

Budowa i testowanie modułów funkcjonalnych w programie Arena do modelowania i sterowania wybranymi obiektami systemu produkcyjnego

Grzegorz Madej, Waldemar Małopolski

Instytut Technologii Maszyn i Automatykacji Produkcji, Wydział Mechaniczny, Politechnika Krakowska

Streszczenie: W artykule przedstawiono metodę budowania nowych modułów funkcjonalnych w programie symulacyjnym Arena. Opracowano dwa moduły symulujące działanie dwóch złożonych obiektów transportowych przykładowego systemu produkcyjnego i przystosowano te moduły do współpracy ze sterownikami PLC. Zbudowano model symulacyjny z użyciem nowych modułów i przeprowadzono sterowanie ich działaniem za pomocą sterownika PLC.

Słowa kluczowe: symulacja, integracja, programowanie PLC, testowanie

DOI: 10.14313/PAR_209-210/106

Wzrost wydajności obliczeniowej produkowanych komputerów oraz rozwój narzędzi informatycznych przyczynia się do wzrostu nieustanniegostalego rozwoju możliwości funkcjonalnych nowo powstających programów komputerowych. W grupie tej znajdują się również programy przeznaczone do modelowania i symulacji procesów dyskretnych lub ciągłych, czy też o przeznaczeniu specjalistycznym. Można zaobserwować wzrost wydajności tych narzędzi, zwiększenie funkcjonalności oraz coraz szersze wykorzystanie elementów grafiki 3D. Inną bardzo ciekawą funkcjonalnością jest możliwość współpracy z urządzeniami rzeczywistymi – to nowa jakość w zakresie stosowania symulacji komputerowej. Przykładem może być program Arena przeznaczony do modelowania procesów dyskretnych. Umożliwia on wymianę danych z układem sterowania PLC w trakcie procesu symulacji. Dzięki temu program Arena może być stosowany do testowania poprawności działania układów sterowania.

1. Wprowadzenie

Bez wątpienia bezpośrednią przyczyną wzrostu liczby stosowanych sterowników PLC jest rozwój gospodarczy związany z wprowadzaniem nowych technologii. Nie bez znaczenia jest też spadek cen sterowników oraz ich coraz większe możliwości funkcjonalne. Sterowniki PLC są wykorzystywane do prostych zastosowań oraz do budowania

złożonych systemów sterujących skomplikowanymi obiektami. W każdym przypadku bardzo ważne jest dobranie odpowiedniego sterownika/sterowników oraz opracowanie poprawnego programu [7, 10]. Wszelkie inwestycje mają swe ograniczenia czasowe i finansowe. Bardzo ważna jest minimalizacja kosztów i skrócenie czasu realizacji projektu, bez względu na to, czy jest to nowy projekt, czy też modernizacja istniejącego obiektu. Zastosowanie programu symulacyjnego do weryfikacji poprawności programu sterującego czy też całego systemu sterowania [9] może przynieść wiele korzyści. Przede wszystkim pozwala dokonać weryfikacji programu sterującego przed jego implementacją do obiektu rzeczywistego. Dzięki temu ewentualne błędy nie doprowadzą do awarii, co w wielu przypadkach mogłoby mieć poważne konsekwencje finansowe.

