

Uzupełnienia środowiska inżynierskiego CPDev dla programowania holenderskiego systemu sterowania statków Mega-Guard

Leszek Trybus, Marcin Jamro, Dariusz Rzońca, Jan Sadolewski, Andrzej Stec, Zbigniew Świder, Bartosz Trybus

Wprowadzenie

Pierwsza wersja środowiska inżynierskiego CPDev (*Control Program Developer*), przeznaczonego do programowania sterowników PLC/PAC zgodnie z normą IEC 61131-3 (także PN-EN), powstała w latach 2006/2007. Do programowania służył język ST, najbardziej uniwersalny spośród języków normy, którym posługują się jednak tylko odpowiednio przygotowani programiści. Pierwszym zastosowaniem CPDeva był rozproszony system kontrolno-pomiarowy ze sterownikiem SMC, którego produkcję podjęły Zakłady Lumel SA w Zielonej Górze. Postępy prac nad CPDevem przedstawiano m. in. na Konferencjach Automatyków w Rytrze [1, 2].

Trzy lata temu kontakt z autorami w Politechnice Rzeszowskiej nawiązała firma Praxis Automation Technology B. V. z Lejdy (teraz w Leidendorp), produkująca system Mega-Guard do automatyzacji i nawigacji statków (*Ship Automation and Navigation System*). W wyniku zawartego kontraktu CPDev trafił do Praxis i obecnie niektóre podsystemy Mega-Guarda już z niego korzystają. Wzajemne kontakty rozwijały się jednak nadal i niedawno PRz i Praxis zawarły trwałą umowę o współpracy nad rozwojem oprogramowania, której celem są nowe funkcjonalności środowiska CPDev implementowane w Mega-Guardzie.

W niniejszym artykule przedstawiono wstępne wyniki prac prowadzonych w tym kierunku, a także innych prac zmierzających do rozwoju CPDeva. W szczególności chodzi o programowanie w językach graficznych, o ułatwienie uruchamiania programów przez wprowadzanie pułapek oraz o narzędzie wizualizacyjne zintegrowane z CPDevem, przeznaczone dla sterowników z panelem HMI. Najpierw jednak scharakteryzujemy krótko podstawowe cechy i wdrożenia CPDeva oraz urządzenia i podsystemy Mega-Guarda.

Środowisko inżynierskie CPDev

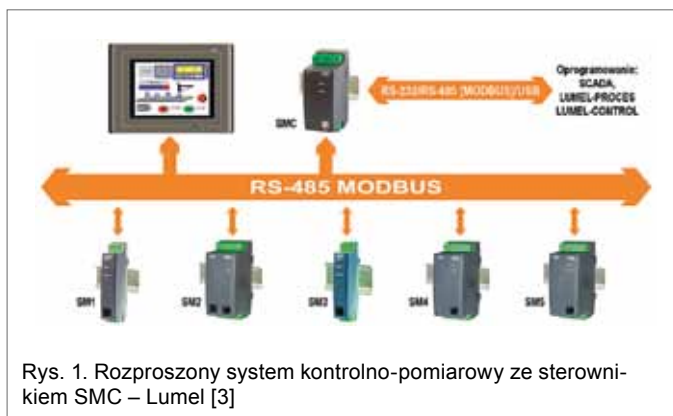
Z założenia środowisko CPDev jest uniwersalne w sensie przeznaczenia dla różnych platform sprzętowych oraz otwarte zarówno dla konstruktorów sterowników, jak i projektantów oprogramowania. Uniwersalność w sensie platform sprzętowych wymaga, aby kompilator tłumaczył program użytkownika na pewien kod pośredni, który po stronie sterownika wykonuje maszyna wirtualna *runtime* właściwa dla konkretnego procesora oraz interfejsów sterownika (będąca firmwarem). Otwartość dla konstruktorów oznacza, że do podstawowego

Streszczenie: Środowisko inżynierskie CPDev jest pakietem przeznaczonych do programowania, symulacji offline, konfiguracji komunikacji oraz uruchamiania online (*commissioning*) sterowników w rozproszonych systemach automatyki i pomiarów. Programować można w tekstowym języku ST. Ze środowiska CPDev korzystają obecnie Zakłady LUMEL w systemie kontrolno-pomiarowym ze sterownikiem SMC oraz holenderska firma Praxis w systemie Mega-Guard do automatyzacji i nawigacji statków. Celem niniejszego referatu jest prezentacja wstępnych wyników prac zmierzających do rozwoju środowiska CPDev, głównie dla systemu Mega-Guard. Przedstawiono nowe edytory języków graficznych FBD, LD i SFC, których diagramy podlegają translacji do bazowego języka ST, a potem kompilacji do kodu wykonywalnego. Symulacja offline jest rozszerzona o mechanizm pułapek ułatwiających znalezienie usterek. Przystąpiono do uzupełnienia CPDeva o narzędzie wizualizacyjne, które będzie potrzebne w sterownikach z własnymi panelami HMI.

