Komunikacja emulatora pracy robotów przemysłowych z oprogramowaniem do symulacji układów automatyki

Andrzej Burghardt, Dariusz Szybicki, Paulina Pietruś

Politechnika Rzeszowska im. Ignacego Łukasiewicza, Katedra Mechaniki Stosowanej i Robotyki, Al. Powstańców Warszawy 8, 35-959 Rzeszów

Streszczenie: W artykule przedstawiono przegląd istniejących rozwiązań do programowania i symulacji pracy robotów przemysłowych. Zaprezentowano informacje dotyczące sterowników PLC, sposobów ich programowania oraz standardu komunikacyjnego OPC. Zbudowano strukturę komunikacji między oprogramowaniem RobotStudio a narzędziem służącym do symulacji układów automatyki – Automation Studio. Przeprowadzono symulację działania oraz testy zaprojektowanego oprogramowania dla przykładowego stanowiska zrobotyzowanego. Połączono wirtualny kontroler robota przemysłowego ze sterownikiem PLC, dla którego program napisano w Automation Studio.

Słowa kluczowe: standard komunikacji OPC, manipulator przemysłowy, komunikacja, sterownik PLC, stanowisko zrobotyzowane

1. Wprowadzenie

Kontrolery robotów mogą komunikować się z zewnętrznymi urządzeniami, np. sterownikami PLC oraz komputerami PC za pomocą modułów wejść/wyjść lub innych standardów komunikacyjnych. Dzięki temu urządzenia mogą sprawować nadzór nad jednostką manipulacyjną robota, wpływać na parametry systemu, komunikować się z systemami zarządzania produkcją itd. Często do komunikacji z kontrolerami robotów wykorzystuje się rodzinę protokołów TCP/IP [2, 3]. Postępująca automatyzacja, upowszechnienie robotów oraz standardów komunikacyjnych rozpowszechnia wdrażanie koncepcji Przemysłu 4.0 [1, 8, 9]. W celu połączenia różnych standardów komunikacyjnych opracowano OPC. Standard ten definiuje sposoby komunikacji między urządzeniami. Pozwala uniezależnić oprogramowanie użytkowe od producentów oprzyrządowania. Do zalet tego typu rozwiązania można zaliczyć: standary-

Rys. 1. Integracja różnych urządzeń i systemów automatyki przed i po wprowadzeniu specyfikacji OPC

Fig. 1. Integration of various devices and automation systems before and after the introduction of the OPC specification

Autor korespondujący:

Paulina Pietruś, p.pietrus@prz.edu.pl

Artykuł recenzowany

nadesłany 14.05.2019 r., przyjęty do druku 28.06.2019 r.

Zezwala się na korzystanie z artykułu na warunkach licencji Creative Commons Uznanie autorstwa 3.0

zację komunikacji oraz wymiany danych, spełnienie warunku skalowalności rozwiązań, obniżkę kosztów integracji dużych systemów [7].

Obecnie występuje kilkanaście rozwiązań zastosowania standardu OPC w przemyśle. Przykłady omówione w dalszej części artykułu prezentują, w jaki sposób można ten standard wykorzystać do usprawnienia funkcjonowania produkcji oraz akwizycji danych. Jednym z przykładów jest wykorzystanie standardu OPC w przemyśle papierniczym. Istotnym elemen-

Rys. 2. Przykładowy system monitorowania stanu środowiska
Fig. 2. Exemplary system for monitoring the state of the environment

tem podczas budowy sytemu sterowania była konieczność uzyskiwania danych z dwóch różnych źródeł. Sterownik PLC marki Siemens zastosowano dla systemu uzdatniania wody, z którego dane trzeba było przekazać do systemu sterowania ABB Advant. Początkowo komunikacja miała wykorzystać sieć Profibus, okazała się jednak kosztowna i czasochłonna. Najlepszym rozwiązaniem dla przedsiębiorstwa okazało się zastosowanie serwera OPC KEPServerEX firmy KEPWare, który obsługuje sterowniki marki Siemens. Na rys. 1 przedstawiono schematycznie komunikację między serwerami OPC [10].