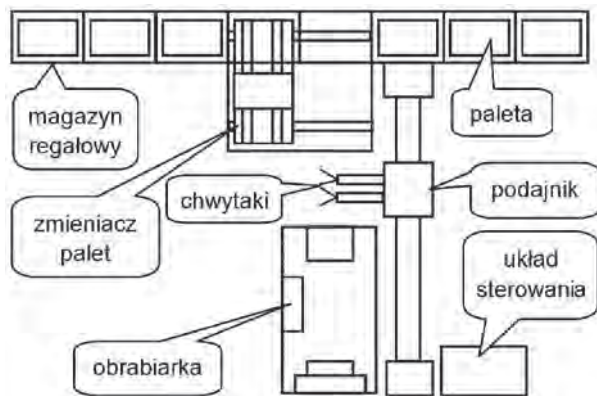
Obecnie na rynku są dostępne liczne komercyjne programy symulacyjne o różnym przeznaczeniu. Można wyróżnić dwie zasadnicze grupy: do procesów ciągłych lub dyskretnych. Do modelowania i symulacji procesów ciągłych są stosowane programy Vensim [5], VisSim [6] oraz LabVIEW [2]. Do modelowania i symulacji procesów dyskretnych można wykorzystać m.in. ProModel [3], Simul8 [4] i Arena [1]. Obecnie większość programów symulacyjnych ma przyjazny interfejs graficzny. Dzięki temu tworzenie modeli jest o wiele łatwiejsze. Można budować modele pojedynczych obiektów lub dużych złożonych systemów. Do budowania modeli w programie Arena wykorzystuje się gotowe moduły o określonej funkcjonalności, które są pogrupowane w odpowiednich szablonach [8]. Dodatkowo bardzo łatwo można wyposażać budowane modele w elementy wizualizacji i animacji. Dzięki temu znacznie łatwiej jest przeprowadzić obserwację procesu symulacji. Najważniejsza jest jednak poprawność zbudowanego modelu. Należy dążyć do tego, aby model był możliwie najprostszymi, a zarazem odpowiednio odwzorowywał zachowanie modelowanego obiektu lub systemu. Program Arena umożliwia współpracę ze sterownikami PLC. Dzięki temu można budować modele do testowania układów sterowania. Ta funkcjonalność nie występuje we wszystkich programach symulacyjnych. Zbudowanie modelu prostego obiektu nie jest bardzo trudne. Znajac ogólną zasadę działania oraz dokumentację techniczną, można dość

dokładnie odwzorować logikę działania żadanego urządzenia w programie symulacyjnym, a może być ona bardzo skomplikowana. Odwzorowanie logiki działania może być pracochłonne i czasochłonne. Budowanie modeli bardziej rozbudowanych układów, w których znajduje się pewna liczba podobnych urządzeń może być bardzo kłopotliwe. Rozwiązaniem tego problemu jest budowa własnych modułów o określonej funkcjonalności. Ich głównym celem jest „spakowanie” rozbudowanej logiki obiektu w małą formę pojedynczego modułu. Ważną cechą takiego rozwiązania jest możliwość połączenia stworzonego modułu ze sterownikiem, co można zrealizować za pomocą serwerów OPC. Głównym zadaniem tych programów jest wymiana danych, np. między sterownikami PLC i aplikacjami z nimi współpracującymi, np. Areną. W dalszej części opisano sposób tworzenia szablonów zawierających nowe moduły funkcjonalne dla przykładowego systemu produkcyjnego oraz sposób połączenia ich ze sterownikiem PLC.

2. System produkcyjny

Metodę budowy bloków funkcjonalnych przedstawiono na przykładzie obiektów uproszczonego systemu produkcyjnego (rys. 1). Podstawowymi obiektami tego systemu są: zmieniacz palet, portalowy podajnik wałków, obrabiarka i magazyn regałowy. Przedmioty do obróbki znajdują się na paletach w magazynie regałowym. W opisie pominięto transport palet w obrębie magazynu. Dwa wydzielone miejsca w magazynie są obsługiwane przez zmieniacz palet. Zmieniacz pobiera paletę z miejsca z przedmiotami do obróbki, następnie oddaje paletę po obróbce przedmiotów. Palety po pobraniu są ustawiane przez zmieniacz w odpowiednich położeniach, aby podajnik portalowy mógł pobrać kolejne przedmioty i załadować je na obrabiarkę. Po zakończeniu obróbki przedmioty są odkładane na paletę, która jest odwożona na miejsce wyjściowe do magazynu regałowego.

