Dydaktyka Informatyki 16(2021)
ISSN 2083-3156; e-ISSN 2543-9847
DOI: 10.15584/di.2021.16.18
http://www.di.univ.rzeszow.pl

Uniwersytet Rzeszowski

Jacek BARTMAN[1], **Dariusz SOBCZYŃSKI**[2]

[1] *ORCID 0000-0001-7372-2029. Dr inż., Uniwersytet Rzeszowski, Kolegium Nauk Przyrodniczych, Centrum Dydaktyczne Nauk Techniczno-Przyrodniczych, ul. Pigonia 1, 35-310 Rzeszów; e-mail: jbartman@ur.edu.pl*
[2] *ORCID 0000-0002-6379-9556. Dr inż., Politechnika Rzeszowska, Wydział Elektrotechniki i Informatyki, ul. W. Pola 2, 59-959 Rzeszów; e-mail: dsobczyn@prz.edu.pl*

CODESYS – UNIWERSALNE NARZĘDZIE DO PROGRAMOWANIA STEROWNIKÓW PLC

CODESYS – THE UNIVERSAL TOOL FOR PLC PROGRAMMING

Słowa kluczowe: sterownik PLC**,** programowanie, symulacja, wizualizacja, IEC-61131-3.
Keywords: controller, programming, simulation, visualization, IEC-61131-3.

Streszczenie

W artykule przedstawiono środowisko CoDeSys jako uniwersalne narzędzie do programowania sterowników PLC zgodne z normą IEC-61131-3. We wstępie do pracy zostały zdefiniowane oczekiwania, jakie należy postawić środowisku do programowania sterowników PLC. Następnie scharakteryzowano CoDeSys, podkreślając jego niezależność od producenta sterownika oraz możliwość pisania, symulowania oraz wizualizacji działania programów bez potrzeby posiadania fizycznego urządzenia. W dalszej części skrótowo omówiono, w jaki sposób przygotować program oraz zasymulować i zwizualizować jego pracę. Podsumowując wskazano na niezwykle aktualną w dobie pandemii COVID cechę środowiska CoDeSys, jakim jest możliwość łatwego wykorzystania go do zdalnego nauczania programowania sterowników PLC.

Abstract

The paper presents CoDeSys as a universal tool for PLC programming in compliance with the IEC-61131-3 standard. In the introduction to the paper, the expectations that should be placed on the PLC programming environment were defined. Then CoDeSys has been characterized, emphasizing its independence from the PLC manufacturer and the possibility of writing, simulating and visualizing programs without the need to own a physical device. The next part briefly discusses how to prepare the program and how to simulate and visualize its operation. In conclusion, an extremely current in the era of the COVID pandemic feature of the CoDeSys environment was pointed out, which is the possibility of its easy use for remote teaching of PLC programming.

Wstęp

Nieustanny rozwój systemów automatyki skutkuje wzrostem zapotrzebowania na wykwalifikowanych specjalistów w zakresie programowania sterowników PLC. Nauczanie programowania sterowników PLC prowadzone jest na wielu kierunkach studiów; poświęcone są mu również prace naukowe[1]. Dużym wyzwaniem w zakresie kształcenia programistów PLC jest powiązanie aplikacji programistycznych ze sprzętem – poszczególni producenci sterowników PLC oferują autorskie aplikacje do ich programowania. Kształcenie na oprogramowaniu konkretnego producenta powoduje, że przekazywana/uzyskiwana wiedza i kompetencje nie mają w pełni uniwersalnego charakteru. Dzięki standaryzacji języków programowania opisanej w najnowszej normie IEC 61131-3[2] otwarła się furtka umożliwiająca programowanie sterowników PLC niezależnie od producenta sterownika – za pomocą tego samego standardowego języka, zaś programy tworzone w takim ustandaryzowanym języku mogą być łatwo przeniesione z jednego kompatybilnego systemu sterowania do innego. Innym wyzwaniem jest bariera sprzętowa. Nie każdy, kto chce się uczyć programowania sterowników PLC posiada do nich dostęp.