Kolejnym przykładem jest zastosowanie architektury klient-serwer w systemie monitorowania stanu środowiska. System monitorowania zrealizowano wykorzystując program komunikacyjny OPC. Układ komunikacyjny między systemem SCADA a koncentrator oparto na serwerze OPC i kliencie OPC. Natomiast komunikacja między systemem SCADA a koncentrator Echelon typu iLon100 (SmartServer) odbywa się za pomocą sieci Ethernet. Do koncentratora iLon100 podpięto przykładowo zamodelowaną sieć składającą się z trzech czujników: temperatury, wilgotności i oświetlenia oraz sterowników we/wy analogowych i sterowników we/wy dwustanowych, co zaprezentowano na rys. 2 [6]. Przeprowadzone testy funkcjonowania oraz działania systemu potwierdziły przydatność zaprezentowanego rozwiązania.

2. Komunikacja robota ze sterownikiem PLC za pomocą serwera OPC

Specyfikacja standardu OPC bazuje na mechanizmie OLE, który korzysta z technologii COM (ang. *Component Object Model*) tworzenia oraz definiowania interfejsów programistycznych na poziomie binarnym. Standard COM umożliwia efektywną komunikację między aplikacjami. Definiuje komponenty programowe niezależnie od języka programowania, umożliwiając dołączenie do aplikacji fragmentów należących do innych programów.

Poszerzonym standardem COM jest standard interfejsu programistycznego DCOM (ang. *Distributed COM*). Standard DCOM umożliwia przepływ danych za pośrednictwem sieci wykorzystując protokół TCP/IP, podczas gdy COM odnosi się do komunikacji na serwerze lokalnym. DCOM zastępuje protokołem sieciowym komunikację lokalną między procesami, używając technologii DCE RPC (ang. *Distributed Computing Enviromental/Remote Procedure Call*) [4]. Standard OPC opiera się na architekturze klient/serwer (Rys. 3). Klient OPC może pracować jako moduł, który umożliwia aplikacjom np. Excel korzystać z danych OPC.

Rys. 3. Koncepcja architektury OPC [5]
Fig. 3. OPC architecture concept [5]

Rys. 4. Okno konfiguracji OPC DA
Fig. 4. OPC DA configuration window

Rys. 5. Widok okna do połączenia się z serwerem OPC
Fig. 5. Window view to connect to the OPC server

W ramach pracy zdecydowano się skomunikować kontroler IRC5 ze sterownikiem PLC. W zaprezentowanym przypadku kontroler komunikuje się za pomocą standardu OPC DA ze sterownikiem PLC, wykorzystując do komunikacji standard OPC UA. Realizacja połączenia jest możliwa przy użyciu programu Cogent DataHub. Konfiguracja połączenia w programie Cogent DataHub jest możliwa dzięki uprzedniemu przygotowaniu środowisk RobotStudio oraz Automation Studio, które opisano w późniejszych rozdziałach.

W celu nawiązania połączenia z kontrolerem robota należy wybrać w oknie głównym programu zakładkę OPC DA, na ekranie wyświetli się okno zaprezentowane na rys. 4. Po wyświetleniu okna widoczne są aktywne połączenia oraz ich stan, co umożliwia nawiązanie komunikacji z kontrolerem robota za pomocą elementów zakładki Add (Rys. 5).

Po rozwinięciu listy *OPC Server Name* wyświetli się lista dostępnych serwerów. W tym przypadku będzie to serwer ABB IRC5. Po zaznaczeniu wymaganych opcji kontroler robota jest podłączony i dane są gotowe do użycia. Połączenie sterownika PLC przez OPC UA odbywa się w zakładce OPC UA (Rys. 4).

3. Projekt zrobotyzowanej stacji w oprogramowaniu RobotStudio

W ramach artykułu zaprojektowano strukturę komunikacji między oprogramowaniem RobotStudio a narzędziem służącym do symulacji układów automatyki – Automation Studio z wykorzystaniem standardu OPC. Na rysunku 6 przedstawiono schemat zaprojektowanego systemu komunikacji.