EXTENSIONS OF CPDEV ENGINEERING ENVIRONMENT FOR PROGRAMMING DUTCH MEGA-GUARD SHIP CONTROL SYSTEM

Abstract: CPDev engineering environment is a software package for programming, simulation, communication configuration and commissioning of controllers in distributed systems. Programs can be written in ST textual language. The CPDev environment is currently used by LUMEL Zielona Góra in systems with SMC programmable controller and by Praxis company from the Netherlands for Mega-Guard ship automation and navigation system. This paper describes preliminary results on CPDev extensions, mainly for the Mega-Guard. New editors of FBD, LD and SFC graphical languages are presented, whose diagrams are translated first to ST, and then compiled into executable code. Simulation is provided with a breakpoint mechanism to facilitate debugging. Development of visualization tool integrated with CPDev for controllers equipped with HMI panels has begun.

kodu maszyny wirtualnej mogą oni dołączać własne niskopoziomowe procedury obsługi interfejsów – portów wejścia/wyjścia, kanałów komunikacyjnych, zegara RTC itp. Z otwartością w sensie programowym wiąże się możliwość tworzenia



przez użytkownika własnych bibliotek z funkcjami, blokami i programami.

Na oryginalną wersję pakietu CPDev składają się [1, 2]:

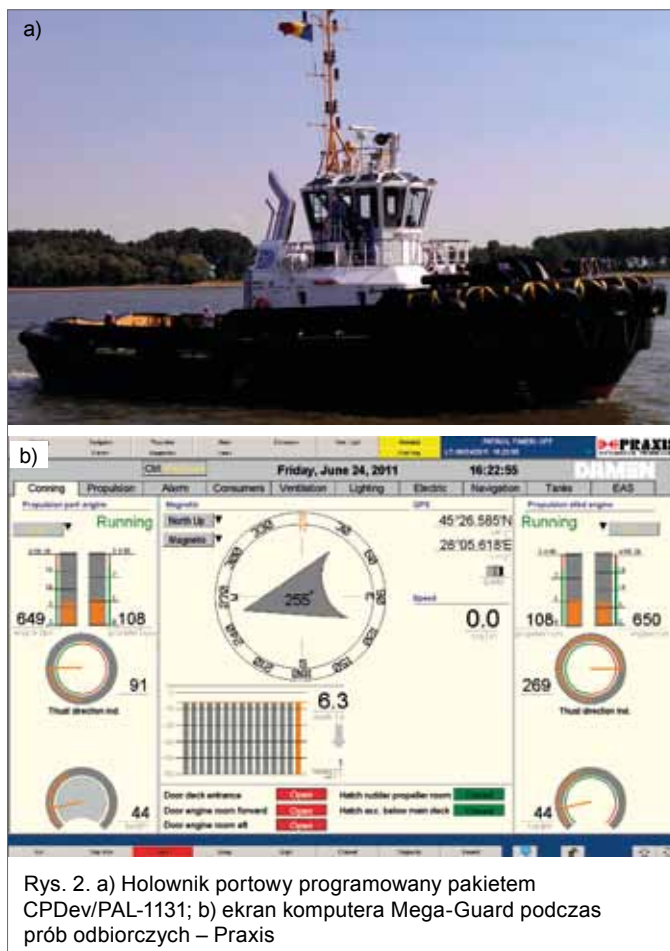
- kompilator CPDev języka ST;
- symulator programów CPSim (offline i online);
- konfigurator komunikacji CPCon;
- podstawowa maszyna wirtualna VM (*runtime*).

Maszyna VM (*Virtual Machine*) jest programem w języku C nieobejmującym procedur obsługi interfejsów, możliwym do uruchomienia na każdym procesorze. Naturalnie jest ona zasadniczą częścią symulatora CPSim (offline).

System ze sterownikiem SMC. Obecną strukturę systemu produkowanego przez zielonogórski Lumel pokazano na rys. 1 [3]. Sterownik SMC nie ma własnych wejść/wyjść, ale korzysta z rozproszonych modułów I/O typu SM1 do SM5, z którymi komunikuje się Modbusem. Do konfiguracji komunikacji służy program CPCon. SMC zawiera 8-bitowy mikrokontroler AVR Atmega. Maszyna wirtualna VM-SMC powstała w wyniku konsolidacji kodu maszyny podstawowej VM z niskopoziomowymi procedurami dostarczonymi przez inżynierów Lumelu.