Jako przykład budowy bloków funkcjonalnych wybrano dwa obiekty – zmieniacz palet i podajnik wałków. Jest to podyktowane dość skomplikowaną konstrukcją tych obiektów oraz złożoną kinematyką ruchów. Głównym elemen-



Rys. 1. Przykładowy system produkcyjny
Fig. 1. An example of a production system

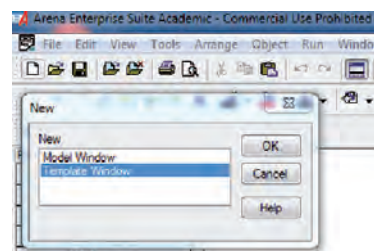
tem zmieniacza palet jest stół, który umożliwia poruszanie się w trzech kierunkach (x, y, z) oraz obrót względem osi pionowej. Dzięki ruchom prostoliniowym możliwe jest pobieranie, ustawianie w odpowiednim położeniu oraz oddawanie palety do magazynu. Za pomocą krańcówek jest ustalonych siedem pozycji stołu zmieniacza, w których może nastąpić pobranie lub oddanie przedmiotu (wałka). Obrót stołu w osi pionowej pozwala na obróbkę przedmiotów umieszczonych na palecie w dwóch rzędach. Wszystkie ruchy prostoliniowe zmieniacza oraz jego obrót są realizowane za pomocą siłowników hydraulicznych, których przemieszczanie jest ograniczone za pomocą odpowiednio ustawionych krańcówek.

Podajnik wałków jest wyposażony w dwa chwytaki. Oba chwytaki działają na tej samej zasadzie, otwierają się za pomocą siłownika hydraulicznego, a zamykają się samoczynnie za pomocą sprężyn. Jeden służy do pobierania wałka z palety, a drugi do pobierania wałka z obrabiarki po zakończeniu obróbki. Podajnik w trakcie wykonywania ruchu między paletą i obrabiarką ma cztery ustalone pozycje: trzy takty oraz położenie skrajne przy obrabiarence. Uwzględniając siedem pozycji zmieniacza palet, można pobrać przedmioty z 21 różnych pozycji.

Ze względu na złożoną kinematykę ruchów tych dwóch obiektów zamodelowanie ich działania jest stosunkowo skomplikowane i trudne. Zatem celowe wydaje się opracowanie modułów funkcjonalnych, które pozwolą na szybkie modelowanie i konfigurowanie takich obiektów, jak również obiektów o podobnej kinematyce ruchów.

3. Budowa bloków funkcjonalnych

Zbudowanie szablonu zawierającego nowe bloki funkcjonalne w programie Arena jest możliwe tylko w wersji licencjonowanej. Pojawia się wtedy możliwość utworzenia nowego szablonu (Template Window) (rys. 2). Następnie

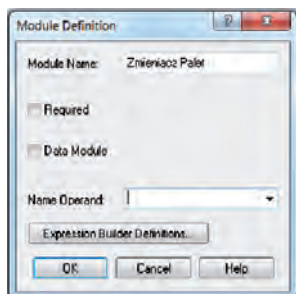


Rys. 2. Tworzenie nowego szablonu
Fig. 2. A new template creation

można dodać nowy moduł funkcjonalny. Przed użyciem takiego modułu należy zdefiniować szereg parametrów, począwszy od nazwy modułu. Pełny proces definiowania obejmuje dane pogrupowane w pięciu oknach: Dialog Design, Logic, Switch, User View i Panel Icon.