Wymogi stawiane oprogramowaniu do nauki programowania sterowników PLC:

- uniwersalność (niepowiązane z pojedynczymi producentami sterowników) oraz zgodne z normą IEC 61161-3,
- powinno umożliwić symulację pracy sterownika tak, aby nie było konieczności posiadania fizycznego urządzenia, w czasach powszechnego nauczania zdalnego wymóg ten nabiera szczególnego znaczenia,
- powinno umożliwiać wizualizację sterowanego procesu, co ułatwia testowanie oprogramowania oraz ocenę poprawności jego działania;
- dostępna powinna być pełna dokumentacja opisująca funkcjonalność oprogramowania,
- bardzo pożądane jest, aby oprogramowanie było dostępne w formie nieodpłatnej; likwiduje to barierę finansową,
- wskazane jest, aby oprogramowanie było dostępne w języku polskim, gdyż ułatwia to posługiwanie sie nim.

---

[1] A.A. Gusarova, S.V. Shilkina, *Modeling the Operation of the System in the CODESYS Software Environmentper*, International Science and Technology Conference "EastConf", 2019, p. 1–6. DOI: 10.1109/EastConf.2019.8725346; S. He, H. Rahemi, K. Mouaouya, *Teaching PLC Programming and Industrial Automation in Mechatronics Engineering*, ASEE Annual Conference & Exposition, 2015. DOI: 10.18260/p.24820.

[2] IEC 61131-3:2013 Programmable controllers – Part 3: Programming languages

W artykule zaprezentowano środowisko CoDeSys, które należy do grupy, dostępnych na rynku aplikacji, służących do programowania sterowników PLC, spełniających prawie wszystkie wymienione wymogi (brak jedynie wersji w języku polskim)[3]. Przedstawiono proces budowy projektu obsługi procesu sterowania zawierającego programowanie sterownika, symulację i wizualizację sterowania.

Według Garego Pratta[4] platformy zgodne z normą IEC 61131-3/PL Copen są wykorzystywane przez ponad 350 producentów wyposażenia oryginalnego.

W kolejnych rozdziały pracy przedstawiono: zestawienie języków programowania dostępnych w CoDeSys, proces tworzenia programu na sterownik PLC, symulację pracy sterownika oraz wizualizację sterowanego procesu w środowisku CoDeSys.

CoDeSys i języki programowania PLC – krótka charakterystyka

CoDeSys to niezależne od sprzętu oprogramowanie firmy 3S, umożliwiające programowanie i symulacje pracy sterowników PLC oraz wizualizację sterowanego procesu. Nazwa CoDeSys jest skrótem od *Co*ntroller *De*velopment *Sys*tem.

Platforma CoDeSys jest zgodna z międzynarodową normą IEC 61131-3 i implementuje wszystkie, wymienione we wspomnianej normie, języki programowania sterowników PLC:

- język drabinkowy LD (*Lader Diagram*), który jest doskonałym narzędziem do opisu prostej logiki dyskretnej, stosowanej w układach przekaźnikowych i czasowych,
- języki bloków funkcyjnych FDB (*Function Block Diagram*) oraz CFC (*Continuous Function Chart*). Język CFC to doskonałe narzędzie programistyczne do umieszczania i wzajemnego łączenia wbudowanych, gotowych lub zbudowanych przez użytkownika, bloków funkcyjnych. Stanowi on rozszerzenie, zaproponowane przez normę IEC 61131-3, które usuwa ograniczenia dotyczące sieci i kolejności wykonywania występujące FBD,
- język strukturalny ST (*Structured Text*), który jest język językiem najbliższym językom służącym do programowania komputerów i dobrze nadaje się do tworzenia iteracji, manipulacji bitami itp.
- lista instrukcji IL (*Instruction List*), język na wzór asemblerów stosowanych do programowania klasycznych komputerów,
- sekwencyjny język graficzny SFC (*Sequential Function Chart*), który został stworzonym do opisu operacji sekwencyjnych lub zależnych od stanu.