System Mega-Guard. Składa się z kilkunastu podsystemów o różnym przeznaczeniu, zawierających sterowniki, moduły I/O, stacje operatorskie i panele HMI połączone redundowanym Ethernetem (zob. następny punkt). Do programowania służył dotychczas język PAL (*Praxis Automation Language*). Przejście na CPDev wymagało od inżynierów Praxis zaadaptowania maszyny wirtualnej VM dla 32-bitowego mikrokontrolera ARM LPC, w który wyposażone są sterowniki i moduły I/O, oraz uzupełnienia jej o procedury obsługi interfejsów, w tym stos ethernetowy. Nieco zmodyfikowane środowisko CPDev z maszyną VM-ARM otrzymało nazwę PAL-1131 i od roku obejmuje stopniowo kolejne podsystemy Mega-Guarda. Na rys. 2 a pokazano jedno z pierwszych wdrożeń – duży holownik portowy typu ASD TUG 2810, a na rys. 2 b jeden z ekranów stacji operatorskiej na mostku kapitańskiego podczas prób odbiorczych 24 czerwca 2011 r.

Do oryginalnej wersji CPDeva wprowadzono w międzyczasie zmienne typu DREAL (podwójna precyzja) dla dokładnej reprezentacji współrzędnych GPS, protokół komunikacyjny NMEA, będący standardem dla urządzeń nawigacyjnych, oraz sporo drobnych rozszerzeń i usprawnień. Dodano także kompilator drugiego języka tekstowego IL. Na rys. 3 pokazano demonstracyjny zestaw urządzeń Mega-Guard służący w PRz do testowania.



Urządzenia i podsystemy Mega-Guard

Mega-Guard składa się z 12 podsystemów mogących funkcjonować jako zintegrowany system lub samodzielnie [4]. Urządzeniami systemu są komputery Marine PC, rozproszone jednostki sterujące (*Distributed Processing Units*), panele operatorskie i sieć komunikacyjna.

Urządzenia

Marine PC. Jest to komputer z płytą główną Intel, dyskiem stałym (*solid state disk* – pamięć flash kilka GB) i systemem

a)



b)



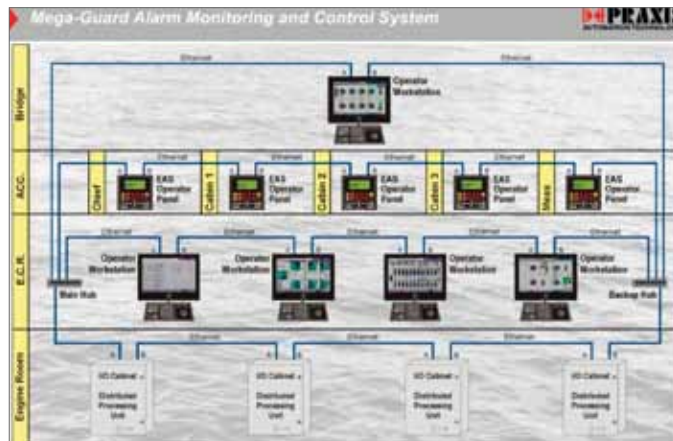
Rys. 4. Urządzenia Mega-Guard: a) sterownik/moduł I/O; b) panel operatorski – Praxis [4]

operacyjnym Windows 7 embedded. Pełni rolę stacji operatorskiej o szeregu wersjach językowych, w tym chińskiej i japońskiej.

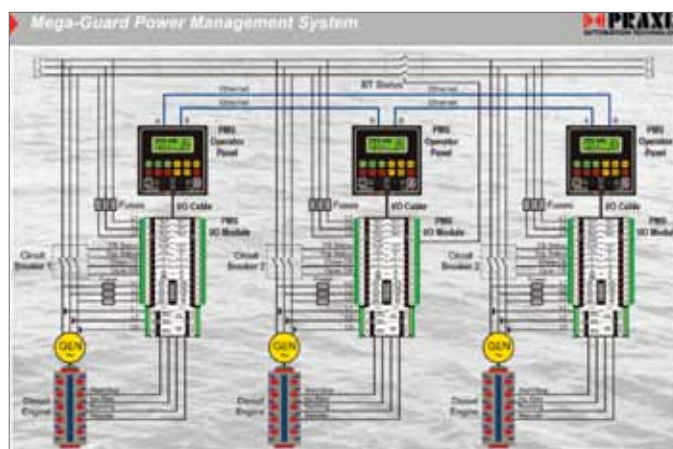
Jednostka sterująca. Składa się z modułu sterownika (*Control Processor*, rys. 4 a) oraz ewentualnie dodatkowych modułów I/O (wyglądających podobnie), z mikrokontrolerami ARM LPC. Moduł sterownika i moduły I/O połączone są CANem. Programy użytkownika uruchamia nieskomplikowany scheduler. Oprócz typowych wejść/wyjść analogowych i binarnych jednostka sterująca ma relatywnie wiele kanałów komunikacji szeregowej dla rozmaitych urządzeń pomiarowo-nawigacyjnych i napędowych statku. Zapewnione jest zasilanie czujników, izolacja galwaniczna i kontrola doziemienia. Sterownik i moduły I/O są montowane na szynie DIN.