3.1. Moduł zmieniacza palet

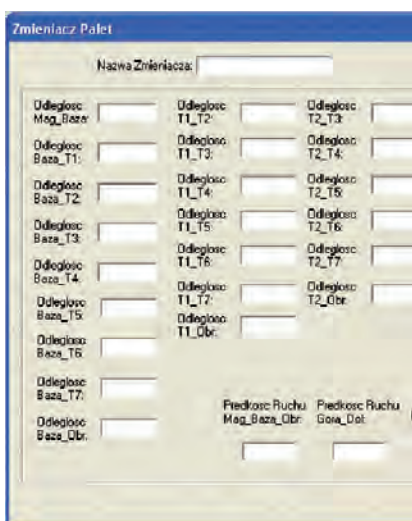
Tworzenie nowego modułu funkcjonalnego (rys. 3) rozpoczyna się od zdefiniowania parametrów w oknie Dialog Design, tj. właściwości wszystkich pól potrzebnych do wprowadzania danych, tzw. operandów. Istnieje dowol-



Rys. 3. Tworzenie nowego modułu
Fig. 3. A new module creation

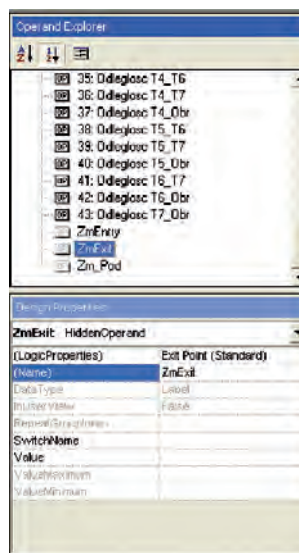
ność w zakresie rozmieszczenia operandów oraz wyboru ich typu w ramach dostępnej listy.

Do zdefiniowania modułu zmieniaacza palet użyto 43 operandy typu „TextBox” oraz 3 operandy ukryte „HiddenOperand”, odpowiedzialne za punkty wejścia i wyjścia modułu. Na rys. 4 pokazano fragment okna z rozmieszczonymi operandami.



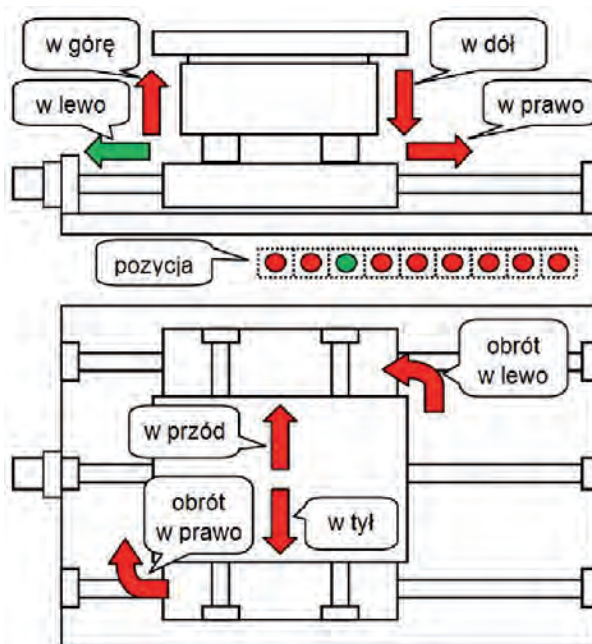
Rys. 4. Fragment okna z operandami
Fig. 4. A part of window with operands

Na rys. 5 przedstawiono okno edycji właściwości poszczególnych operandów dodanych do nowego modułu. Po zdefiniowaniu właściwości wszystkich dodanych operandów należy przejść do definiowania danych w oknie Logic. Są to najważniejsze dane, ponieważ to właśnie one opisują logikę budowanego modułu. Do budowy logiki modułu zmieniaacza palet wykorzystano moduły z następujących standardowych szablonów zawartych w programie Arena: Basic Process, Advanced Process, Advanced Transfer, Blocks i Elements. Z ich pomocą zbudowano model działania zmieniaacza, który będzie zawarty (ukryty) w nowym module. Kolejny etap budowania modułu jest związany z wprowadzeniem danych w oknie Switch. Definiowane są tu warunki przełączania między widocznymi operandami z okna Dialog Design.