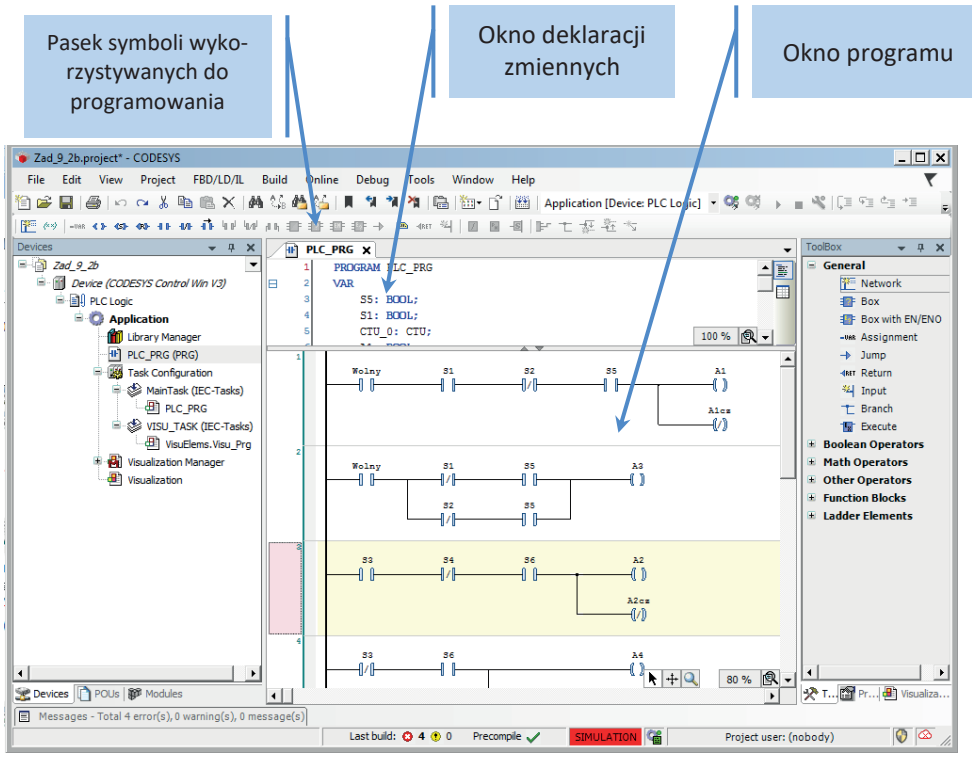
[3] help.codesys.com

[4] G. Pratt, *Standaryzacja programowania przemysłowych systemów sterowania dzięki językom opisanym w normie IEC 61131-3*, „Control Engineering" 2018, nr 6, s. 82–87.

Środowisko CoDeSys umożliwia wybór platformy procesora, systemu operacyjnego do kontrolera, a także typu i producenta urządzenia. Zainstalowanie tzw. targetów, czyli plików sprzętowych, które pozwalają środowisku CoDeSys na pracę z danym sterownikiem sprawia, że środowisko, jest kompatybilne z konkretnym sterownikiem PLC, posiadającym określoną konfigurację[5].

Programowanie w CoDeSys

Pracę z CoDeSys musimy rozpocząć od ściągnięcia programu instalującego ze strony producenta https://store.codesys.com. Wymaga to określenia profilu naszej działalności (biznesowy bądź prywatny) oraz zarejestrowania się na stronie. Sama instalacja przebiega bardzo prosto.



Rys. 1. Widok okna CoDeSys w trybie edycji programu w języku LD

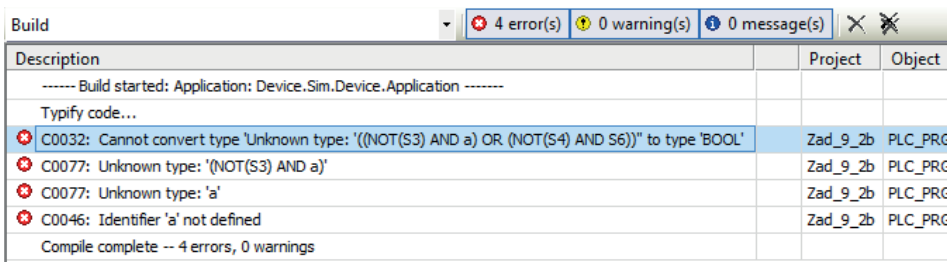
Źródło: opracowanie własne.

[5] D.H. Hanssen, *Programmable Logic Controllers: A Practical Approach to IEC 61131-3 using CoDeSys*, Willey 2015, s. 416.

