika PLC oraz symulator obiektu uruchamiane na komputerze PC
Fig. 2. A PLC simulator, and a software object simulator running in the same PC

Należy podkreślić, że w przedstawionych powyżej metodach programista nie ma do czynienia z fizycznym sterownikiem, lecz z mniej lub bardziej doskonałym symulatorem programowym.

Działanie fizycznego sterownika może być różne od działania symulatora np. ze względu na błędy w programie symulatora.

Ciekawa propozycja testowania oprogramowania została zaproponowana w pracy [10]. Autor proponuje dodanie do programu sterowania podprogramu symulującego działanie sterowanego obiektu. Podprogram ten wywoływany jest na początku głównej pętli systemowej sterownika. Podprogram symulatora odczytuje stan wyjść sterownika z pamięci wyjść sterownika PIQ (ang. *Process Output Image*) i na tej podstawie wyznacza stany sygnałów sprzężeń zwrotnych z symulowanego obiektu. Stany te są bezpośrednio zapisywane do pamięci wejść PII (ang. *Process Input Image*). Po wykonaniu podprogramu symulatora wykonywany jest program główny sterownika (rys. 3). Podprogram symulatora działa w sposób całkowicie transparentny dla testowanego (uruchamianego) programu głównego.



Rys. 3. Symulator obiektu jako podprogram w sterowniku PLC
Fig. 3. Object simulator as a subroutine inside the PLC

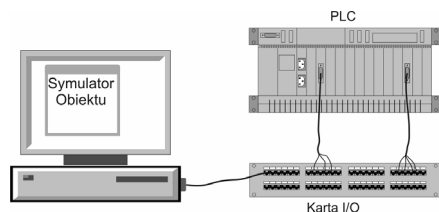
Ze względu na synchronizację stanu wyjść symulatora obiektu z pętlą główną sterownika PLC, metoda ta nie pozwala na testowanie szybkich zależności czasowych. Dlatego metoda jest polecana do testowania algorytmów na najwyższym poziomie hierarchii. Podobna idea została też przedstawiona w pracy [11], choć tam autorzy zamiast fizycznego sterownika proponują wykorzystanie jego symulatora.

Należy podkreślić, że we wszystkich przedstawionych powyżej koncepcjach programista nie ma do czynienia z fizycznymi sygnałami wejściowymi i wyjściowymi sterownika, lecz z ich odwzorowaniem w pamięci komputera. Omówione metody nie pozwalają na przetestowanie krytycznych czasowo funkcji sterownika takich jak: przerwania, szybkie liczniki czy wyjścia impulsowe, które wykonywane są w dużej części przez hardware wbudowany w moduły wejścia-wyjścia.

Uniknięcie tych ograniczeń jest możliwe, jeśli sygnały do i z obiektu będą przechodzić przez fizyczne moduły wejściowe i wyjściowe. Symulator obiektu powinien więc być niezależnym urządzeniem, podłączonym fizycznie do wejść i wyjść sterownika. Symulator powinien próbować wyjścia sterownika w regularnych odstępach czasu, wyliczać sygnały sprzężeń zwrotnych z symulowanego obiektu i podawać odpowiednie stany na wejścia sterownika. Okres próbkowania sygnałów wyjściowych powinien być krótszy od najkrótszego możliwego impulsu, który może się pojawić na wyjściu sterownika PLC.