Panel operatorski. Zawiera niewielki wyświetlacz LCD i zespół podświetlanych przycisków w układzie dedykowanym dla konkretnego podsystemu (rys. 4 b). Oprócz roli interfejsu HMI, panele monitorują alarmy oraz prowadzą niektóre obliczenia. Systemem operacyjnym jest Linux.

Sieć komunikacyjna. Komputery Marine PC, jednostki sterujące i panele operatorskie są połączone redundowanym Ethernetem. Możliwa jest architektura gwiazdy, pierścienia lub mieszana. Przełączniki i rozdzielacze sieciowe konfiguruje się tak, aby do danego węzła docierały głównie te informacje, których on potrzebuje.



Rys. 5. Architektura podsystemu monitorowania alarmów i sterowania



Rys. 6. Architektura podsystemu zarządzania energią

Podsystemy

Monitorowanie alarmów i sterowanie (*Alarm Monitoring and Control System*). Jest to zasadniczy podsystem Mega-Guarda, zdolny do obsługi od 24 do 24 tysięcy sygnałów I/O. Służy jako baza do integracji pozostałych podsystemów, tzn. zarządzania energią, napełniania zbiorników, sterowania napędami, pozycjonowania dynamicznego i nawigacji statku. Architekturę AMCS pokazano na rys. 5.

Napełnianie zbiorników (*Valve Control and Monitoring System*). Podsystem służy do zdalnej obsługi zaworów i pomp przy załadunku tankowców oraz do uzupełniania paliwa, balastowania itp. VCMS był jedną z pierwszych implementacji środowiska CPDev/PAL-1131.

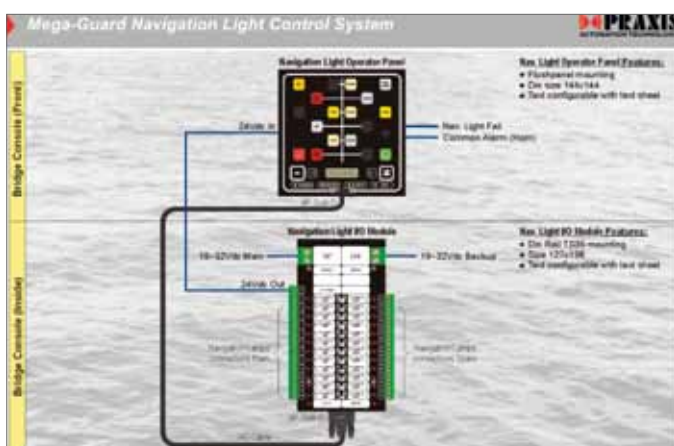
Sterowanie napędami (*Propulsion Control System*). Jest przeznaczony do obsługi z mostka kapitańskiego napędów głównych, przekładni, pędników kierunkowych itp.

Zintegrowana nawigacja (*Integrated Navigation System*). Obsługiwane są radary ARPA X i S (zakresy) oraz pomocnicze urządzenia nawigacyjne, z obrazami prezentowanymi na przynajmniej trzech stacjach operatorskich.

Zarządzanie energią (*Power Management System*). Każdy zespół silnik – generator, których może być 16, jest wyposażony we własny sterownik i panel operatorski (rys. 6). Podsystem PMS zapewnia sterowanie i ochronę generatorów, synchronizację, rozdział obciążenia itp.



Rys. 7. Nauka obsługi podsystemu pozycjonowania dynamicznego w sali szkoleń Praxis



Rys. 8. Architektura podsystemu obsługi świateł nawigacyjnych

Pozycjonowanie dynamiczne (Dynamic Positioning System). Chodzi np. o bezpieczne doprowadzenie statku dostawczego do platformy wiertniczej. Sterowanie pędników wzdłużnych i bocznych statku odbywa się za pomocą joysticków (rys. 7), zapewniając dokładność pozycjonowania na poziomie 0,5 m. Pozycję statku określa DGPS/Glan (dedykowany GPS), system hydroakustyczny lub linki naciągowe (*taut wire*). DPS jest wizytówką Praxis, bo na świecie produkują go tylko trzy inne firmy.

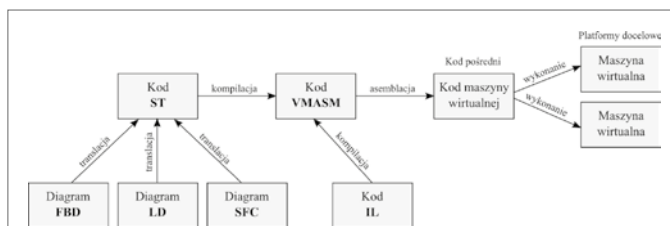
Warta na mostku nawigacyjnym (Bridge Navigation Watch). Monitorowana jest aktywność oficera na mostku, a w razie gdyby ustąpiła, najpierw włączany jest alarm, a potem przyzywany officer rezerwowy.