Obsługa programu jest bardzo intuicyjna, a sam interfejs czytelny. Ponadto oprócz standardowej pomocy, jaką możemy uzyskać wybierając menu *Help,* dostępna jest na stronie help.codesys.com bardzo szczegółowa instrukcja online.

Pracę nad nowym programem rozpoczynamy od stworzenia nowego projektu (*File→New Project*). Następnie należy podać jego nazwę oraz lokalizację, dodatkowo musimy wybrać, czy chcemy, aby został utworzony pusty projekt (*Empty Project*) czy też projekt standardowy (*Standard Project*). W przypadku wyboru projektu pustego, konieczne jest dodanie wszystkich niezbędnych komponentów „ręcznie", co pozwala lepiej poznać strukturę drzewa projektu. W przypadku projektu standardowego wszystkie niezbędne elementy automatycznie zostaną umieszczone w drzewie projektu, my musimy tylko wybrać urządzenie oraz język programowania.

Po stworzeniu projektu i kliknięciu na nasz program otworzy nam się okno do edycji (rys. 1), podzielone na dwie sekcje: górną do deklaracji zmiennych i dolną do pisania programu; obie części okna można skalować niezależnie. Podczas pisania programu bardzo pomocna jest opcja automatycznego podpowiadania typu zmiennej, którą po raz pierwszy używamy w programie. Podstawowe elementy wykorzystywane do edycji programu pojawiają się kontekstowo na pasku Menu.



Rys. 2. Widok okna informującego o wynikach kompilacji programu

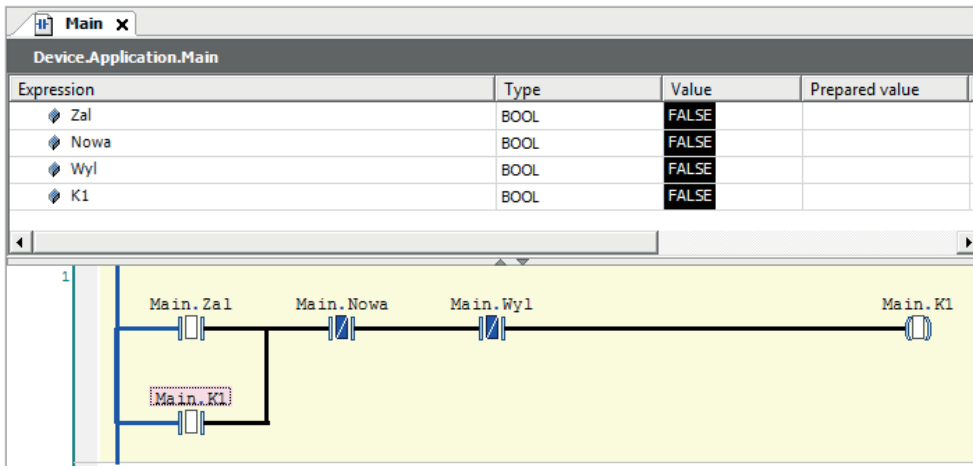
Źródło: opracowanie własne.

Gotowy program należy skompilować; w tym celu wybieramy *Build→Build*, możemy również skorzystać ze stosownej ikonki w *Menu* programu lub nacisnąć klawisz *F11*. W wyniku wykonania tej operacji pojawi się informacja o wynikach kompilacji, jeżeli wystąpiły błędy, to zostają one wskazane. Na rys. 2 pokazano kompilację programu, w którym nie została zadeklarowana zmienna *a*. Jak widać na rys. 2, pomimo iż jest tylko jeden błąd, system wygenerował komunikat informujący o czterech błędach. Najczęściej jednak analiza opisów pozwala szybko i jednoznacznie określić miejsce i rodzaj błędu.

Symulacja pracy sterownika

Symulacja pracy sterownika pozwala na sprawdzenie, czy stworzony program wykonuje wszystkie operacje zgodnie z naszymi oczekiwaniami. Jest to bardzo ważna funkcjonalność programu umożliwiająca realizację programu bez fizycznego urządzenia sterującego (bez sterownika PLC). Symulację załączamy wybierając *Online→Simulation*, następnie logujemy się do sterownika wirtualnego *Online→Login,* możemy to też zrobić wykorzystując skrót klawiszowy *Alt-F8* lub klikając odpowiednią ikonę w pasku *Menu*. Po zalogowaniu się do sterownika system automatycznie sprawdza, czy aplikacja, nad którą pracujemy, znajduje się w pamięci sterownika. Jeżeli nie, to wykonuje jej kompilację i w przypadku pomyślnego jej zakończenia przesyła program do sterownika-symulatora. Jednocześnie okno deklaracji zmiennych oraz okno programu zmieniają swój wygląd (rys. 3).