W celu weryfikacji działania zaprojektowanego schematu komunikacji między sterownikiem PLC a kontrolerem robota zbudowano przykładowe stanowisko do paletyzacji. Przedmiotem są pudełka kartonowe o wymiarach 390 mm × 250 mm × 250mm. Widok zaprojektowanego stanowiska w środowisku RobotStudio przedstawiono na rys. 7.

Stanowisko wyposażono w manipulator ABB IRB 460, chwytak podciśnieniowy, przenośnik rolkowy oraz przenośnik łańcuchowy.

Rys. 6. Schemat połączenia zbudowanego systemu komunikacji
Fig. 6. The connection diagram of the built-in communication system

Rys. 7. Widok stanowiska do paletyzacji
Fig. 7. View of the palletizing station

Zadaniem przenośnika rolkowego jest transport pudełek do miejsca poboru przez manipulator. Na przenośniku łańcuchowym umieszczane są palety. Zadaniem manipulatora wyposażonego w chwytak podciśnieniowy jest transport pudełek na palety.

4. Budowa oprogramowania za pomocą narzędzi do symulacji układów automatyki

W opisanym w artykule stanowisku, do oprogramowania sterownika PLC zastosowano środowisko developerskie Automation Studio, które służy do konfiguracji oraz programowania komponentów wyprodukowanych przez firmę B&R.

Budowę oprogramowania sterownika w środowisku Automation Studio rozpoczęto od utworzenia nowego projektu oraz wyboru kontrolera. Istotnym etapem jest wybór z toolboxa pliku, w jakim pisany będzie program, tj. język drabinkowy (*Ladder Diagram*), co przedstawiono na rysunku 8.

Kolejnym krokiem było dodanie potrzebnych sygnałów oraz stałych w oknie *Variables*, które są dostępne w sekcji *Program*. Program został utworzony przez dodawanie kolejnych linii oraz uzupełnianie ich odpowiednimi elementami. Po utworzeniu programu zostanie on skompilowany, dzięki czemu można sprawdzić poprawność jego działania.

Następnym etapem była konfiguracja komunikacji zgodnie ze standardem OPC UA. W tym celu w eksploratorze projektu należy wybrać widok fizyczny (*Physical View*). Kolejnym

Rys. 8. Wybór języka programowania
Fig. 8. Selection of programming language

Rys. 9. Widok okna konfiguracji
Fig. 9. View of the configuration window

Rys. 10. Okno wyboru zmiennych do komunikacji
Fig. 10. Selection window for variables for communication

Rys. 11. Pobór pudełek przez manipulator
Fig. 11. Collection of boxes by the manipulator

crokiem jest aktywacja komunikacji w zakładce *OPC-UA System*. W opcji *Network Settings* należy wpisać port potrzebny do komunikacji. Po ustaleniu wszystkich opcji wybieramy widok konfiguracyjny (*Configuration View*) przedstawiony na rys. 9.

W tej sekcji rozwinęto folder modelu sterownika (X20_CP1584) i w zakładce *Connectivity* wybrano *OPC UA Default View File*, co pozwoli na udostępnienie zmiennych. W utworzonym programie wybiera się odpowiednie *Tagi* dostępne w menu *Enable Tags*, co zaprezentowano na rys. 10.

Po wybraniu odpowiednich zmiennych potrzebnych do komunikacji należy skompilować projekt i wgrać go do sterownika. Proces konfiguracji komunikacji za pomocą standardu OPC w środowisku Automation Studio został zakończony.

5. Symulacja i testy wykonanego oprogramowania

Weryfikacja poprawności komunikacji między sterownikiem PLC a kontrolerem robota została zrealizowana na zbudowanym przykładowym stanowisku do paletyzacji pudełek w programie RobotStudio (Rys. 7). W tym celu utworzono odpowiednie oprogramowanie w środowisku Automation Studio, z którym połączono odpowiednie sygnały za pomocą programu Cogent DataHub w standardzie OPC.