Rys. 5. Okno właściwości przykładowego operandu
Fig. 5. The properties window of an exemplary operand

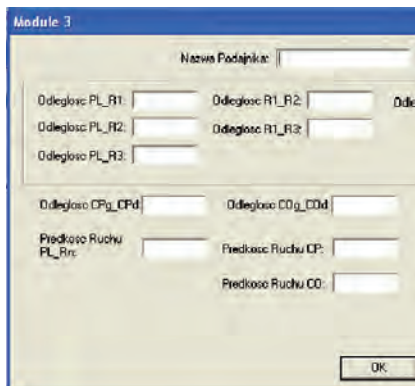
Zdefiniowane operandy można podzielić na grupy, które w zależności od spełnienia określonych warunków będą widoczne lub nie, np. zmiana wartości jakiegoś operandu może spowodować pojawienie się ukrytych do tej pory operandów (parametrów) modułu. W oknie User View widzimytworzona jest postać graficzna modułu, jaki przyjmie on po umieszczeniu w oknie budowanego modelu. Grafika dla zmieniaacza palet jest przedstawiona na rys. 6. Po utworzeniu graficznej prezentacji modułu należy umieścić w niej elementy, które będą reprezentować np. czujniki oraz napędy. Dzięki zmianie ich koloru użytkownik będzie w stanie śledzić działanie modelowanego obiektu.



Rys. 6. Wizualizacja modułu zmieniaacza palet
Fig. 6. Visualization of pallet changer module

Zmieniacz Palet

Rys. 7. Wygląd modułu zmieniacza palet w szablonie
Fig. 7. Appearance of pallet changer module in the template



Rys. 8. Okno dialogowe modułu podajnika wałków
Fig. 8. Dialog box of the feeder shafts module

Możliwe jest zdefiniowanie, jaki kolor będzie przyjmował dany element w zależności od wartości określonej zmiennej.

Ostatnia grupa danych zawartych w oknie Panel Icon Editor pozwala na utworzenie dla nowego modułu jego wyglądu wewnątrz nowo tworzonego szablonu. W tym przypadku wybrano najprostsz kształt (rys. 7). W ten sposób zbudowany moduł został przygotowany do użycia po wygenerowaniu nowego szablonu.

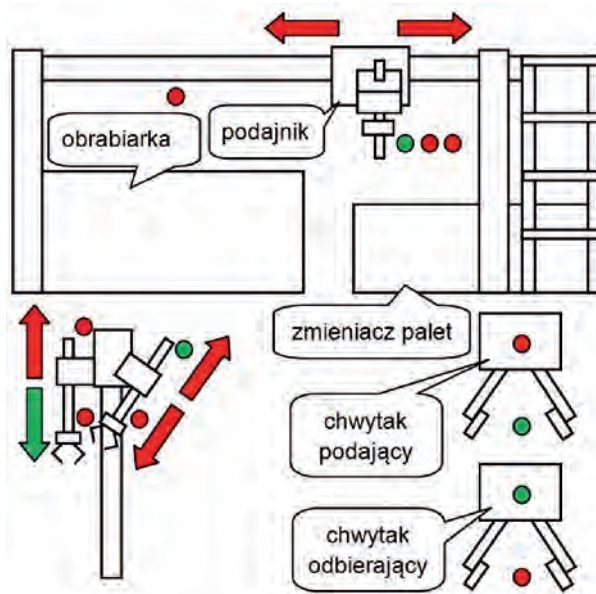
3.2. Moduł podajnika wałków

Tok postępowania przy budowaniu modułu podajnika wałków jest taki sam, jak dla zmieniacza palet. Na rys. 8 przedstawiono okno dialogowe modułu Podajnik Wałków. Moduł ten ma 12 operandów typu TextBox oraz 2 operandy ukryte. W oknie Logic Window, podobnie jak w przypadku zmieniacza palet, zawarto całą logikę działania podajnika wałków. W oknie Switch Window nie definiowano żadnych warunków. W oknie User View zdefiniowano grafikę (rys. 9) reprezentującą poszczególne elementy podajnika. W oknie Panel Icon Editor zdefiniowano wygląd ikony dla modułu podajnika. W tym przypadku również ograniczono się do prostokąta (rys. 10). Na rys. 11 pokazano ikonę reprezentującą podajnik wałków w nowym szablonie.