W oknie deklaracji obok nazwy zmiennej pojawia sie jej aktualna wartość (kolumna *Value* na rys. 3) oraz miejsce na wpisanie nowej, proponowanej wartości (kolumna *Propared value* na rys. 3), dzięki czemu uzyskujemy możliwość symulowania zmiany sygnałów wejściowych sterownika. Aby zaobserwować, jak sterownik reaguje na zmiany wejść, uruchamiamy go, wpisujemy nową wartość wejścia i akceptujemy ją wybierając *Debug→Write Values* (*Ctrl+F7*). W efekcie zmieniają się wartości zmiennych wyjściowych widocznych w oknie deklaracji.



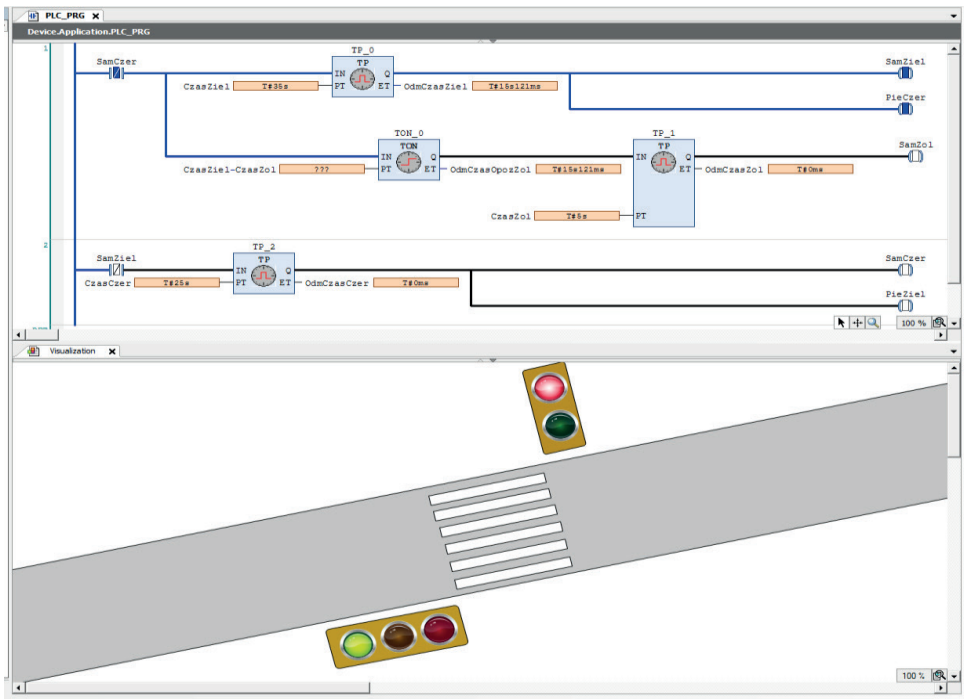
Rys. 3. Widok okna deklaracji oraz okna programu CoDeSys podczas symulacji pracy sterownika

Źródło: opracowanie własne.

W oknie programu styki i linie je łączące przyjmują kolor niebieski, gdy są w stanie wysokim (1) lub pozostają czarne, gdy są w stanie niskim (rys. 3). Zmiana sygnałów wejściowych w oknie deklaracji powoduje odpowiednią reakcję w oknie programu, co pozwala obserwować, czy sterowanie jest realizowane właściwie.

Wizualizacja pracy sterownika

Symulacja jest bardzo pomocna, posiada jednak pewne wady. Zmiany wartości sygnałów wejściowych są realizowane w mało wygodny sposób, nie można rozróżnić styków astabilnych od bistabilnych, analiza działania większego programu napisanego w języku LD jest bardzo uciążliwa. Najczęściej kod programu nie mieści się na jednym ekranie i w celu przeanalizowania jego działania konieczne jest przewijanie treści; zajmuje to czas i utrudnia analizę. W takiej sytuacji bardzo pomocna jest wizualizacja działania programu.