Alarm pożarowy (Fire Alarm System). Wykorzystuje się detektory dymu, ciepła i płomienia rozmieszczone w różnych sekcjach statku.

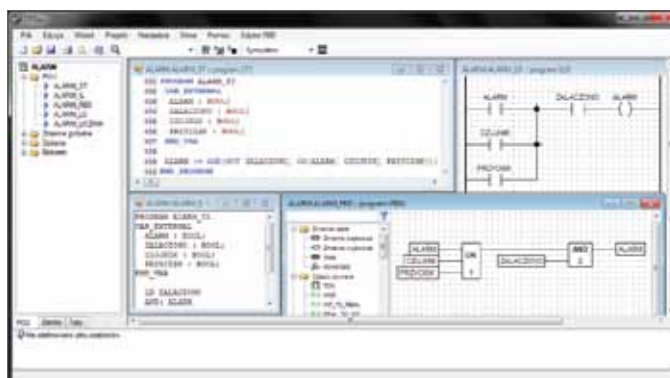
Alarmowanie i monitorowanie (Alarm and Monitoring System). Jest to mały podsystem (max. 192 DI, 96 AI, 96 SI – serial) kontrolujący podstawowe parametry niewielkiej jednostki, np. jachtu morskiego.

Zawartość zbiorników (Tank Gauging and Monitoring). Podsystem określa zawartość cieczy w zbiornikach o rozmaitych kształtach, uwzględniając jej ciężar właściwy.

Światła nawigacyjne (Navigation Light Control System). Jest to jeden z najmniejszych podsystemów Mega-Guarda, składający się tylko z panelu operatorskiego i modułu I/O (rys. 8).



Rys. 9. Proces wytwarzania kodu wykonywalnego w środowisku CPDev



Rys. 10. Elementarne programy napisane w językach ST, LD, IL i FBD

Świateł nawigacyjnych może być 12, każde o dwu lampach (*main, spare*).

Wycieraczki (Wiper Control System). Podsystem obsługuje 9 wycieraczek okien na mostku umożliwiając wybór prędkości, synchronizację ruchu, ogrzewanie, spryskiwanie.

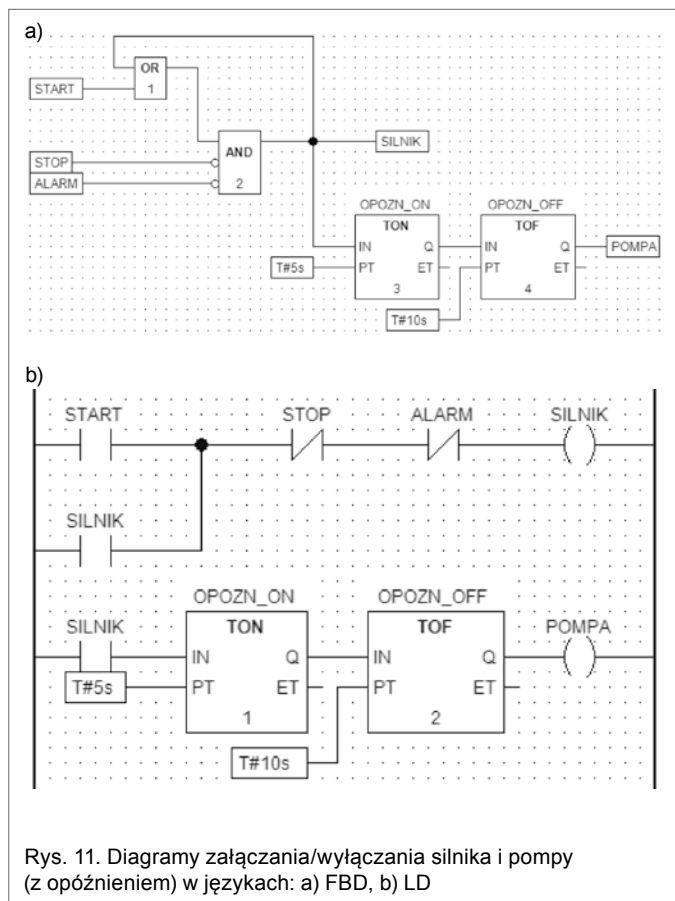
Podsumowując krótko powyższy przegląd, warto zwrócić uwagę na różnorodność funkcji, które pełnią podsystemy Mega-Guarda. Różnorodność ta jest wyraźnie większa niż w typowych systemach stosowanych w przemyśle.