Na rys. 12 pokazano wygenerowany nowy szablon o nazwie System Obróbki, zawierający dwa nowo utworzone moduły: zmieniacz palet oraz podajnik wałków.

4. Testowanie nowych modułów

W celu sprawdzenia poprawności działania utworzonych modułów zbudowano zawierający je model symulacyjny systemu produkcyjnego. Do modelu podłączono sterownik PLC i przeprowadzono próbę sterowania modelem



Rys. 9. Wizualizacja modułu podajnika wałków
Fig. 9. Visualization of the feeder shafts module

Podajnik Wałków

Rys. 10. Wygląd modułu podajnika wałków w szablonie
Fig. 10. Appearance of feeder shafts module in the template



Rys. 11. Nowy szablon System Obróbki
Fig. 11. New template – production system

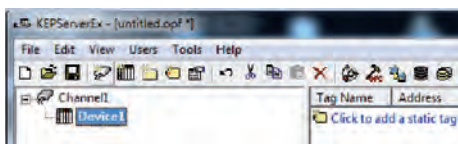
systemu. Poniżej przedstawiono etapy przygotowania do przeprowadzenia testów.

4.1. Konfiguracja serwera OPC

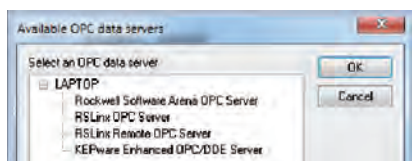
Aby połączyć sterownik PLC z nowymi modułami, konieczne jest odpowiednie skonfigurowanie serwera OPC, który będzie pośredniczył w wymianie danych. Do testów wybrano serwer firmy KEPCware. Po przeprowadzeniu odpowiedniej konfiguracji (rys. 12), można dodać odpowiedni model sterownika, który zostanie zastosowany w testach. W tym przypadku wybrano model sterownika PACSystems, a następnie zdefiniowano szereg parametrów, m.in. adres IP. Na rys. 13 przedstawiono wygląd okna programu po zakończeniu konfigurowania. Kolejnym krokiem jest zdefiniowanie zmiennych, które będą używane podczas wymiany danych. Po wprowadzeniu nazwy zmiennej oraz jej adresu należy sprawdzić, czy wpisane dane są poprawne.



Rys. 12. Konfigurowanie serwera OPC
Fig. 12. Configuring the OPC server



Rys. 13. Konfiguracja serwera OPC
Fig. 13. Configuration of OPC server



Rys. 14. Wybór serwera OPC
Fig. 14. Choosing OPC server

Po zdefiniowaniu wszystkich znaczników (zmiennych) serwer OPC jest przygotowany do współpracy ze sterownikiem. Kolejny etap przygotowań polega na skonfigurowaniu programu RSLinx, który służy do komunikacji Areny z już utworzonym serwerem OPC sterownikiem PLC. Podczas konfiguracji wybrano KEPware Enhanced OPC/DEE Server (rys. 14).