Rys.4. Widok okien programu CoDeSys podczas symulacji pracy sterownika z wizualizacją

Źródło: opracowanie własne.

W celu utworzenia wizualizacji wybieramy w drzewie projektu menu podręczne gałęzi *Application,* a następnie *Add Object→Visualization*; powoduje to otwarcie w miejsce okna deklaracji i okna programu okna wizualizacji oraz wyświetlenie w prawym panelu okna aplikacji narzędzi nazwanego *Visualization Toolbox*. W szufladach toolboxa znajdują się pogrupowane narzędzia do wizualizacji.

Tworzenie wizualizacji przypomina nieco pracę z programem graficznym lub symulatorami obwodów elektrycznych/elektronicznych. Z odpowiednich szyflad-zakładek toolboxa przeciągamy potrzebne elementy do okna wizualizacji. Następnie elementy należy skonfigurować. Możemy to uczynić wykorzystując okno *Properties,* do którego dostęp uzyskujemy po kliknięciu na element. W zależności od przeznaczenia elementu ustawiamy różne parametry. W szczególności może to być funkcjonalność elementu, nazwa przypisanej zmiennej, kolor, opis itp. W wizualizacji możemy również wykorzystywać schematy czy obrazy zaimportowane z zewnątrz i umieszczać na nich elementy wykorzystywane w procesie sterowania.

Uruchomienie wizualizacji odbywa się automatycznie wraz z uruchomieniem symulacji pracy sterownika. Odpowiednio ustawiając okna mamy możliwość jednoczesnego obserwowania wizualizacji oraz zmian sygnałów w oknie symulacji (rys. 4).

Podsumowanie

W pracy pokazano możliwości środowiska CoDeSys jako narzędzia do nauczania programowania sterowników PLC. Narzędzia, które doskonale sprawdzi się w nauczaniu zdalnym, gdyż nie wymaga posiadania fizycznego sterownika oraz jest wraz z instrukcją dostępne bezpłatnie. W tym celu pokazano, w jaki sposób można przygotować program na sterownik PLC, zasymulować jego pracę oraz przedstawić wizualizację procesu sterującego, nie dysponując fizycznym sterownikiem, a wykorzystując jedynie środowisko CoDeSys.

Należy jednak podkreślić, że oprogramowanie nie ma wymiaru tylko dydaktycznego, ale jest również doskonałym narzędziem do wykorzystania, przez specjalistów z zakresu automatyzacji systemów sterowania, do zarządzania procesami technologicznymi i produkcją.

Co istotne, zarówno z punktu widzenia dydaktycznego, jak i inżynierskiego, środowisko CoDeSys umożliwia opracowanie, uruchomienie oraz wizualizację zautomatyzowanego systemu sterowania. Program stworzony dla konkretnego procesu w CoDeSys można łatwo zweryfikować za pomocą modelowania gra-

ficznego oraz dostosować jego funkcjonowanie bezpośrednio w procesie eksploatacji, z uwzględnieniem wszelakich zmian w nim zachodzących.

Najważniejsze jednak jest, aby zrozumieć, jakie to narzędzie i jakie olbrzymie daje możliwości.

Bibliografia

Gusarova A.A., Shilkina S.V., *Modeling the Operation of the System in the CODESYS Software Environmentper*, International Science and Technology Conference "EastConf", 2019. DOI: 10.1109/EastConf.2019.8725346.

He S., Rahemi H., Mouaouya K., *Teaching PLC Programming and Industrial Automation in Mechatronics Engineering*, ASEE Annual Conference & Exposition, 2015. DOI: 10.18260/p.24820.

IEC 61131-3:2013 Programmable controllers – Part 3: Programming languages

Hanssen D.H., *Programmable Logic Controllers: A Practical Approach to IEC 61131-3 using CoDeSys*, Willey 2015.

Pratt G., *Standaryzacja programowania przemysłowych systemów sterowania dzięki językom opisanym w normie IEC 61131-3*, „Control Engineering" 2018, nr 6.

Netografia

help.codesys.com