Całym procesem paletyzacji steruje sterownik PLC, który zarządza odpowiednimi sygnałami wejścia/wyjścia. Wybór polecenia *Start* na panelu operatorskim uruchamia proces produkcyjny. W zaprezentowanym rozwiązaniu pudełka transportowane

są na przenośniku rolkowym. Jednocześnie odbywa się proces poboru palety z magazynu za pomocą manipulatora oraz transport na przenośnik łańcuchowy.

Zliczanie pudełek realizowane jest dzięki zastosowaniu bramki optycznej. Po przejściu przez bramkę trzech pudeł, przenośnik zatrzymuje się i następuje przygotowanie pudeł do paletyzacji. Odpowiednie ustawienie pudeł umożliwia siłownik pneumatyczny, który przesuwają je do miejsca poboru przez robota. Na rysunku 11 przedstawiono manipulator, który czeka w pozycji startowej do chwili, kiedy pudełka zostaną ułożone w odpowiedniej konfiguracji.

Kolejnym etapem jest transport pudeł na paletę, który realizowany jest przez robota wyposażonego w chwytak podciśnieniowy. Algorytm programu tak opracowano, aby po wypełnieniu drugiej warstwy na palecie został wysłany sygnał, który informuje o tym fakcie (Rys. 12).

Całym cyklem procesu steruje sterownik PLC firmy B&R, który zgodnie z programem steruje odpowiednimi wejściami/wyjściami. Fragment kodu programu w języku drabinkowym sterownika PLC pokazano na rys. 13.

6. Podsumowanie i wnioski

W artykule omówiono opracowany system komunikacji kontrolera robota ze sterownikiem PLC, w którym wykorzystano standard OPC. Zaprojektowano oraz zbudowano wirtualne zrobotyzowane stanowisko realizujące proces paletyzacji w środowisku RobotStudio.

Rys. 12. Widok zrobotyzowanej stacji

Fig. 12. View of robotic station

Zbudowano strukturę komunikacji między oprogramowaniem RobotStudio a narzędziem do budowy i symulacji układów automatyki – Automation Studio z wykorzystaniem standardu OPC za pomocą programu Cogent DataHub. Program ten umożliwia połączenie odpowiednich sygnałów oraz ich transmisję i gromadzenie danych w czasie rzeczywistym. Za sterowanie nadrzędne w zaprezentowanym przykładzie odpowiada sterownik PLC, który zarządza odpowiednimi sygnałami wejścia/wyjścia.

Opracowany system komunikacji można zastosować do symulacji innych układów automatyki, jak moduły bezpieczeństwa, napędy, sterowniki czy panele HMI. Standard OPC umożliwia także zbieranie i analizę danych, co może być istotne podczas kontroli przebiegu procesu produkcyjnego czy technologicznego.

Bibliografia

1. Badurek J., *Systemy ERP dla wytwórczości nowej generacji*. „Przedsiębiorstwo we współczesnej gospodarce – teoria i praktyka”, Nr 2, 2014, 79–90.
2. Burghardt A., Kurc K., Szybicki D., *Robotic automation of the turbo-propeller engine blade grinding process*. “Applied Mechanics & Materials”, Vol. 817, 2016, 206–213, DOI: 10.4028/www.scientific.net/AMM.817.206.
3. Kaczmarek W., Panasiuk J., Gospodarczyk K., *Wykorzystanie robota przemysłowego do symulacji ruchu obiektu w badaniu optoelektronicznych głowic śledzących*. „Mechanik”, R. 86, Nr 7CD, 2013, 287–296.
4. Kwaśniewski J., *Sterowniki PLC w praktyce inżynierskiej*, Wydawnictwo btc, Legionowo 2008.
5. Szulewski P., *Wykorzystanie technologii OPC do udostępniania danych pomiarowych z procesu i obrabiarki*. „Mechanik”, R. 88, Nr 8-9CD2, 2015, 614–622, DOI: 10.17814/mechanik.2015.8-9.474.
6. Tomczyk A., *Wykorzystanie architektury klient-server w systemach monitorowania stanu środowiska*, „Mechanizacja i Automatykacja Górnicza”, Nr 48, 2010, 133–136.
7. [www.asimo.pl/teoria/OPC.php].
8. Szulewski P., *Koncepcje automatyki przemysłowej w środowisku Industry 4.0*. „Mechanik”, R. 89, Nr 7, 2016, 574–578, DOI: 10.17814/mechanik.2016.7.221.
9. Wittbrodt P., Łapuńska I., *Przemysł 4.0 – Wyzwanie dla współczesnych przedsiębiorstw produkcyjnych*. „Innowacje w Zarządzaniu i Inżynierii Produkcji”, Nr 2, 793–799.
10. [https://eia.pg.edu.pl/documents/1113028/0/PSI%20OPC%20%28OLE%20for%20Process%20Control%29%20Zastosowania_Wyklad.pdf] pobrano 07.06.2018 r.