4.2. Konfiguracja środowiska programistycznego sterownika PLC

Do tworzenia programu dla sterownika PLC użyto oprogramowania Proficy Machine Edition. Tworzenie nowego projektu rozpoczęto od zdefiniowania elementów (slotów), z których składa się sterownik. W slotcie 0 wybrano rodzaj

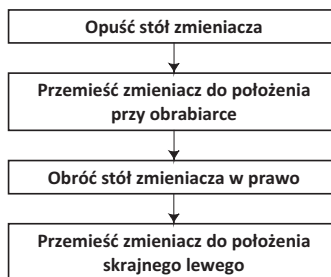
Tab. 1. Konfiguracja sterownika PLC
Tab. 1. PLC controller configuration

Slot	Element	Oznaczenie
0	Zasilacz	IC695PSD140
1	Moduł komunikacji	IC695ETM001
2	Sterownik	IC695CPU310
3		
4	Symulator zmiennych analogowych	IC694ACC300
5	-----	-----
6	Moduł wejść analogowych	IC695ALG608

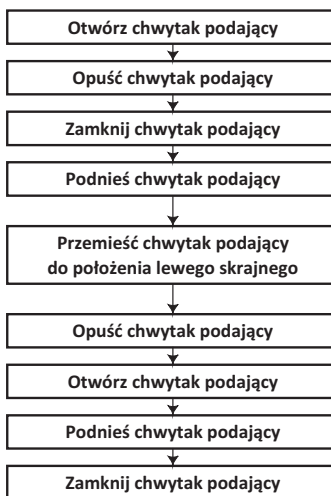
zasilacza, z jakiego korzysta sterownik (IC695PSD140). W podobny sposób dodano pozostałe moduły sterownika. Zostały one opisane w tab. 1. Po zakończeniu konfiguracji przystąpiono do tworzenia programu sterującego za pomocą języka drabinkowego, po czym przesłano go do sterownika.

4.3. Połączenie zbudowanych modułów ze sterownikiem

W celu sprawdzenia poprawności działania nowego szablону, w programie Arena zbudowano prosty model zawierający jego elementy. W modelu utworzono jednostkę reprezentującą paletę, którą przesłano do logiki wewnętrznej zmieniaacza palet, a następnie do podajnika wałków. Po przejściu przez logikę podajnika wałków, jednostkę przesłano z powrotem do zmieniaacza palet. W ten sposób krąży ona między tymi modułami do chwili zrealizowania zaplanowanych działań dla całej palety. Program sterujący tym procesem realizują algorytmy przedstawione na rys. 15 i 16.



Rys. 15. Sekwencja ruchów zmieniaacza palet
Fig. 15. The pallet changer sequence of movements



Rys. 16. Sekwencja ruchów podajnika wałków
Fig. 16. The feeder shafts sequence of movements

Name	Connection	Connected To
START_S	Connection 1	...[MAG]Program.MainProgram.SI
Zmiennicz Palet 1_Taktizacyjny	Connection 1	...[MAG]Program.MainProgram.takt
Podajnik Wałków 1_B1	Connection 1	...[MAG]Program.MainProgram.B1
Podajnik Wałków 1_B2	Connection 1	...[MAG]Program.MainProgram.B2
Zmiennicz Palet 1_Taktizacyjny	Connection 1	...[MAG]Program.MainProgram.takt
Podajnik Wałków 1_P13_PM	Connection 1	...[MAG]P13_PM
Zmiennicz Palet 1_PS_DMS	Connection 1	...[MAG]PS_DMS
Zmiennicz Palet 1_DED1	Connection 1	...[MAG]DED1

Rys. 17. Podłączanie zmiennych sterownika

Fig. 17. Connecting the controller variable

W celu przygotowania modułów do poprawnego działania należy jeszcze zdefiniować zmienne biorące udział w wymianie danych. Mając w pełni przygotowany model można przystąpić do połączenia go ze sterownikiem. W tym celu należy skonfigurować połączenie programu Arena z wcześniej przygotowanym OPC serwerem. Dzięki temu możliwe jest połączenie ze zmiennymi sterownika (rys. 17).

Przed rozpoczęciem symulacji należy uruchomić serwer OPC. Wynikiem współdziałania sterownika z modelem była symulacja działania zestawu zmiennych palet – podajnik wałków. Analizując program sterujący stwierdzono, że logika zawarta wewnątrz zbudowanych modułów jest poprawna. Algorytm sekwencji ruchów poszczególnych elementów układu odpowiadał założonemu schematowi.