Języki graficzne FBD, LD i SFC

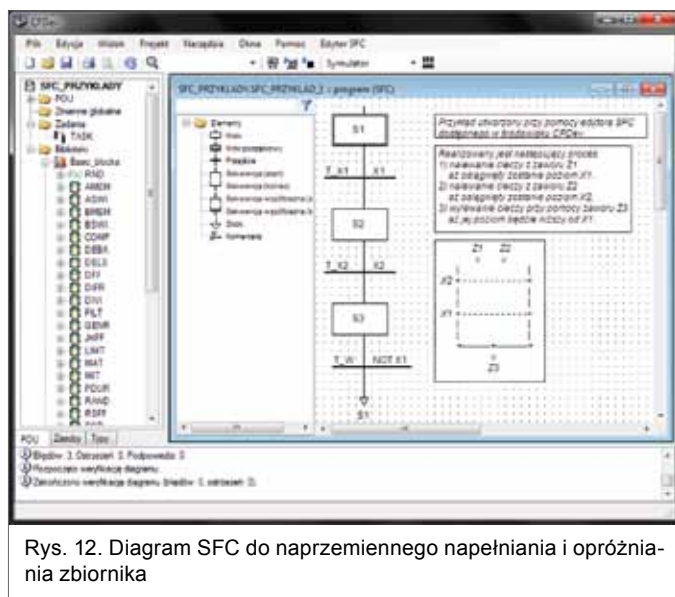
Język ST, jakkolwiek najbardziej uniwersalny, nie zapewnia jednak takiej przejrzystości algorytmów, jak diagramy utworzone w językach graficznych FBD, LD i SFC. Podjęto więc prace nad edytorami tych języków (M. Jamro), a następnie translatorami diagramów do kodu w języku ST, pozostającym bazowym językiem pakietu CPDev. Proces wytwarzania kodu wykonywanego na platformie docelowej przedstawia rys. 9. VMASM jest kodem pośrednim w języku maszyny wirtualnej (*Virtual Machine Assembler*). Bezpośrednio do kodu VMASM kompilowane są programy IL.

Okno środowiska CPDev z czterema elementarnymi programami napisanymi w językach ST, LD, IL i FBD pokazano na rys. 10. Podstawowymi cechami edytorów graficznych FBD, LD i SFC są:

- dodawanie, usuwanie, kopiowanie elementów diagramów;
- konwersja diagramów do formatu XML (wg PLC Open);
- automatyczne prowadzenie linii łączących (algorytm A*);
- zapisywanie i ładowanie diagramów;
- automatyczny dobór szerokości bloków;
- wykrywanie przecięć linii łączących;
- weryfikacja kompletności diagramu.



Rys. 11. Diagramy załączania/wyłączania silnika i pompy (z opóźnieniem) w językach: a) FBD, b) LD

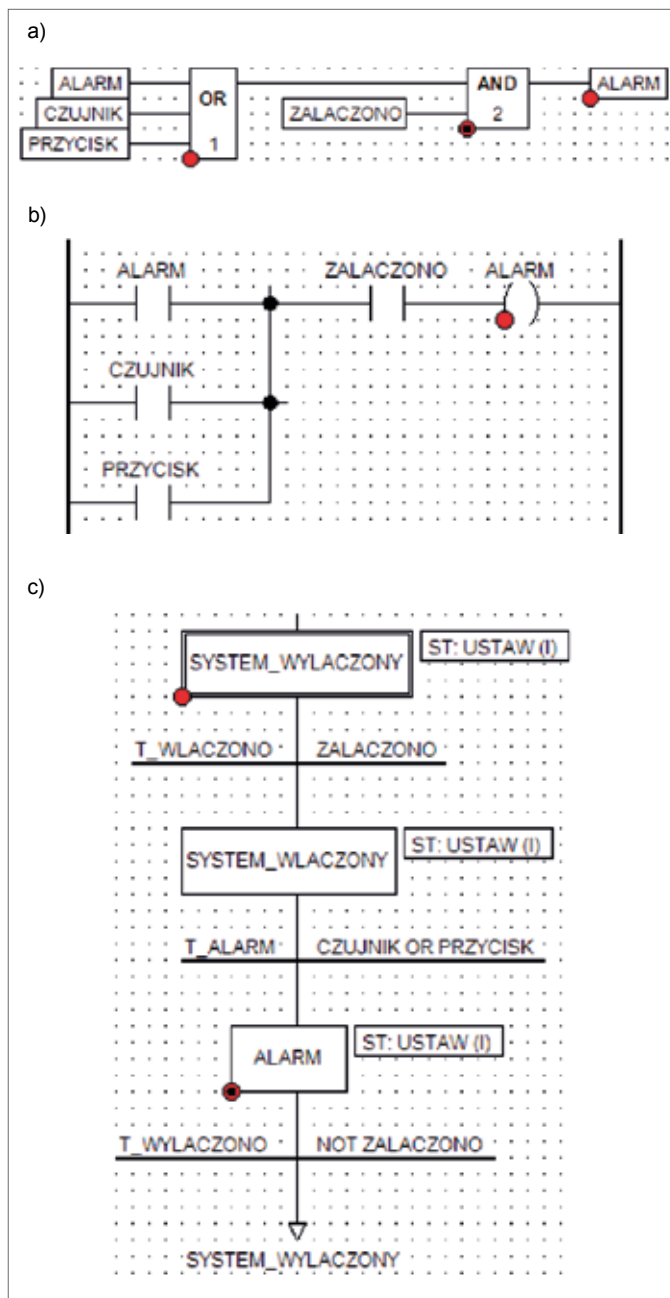


Rys. 12. Diagram SFC do naprzemiennego napełniania i opróżniania zbiornika

Jeżeli weryfikowany diagram okaże się kompletny, następuje translacja do ST, a potem kompilacja do VMASM.