Rys. 13. Fragment programu sterownika PLC

Fig. 13. A fragment of the PLC program

Communication of Industrial Robot Emulator with Software for Simulation of Automation Systems

Abstract: The article presents an overview of existing solutions for programming and simulating the work of industrial robots. Information on PLCs, their programming methods and the OPC communication standard has been presented. A communication structure between the RobotStudio software and a tool for simulation of automation systems – Automation Studio was built. Simulated operation and tests of designed software for an exemplary robotic station were carried out. The virtual controller of the industrial robot was connected to the PLC for which the program was written in Automation Studio.

Keywords: OPC communication standard, industrial manipulator, communication, PLC controller, robotic station

dr hab. inż. Andrzej Burghardt, prof. PRz

andrzejb@prz.edu.pl

Absolwent AGH w Krakowie (2001 r.) Wydziału Inżynierii Mechanicznej i Robotyki – specjalności Robotyka i Mechatronika oraz Wydziału Górniczego – specjalność Marketing i Zarządzanie w Przemysle. Stopień doktora nauk technicznych w dyscyplinie budowa i eksploatacja maszyn uzyskał w 2005 r. na Wydziale Budowy Maszyn i Lotnictwa Politechniki Rzeszowskiej, na tym samym wydziale w 2014 r. uzyskał stopień doktora habilitowanego nauk technicznych w dyscyplinie mechanika. Zajmuje się problematyką szeroko pojętej robotyki dotyczącej m.in. zagadnień prototypowania, sztucznej inteligencji, implementacji układów sterowania mobilnych robotów kołowych, planowania trajektorii, sterowania behawioralnego, programowania robotów przemysłowych oraz systemów wizyjnych.

dr inż. Dariusz Szybicki

dszybicki@prz.edu.pl

Urodził się w Przeworsku. Studia wyższe ukończył na Politechnice Rzeszowskiej im. Ignacego Łukasiewicza. W latach 2009–2013 był uczestnikiem studiów doktoranckich z zakresu Mechaniki na Wydziale Budowy Maszyn i Lotnictwa Politechniki Rzeszowskiej. W 2014 r. obronił pracę doktorską w dyscyplinie Mechanika. Od 2014 r. jest zatrudniony na stanowisku adiunkta w Katedrze Mechaniki Stosowanej i Robotyki. Jego zainteresowania naukowe dotyczą szeroko pojętej mechatroniki. W ramach projektów badawczych zajmuje się projektowaniem stacji zrobotyzowanych oraz programowaniem robotów przemysłowych.

mgr inż. Paulina Pietruś

p.pietrus@prz.edu.pl

Urodziła się w Rzeszowie. Studia wyższe ukończyła na Politechnice Rzeszowskiej im. Ignacego Łukasiewicza. W roku akademickim 2016/2017 podjęła studia doktoranckie na Wydziale Budowy Maszyn i Lotnictwa Politechniki Rzeszowskiej. W tym samym roku rozpoczęła pracę w Katedrze Mechaniki Stosowanej i Robotyki, gdzie obecnie jest asystentem. Jej zainteresowania naukowe – szeroko pojęta mechatronika, programowanie robotów przemysłowych.