Ad hoc można by zaproponować, aby symulatorem obiektu był inny sterownik PLC, np. tzw. Soft-PLC wykonujący program symulujący działanie obiektu. Jest to jednak rozwiązanie drogie i mało wygodne, gdyż nie zapewnia ono wizualizacji symulowanego obiektu. Sterownik należałoby zatem uzupełnić o jakiś system wizualizacji. Takie podejście wymaga więc opisanie symulowanego obiektu równoległe w dwóch niezależnych programach: w programie służącym do programowania sterownika oraz w programie do wizualizacji. Wygodniejszym rozwiązaniem wydaje się być zastosowanie komputera PC wyposażonego w odpowiedni interfejs umożliwiający podłączenie do wejść/wyjść sterownika PLC i napisanie dedykowanego oprogramowania. Idea ta została pokazana na rys. 4. Na komputer PC można napisać oprogramowanie umożliwiające łatwe tworzenie modeli graficznych symulowanych obiektów sterowania oraz przypisywanie im właściwości behawioralnych. W skład oprogramowania powinna wchodzić biblioteka podstawowych parametryzowanych elemen-

tów (np. zbiorniki, zawory, pompy), które można wykorzystać do zestawienia modelu sterowanej instalacji.

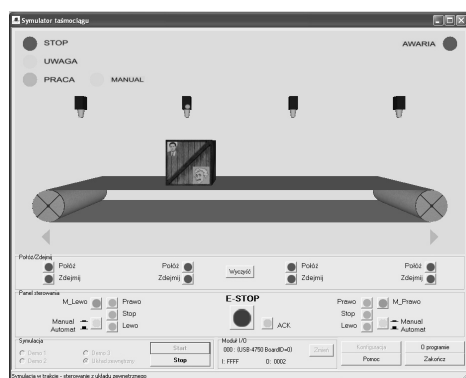


Rys. 4. Symulator obiektu pracujący na komputerze PC i wymieniający dane ze sterownikiem PLC poprzez moduły wejścia-wyjścia

Fig. 4. Object simulator as a separate device communicating with PLC I/O modules

3. Konstrukcja proponowanego symulatora obiektu

W celu weryfikacji działania oraz możliwości praktycznego wykorzystania zaproponowanej idei przygotowano symulator prostego obiektu, jakim jest przenośnik taśmowy. Wygląd okna głównego programu został przedstawiony na rys. 5. Program symulatora został napisany w języku C/C++ z wykorzystaniem środowiska programistycznego Borland C++ Builder. Elementy graficzne zostały przygotowane jako grafika trójwymiarowa (3D) w oparciu o bibliotekę OpenGL.



Rys. 5. Wygląd okna głównego stworzonego symulatora obiektu

Fig. 5. The main window of the object simulator

Program symulatora odpowiedzialny jest również za wymianę danych ze specjalizowaną kartą wejścia-wyjścia. Jako kartę wejścia-wyjścia wykorzystano kartę akwizycji danych USB-4750 firmy Advantech [12]. Karta ta posiada 32 cyfrowe izolowane kanały wejścia-wyjścia i podłączona jest do komputera PC poprzez interfejs USB. Dostępne są również karty z wejściami oraz wyjściami analogowymi.

Próbkowanie sygnałów wyjściowych sterownika odbywa się cyklicznie co 1ms. Na podstawie stanów wyjść sterownika oraz według zaprogramowanego algorytmu działania obiektu symulator generuje odpowiednie sygnały sprzężenia zwrotnego podawane na wejścia sterownika.

Istotną kwestią przy tworzeniu symulatora jest dokładne odwzorowanie zachowania się fizycznego obiektu. Weryfikacja poprawności programu sterowania w sterowniku będzie tym lepsza, im wierniej symulator obiektu będzie odzwierciedlał rzeczywisty obiekt sterowania. Oprogramowanie symulacyjne daje ponadto możliwość wykrywania i sygnalizowania mogących się pojawić w programie sterowania błędów, w szczególności takich które mogą spowodować fizyczne uszkodzenie sterowanego obiektu. W przypadku przenośnika taśmowego błędem takim jest jednoczesne włączenie napędów dla lewego i prawego kierunku przesuwu taśmy.