Nieco bardziej złożony przykład załączania/wyłączania silnika i pompy odpowiednio po 5 i 10 sekundach w postaci diagramów FBD i LD pokazano na rys. 11 a, b. TON i TOF są standardowymi czasomierzami z biblioteki IEC_61131 pakietu CPDev.

Z kolei na rys. 12 przedstawiono diagram SFC realizujący naprzemienne napełnianie i opróżnianie zbiornika. Do jego utworzenia posłużyły elementy widoczne na liście pośrodku ekranu, tzn. krok początkowy, przejście, krok (zwykły) i skok. Oprócz nich elementami SFC mogą być sekwencje zwykłe i współbież-



Rys. 13. Pułapki bezwarunkowe i warunkowe na diagramach: a) FBD, b) LD, c) SFC

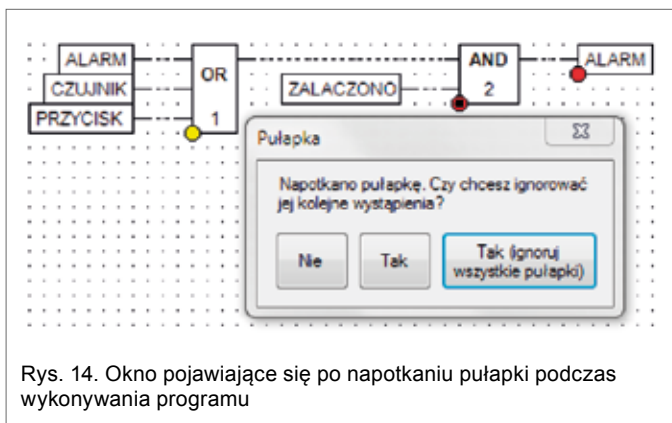
ne. Programy kroków, przejść i sekwencji można tworzyć w dowolnym języku spośród ST, FBD, LD i IL.

Pułapki na diagramach graficznych

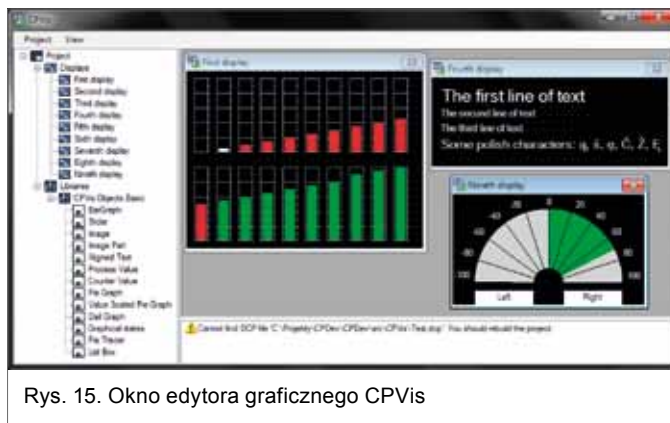
Uruchamianie w symulatorze CPSim programów napisanych w językach graficznych FBD, LD i SFC staje się łatwiejsze ze względu na możliwość wprowadzania pułapek. W razie napotkania pułapki wykonywanie programu jest przerywane, a użytkownik może skontrolować wartości zmiennych i lokalnych na listach i w okienkach dodatkowych symulatora CPSim [5].

Obecnie dostępne są dwa rodzaje pułapek:

- **bezw warunkowe** – powodujące każdorazowe przerywanie wykonywania programu, kiedy w kodzie ST odpowiadającym diagramowi napotkana zostanie pierwsza instrukcja dotycząca elementu diagramu, w którym umieszczono pułapkę;



Rys. 14. Okno pojawiające się po napotkaniu pułapki podczas wykonywania programu



Rys. 15. Okno edytora graficznego CPVis

- **warunkowe** – umożliwiające zatrzymanie programu jedynie wtedy, gdy spełniony jest dodatkowy warunek określony wraz z definiowaniem pułapki; warunek ten ma postać kodu zapisanego w ST.

Pułapki przedstawiane są na diagramach jako niewielkie czerwone kółka w odpowiednich miejscach z tym, że w przypadku pułapek warunkowych wewnątrz kółka znajduje się dodatkowa kropka (rys. 13 a, b, c).