5. Podsumowanie

Opisane w artykule próby budowy w programie Arena nowych modułów funkcjonalnych, zdolnych do współpracy ze sterownikami PLC, zakończyły się pomyślnie. Zaplanowana do zrealizowania sekwencja czynności w modelu zbudowanym z nowych modułów została odwzorowana poprawnie. Wymiana danych między sterownikiem a modelem również przebiegła bez problemów. W odróżnieniu od obiektów rzeczywistych, użytkownik programu symulacyjnego, dzięki dużej liczbie możliwych do modyfikacji parametrów może wykorzystać nowe moduły stworzonego szablonu w różny sposób. Na przykład, moduł zmiennych palet z odpowiednio ustalonymi odległościami może być traktowany jak rozbudowany system transportowy, np. jako wózek z dziewięcioma możliwymi do zaprogramowania stacjami docelowymi. W podobny sposób można zastosować drugi ze zbudowanych modułów. Podajnik wałków może obsługiwać np. cztery obrabiarki. Ewentualna rozbudowa modułów o dalsze możliwości nie powinna nastęrczać większych trudności.

Należy jednak wspomnieć o pewnych ograniczeniach. Program Arena nie jest systemem czasu rzeczywistego i nie gwarantuje odpowiednio szybkiej wymiany danych. Należy o tym pamiętać podczas pracy ze sterownikiem PLC i zagwarantować odpowiedni czas na aktualizację wartości wymienianych informacji. Mimo tych ograniczeń przedstawione w artykule rozwiązanie może być bardzo przydatne na etapie projektowania i testowania złożonych systemów sterowania.

Bibliografia

- [www.arenasimulation.com] – Arena simulation software (18 listopada 2013).
- [www.mathworks.com] – MathWorks (18 listopada 2013).
- [www.promodel.com] – ProModel (18 listopada 2013).
- [www.simul8.com] – SIMUL8 (18 listopada 2013).
- [www.vensim.com] – Vensim (18 listopada 2013).
- [www.vissim.com] – VisSim (18 listopada 2013).
- Kasprzyk J., *Programowanie sterowników przemysłowych*, Wydawnictwo Naukowo-Techniczne, Warszawa 2006.
- Kelton W.D., Sadowski R.P., Sadowski D.A., *Simulation with Arena*, McGraw-Hill, New York 2004.
- Małopolski W., Madej G., *Testowanie programów dla PLC w środowisku symulacyjnym Arena*, „Pomiary Automatyka Robotyka”, Nr 2, 2012, 424–428.
- Sałat R., Korpysz K., Obstawski P., *Wstęp do programowania sterowników*, Wydawnictwo Komunikacji i Łączności, Warszawa 2010. ■

Construction and testing the functional modules in Arena software for modeling and control of selected objects of the production system

Abstract: The article presents a method to build new functional modules in the Arena simulation software. Two modules have been developed to simulate the operation of two complex objects of transport in exemplary production system. These modules were adapted to interact with PLCs. Simulation model was built using the new modules and control their operation was carried out by the PLC controller.

Keywords: simulation, integration, PLC programming, testing

Artykuł recenzowany, nadesłany 02.12.2013 r., przyjęty do druku 07.05.2014 r.

mgr inż. Grzegorz Madej

Absolwent studiów drugiego stopnia na kierunku Automatyka i Robotyka na Wydziale Mechanicznym Politechniki Krakowskiej.

e-mail: madejgrzegorz@poczta.fm



dr inż. Waldemar Małopolski

Adiunkt w Instytucie Technologii Maszyn i Automatykacji Produkcji Politechniki Krakowskiej. Prowadzi zajęcia z zakresu modelowania, symulacji, optymalizacji i sterowania procesami dyskretnymi.

e-mail: malopolski@mech.pk.edu.pl