Szybkość symulatora oraz zastosowanej karty wejścia-wyjścia są wystarczające także do testowania poprawności działania przebrań w sterowniku. Niestety testowanie szybkich liczników oraz wyjść impulsowych sterownika z wykorzystaniem karty USB-4750 nie jest możliwe. Wymagana jest wtedy bardziej zaawansowana karta wejścia-wyjścia ze sprzętowym wsparciem akwizycji tego typu sygnałów.

4. Wnioski

W niniejszym artykule zaproponowano alternatywne podejście do testowania oprogramowania dla sterowników PLC – symulator obiektu, który jest oddzielnym urządzeniem zbudowanym w oparciu o komputer PC wyposażony w odpowiedni interfejs. Działanie sterowanego obiektu symulowane jest za pomocą oprogramowania pracującego na komputerze PC. Praktyczna przydatność proponowanej metody została zweryfikowana przez napisanie przykładowej symulacji prostego obiektu przemysłowego.

Uzyskany okres próbkowania wyjść sterownika na poziomie 1 ms, w porównaniu do czasu obiegu pętli typowego sterownika PLC, jest na tyle krótki, że umożliwia on w praktyce symulację większości obiektów przemysłowych. Nie mniej jednak testowanie pewnych czasowo krytycznych elementów programu sterowania może wymagać zastosowania bardziej zaawansowanej karty wejścia-wyjścia podłączonej do komputera PC.

Dalsze prace rozwojowe nad opracowanym narzędziem będą miały na celu rozbudowę biblioteki standardowych elementarnych obiektów przemysłowych oraz dodanie możliwości symulowania obiektów, w których występują wielkości analogowe.

Należy podkreślić, że wykorzystanie symulatora obiektu do uruchamiania oraz testowania aplikacji dla sterowników PLC, choć powinno umożliwić wykrycie większości błędów, nie może jednak zastąpić ostatecznej weryfikacji programu na rzeczywistym fizycznym obiekcie.

5. Literatura

- [1] PLCsim, S7-PLCSim V5.4 – User Manual, A5E00992424-01, Siemens AG, 2007.
- [2] RSLogix, RSLogix Emulate. Getting Results Guide Doc ID EMULAT-GR002A-EN-P, Rockwell Automation, www.software.rockwell.com, 2004.
- [3] RSLogix, RSLogix Emulate 5000. Getting Results Guide, Publication LGEM5K-GR015A-EN-P, Rockwell Automation, www.software.rockwell.com, 2005.
- [4] Unity, Modicon M340 PLC Automation Platform Unity Software, DIA6ED2061001EN-US, Schneider Electric, www.us.telemecanique.com, 2007.
- [5] IBHSoftec, S7 Simulation PLC, http://www.ibhsoftec-sps.de/english/Selection_Sim.htm.
- [6] IBHSoftec, Extended Simulation S5, http://www.ibhsoftec-sps.de/english/Selection_Sim.htm.
- [7] CoDeSys, User Manual for PLC Programming with CoDeSys 2.3, 3S - Smart Software Solutions GmbH, 2003.
- [8] ICS Triplex ISaGRAF, ISaGRAF <http://www.isagraf.com>.
- [9] TrySim, Cephalos Gesellschaft für Automatisierung mbH, <http://www.trysim.de/en/>.
- [10] Kulisz J.: PLC Software Development supported by utilising a Simulator Program Block, Proc. of the International Conference on Programmable Devices and Systems, PDS'2004, Cracow, Poland, 18 - 19 listopada 2004, str. 121-125.
- [11] Mikoś Z., Wróbel G., Hayduk G., Kwasnowski P., Jachimski M., Zygmunt H.: Środowisko ISaGRAF do tworzenia rozproszonych aplikacji sterowania i regulacji, Systemy czasu rzeczywistego. Postępy badań i zastosowania, WKiŁ, Warszawa, 2009, str. 43-62.
- [12] Advantech, USB-4750 32-channel Isolated Digital I/O USB Module, User Manual, http://www.advantech.com/products/USB-4750/mod_1-2MLJMR.aspx, 2009.