Podczas wykonywania programu przez symulator CPSim po napotkaniu pułapki kolor jej zmienia się na żółty, a obok w małym oknie pojawia się pytanie o ignorowanie (rys. 14). Odpowiedź „Nie” powoduje przejście do okna CPSim z wartościami zmiennych globalnych i ewentualnie lokalnych. Wszystkie pułapki można ponadto zignorować.

Edytor obrazów graficznych CPVis

Jak wspomniano na początku, obecne panele operatorskie systemu Mega-Guard są dedykowane dla każdego z podsystemów i nie mogą być stosowane zamiennie. Lepiej byłoby, gdyby je konstrukcyjnie zunifikować, a o konkretnym przeznaczeniu decydować na podstawie programu użytkowego i obrazów prezentowanych na kolorowym wyświetlaczu LCD, jak to ma miejsce w uniwersalnych przemysłowych panelach HMI. Trzeba jednak pamiętać, że panele Mega-Guarda monitorują również zmienne procesowe i prowadzą niektóre obliczenia. Najkorzystniej byłoby więc, aby oprogramowanie użytkowe i wizualizację zintegrować w środowisku CPDev, tworząc programowalne urządzenie z interfejsem HMI, podobne z grubsza do niektórych sterowników PLC/PAC, np. Beckhoffa czy Hornera.

W odróżnieniu od paneli dedykowanych, uniwersalny panel wymaga opracowania edytora obrazów graficznych korzystającego z zestawu gotowych obiektów typowych dla wizualizacji sterowania i nawigacji statku. Prace nad takim edytorem, nazwanym CPVis, zostały niedawno rozpoczęte (M. Jamro), na podstawie specyfikacji obiektów graficznych przygotowanej przez Praxis.

Początkowy rezultat pokazano na rys. 15, gdzie po lewej stronie na dole widać listę obiektów przewidzianych do tworzenia obrazów, w tym m.in. bargraf, wycinek koła, wyświetlacz zmiennej procesowej itd. Kody źródłowe tych obiektów są tworzone w języku C++, a składający się z nich obraz przechowywany jest jako plik formatu XML. Po kompilacji do postaci binarnej plik ten będzie przysyłany do panelu, w którym znajdzie się również „szybka” biblioteka kreślenia elementarnych figur geometrycznych oraz alfabet chiński i japoński.

Podsumowanie

Środowisko inżynierskie CPDev (którego powstanie zainspirowały rozmowy podczas jednej z Konferencji Automatyków w Rytrze) dzięki przeznaczeniu na różne platformy sprzętowe oraz prostemu dołączaniu niskopoziomowego oprogramowania opracowanego przez konstruktorów-elektroników jest obecnie stosowane przez Zakłady Lumel oraz holenderską firmę Praxis Automation Technology do programowania sterowników w rozproszonych systemach automatyki i pomiarów. Dotychczas jednak dostępny był tylko język ST (od pewnego czasu także IL). Z inicjatywy Lumelu i Praxis podjęto prace nad udostępnieniem również języków graficznych FBD, LD i SFC ze względu na przejrzystość i przydatność do określonych zastosowań. Wiadomo bowiem, że FBD jest odpowiedni dla pętli PID i przetwarzania pomiarów, LD dla obwodów elektrycznych, a SFC dla sekwencji rozruchu i zatrzymania. Każdy z tego typu problemów występuje w systemach automatyzacji i nawigacji statków będących domeną Praxis. Nowością wprowadzoną w diagramach FBD, LD i SFC jest możliwość ustawiania pułapek ułatwiających wykrycie usterek i wprowadzenie poprawek. Podjęto również prace nad uzupełnieniem środowiska CPDev o projektowanie i obsługę wizualizacji ze względu na realne potrzeby oraz obserwowaną tendencję wyposażania sterowników w panele operatorskie HMI.

Literatura

- [1] RZOŃCA D., SADOLEWSKI J., STEC A., ŚWIDER Z., TRYBUS B., TRYBUS L.: *Prototypowy pakiet inżynierski do programowania sterowników w języku ST według normy IEC 61131-3*. XI Konferencja Automatyków, Rytró, 22–23.05.2007, s. 1–13.
- [2] RZOŃCA D., SADOLEWSKI J., STEC A., ŚWIDER Z., TRYBUS B., TRYBUS L.: *Dwustronna komunikacja i czas rzeczywisty w pakiecie CPDev dla rozproszonego systemu kontrolno-pomiarowego*. XII Konferencja Automatyków, Rytró, 20–21.05.2008, s. 1–11.
- [3] www.lumel.com.pl/.../regulacja_sterowanie
- [4] www.praxis-automation.nl/products_...
- [5] Sterownik programowalny typu SMC. Instrukcja programowania. Lumel SA, maj 2010.

Leszek Trybus, Marcin Jamro, Dariusz Rzońca, Jan Sadolewski, Andrzej Stec, Zbigniew Świder, Bartosz Trybus – Katedra Informatyki i Automatyki; Politechnika Rzeszowska, www.kia.prz.edu.pl