

Wykorzystanie kart wejść/wyjść obiektowych w pakiecie inżynierskim CPDev

Konrad Krok
Dariusz Rzońca
Jan Sadolewski

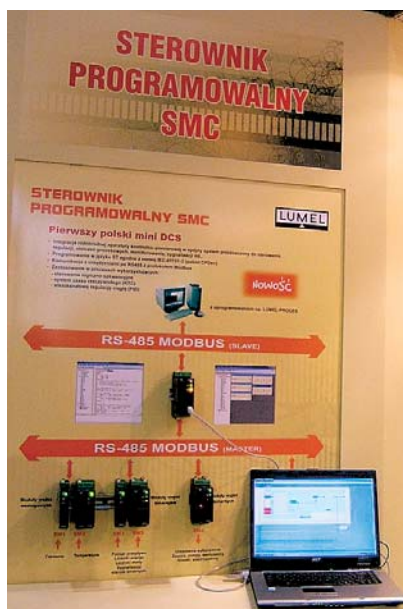
Dostępne na rynku liczne laboratoryjne karty wejść/wyjść obiektowych do komputera PC pozwalają na akwizycję danych z różnego rodzaju obiektów. Producenci kart zazwyczaj udostępniają własne, specjalizowane programy do ich obsługi wraz z odpowiednimi bibliotekami. W Katedrze Informatyki i Automatyki Politechniki Rzeszowskiej stworzono oprogramowanie pozwalające na stosowanie popularnych kart National Instruments NI-DAQ USB 6008 za pomocą pakietu inżynierskiego CPDev przeznaczonego do programowania sterowników zgodnie z normą IEC 61131-3. Komputer PC może być wykorzystywany jako *soft controller*.

CPDev

CPDev (*Control Program Developer*) [1, 2] jest otwartym środowiskiem inżynierskim do programowania sterowników i regulatorów w języku strukturalnym ST normy IEC 61131-3 (polski odpowiednik PN-EN 61131-3) [3, 4]. Otwartość jest tu rozumiana jako możliwość dostosowania do współpracy z platformami sprzętowymi różnych producentów. Pierwszym wdrożeniem pakietu CPDev jest nowy sterownik SMC [5] opracowany wspólnie przez LUMEL S.A. z Zielonej Góry i Politechnikę Rzeszowską. Niewielki rozproszony system sterowania i monitorowania oparty na sterowniku SMC (master) zaprezentowano po raz pierwszy na targach *AUTOMATICON 2008* (rys. 1).

W zastosowaniach laboratoryjnych i dydaktycznych można również wykorzystywać CPDev jako symulator lub programowany sterownik współpracujący z dołączoną do komputera PC kartą wejścia/wyjścia (*soft controller*). Przykładem takiej karty jest NI-DAQ USB 6008 [6] produkcji National Instruments (rys. 2).

Karta ta jest wielofunkcyjnym urządzeniem dołączanym przez port USB, które udostępnia 12 konfigurowalnych wejść/wyjść binarnych oraz 8 wejść i 2 wyjścia analogowe. Maksymalna częstotliwość prób-



Rys. 1. System sterowania mini-DCS ze sterownikiem SMC LUMEL - targi *AUTOMATICON 2008*



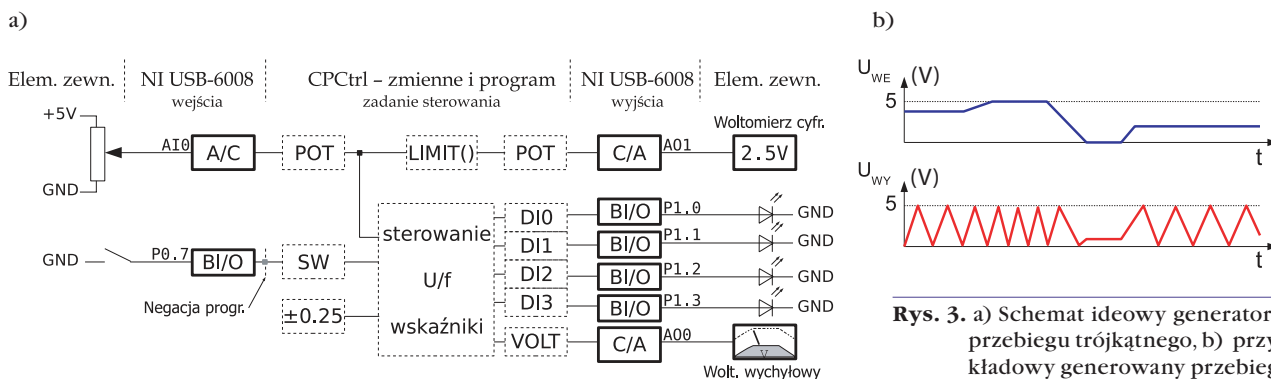
Rys. 2. Karta wejścia/wyjścia NI-DAQ USB 6008

kowania wynosi jedynie 10 kS/s. Jest to wystarczające, gdyż dodatkowe narzuty czasowe wnoszone przez interfejs USB i system Windows powodują, że sterownik realizowany wyłącznie programowo i tak nie wykorzystywałby należycie zalet szybszej karty. Dużą zaletą karty NI-DAQ USB jest dostępność bibliotek dla różnych języków programowania i różnych systemów operacyjnych oraz dobra dokumentacja zawierająca gotowe przykłady. Tutaj zastosowano biblioteki dla platformy .NET [7] zawierające klasy i metody służące do konfiguracji i komunikacji z tą kartą. Sterownik karty korzysta z technologii NI-DAQmx upraszczającej tworzenie oprogramowania do akwizycji danych.

Biblioteka obsługi kart wejścia/wyjścia

Soft controller realizowany za pomocą pakietu CPDev ma budowę modułową, pozwalającą na tworzenie rozszerzeń obsługujących sprzęt różnych producentów. Aby zbudować własny moduł (*assembly*) należy utworzyć bibliotekę dla platformy .NET odwołującą się do dwóch modułów dostarczonych przez CPDev: CPDev.CPCom.dll oraz CPDev.Public.dll. Pierwszy udostępnia interfejs CPDev.CPCom.ICommDev, który należy zaimplementować, aby własny moduł był obsługiwany przez główny komponent *soft controllera* - CPCtrl. Niektóre z funkcji tego interfejsu są omówione w dalszej części. Odwołanie do CPDev.Public.dll wynika z założenia, że każdy moduł

Konrad Krok, mgr inż. Dariusz Rzońca, mgr inż. Jan Sadolewski – Politechnika Rzeszowska, Katedra Informatyki i Automatyki



Rys. 3. a) Schemat ideowy generatora przebiegu trójkątnego, b) przykładowy generowany przebieg

musi wprowadzić do struktury CPDev.Public.ModuleInfoData informacje o nazwie modułu, wersji, prawach autorskich itp. Skompilowany do postaci binarnej moduł należy zarejestrować w *soft controllerze*. Można to zrealizować poprzez „ręczną” modyfikację pliku konfiguracyjnego CPDev.CPCtrl.modules.xml lub za pomocą opcji menu do zarządzania modułami. Plik z nowym modułem (niekiedy kilka plików) należy umieścić w folderze, gdzie znajduje się moduł wykonywalny CPDev.CPCtrl.exe. Następnie z menu *soft controllera* wybiera się opcję „Moduły | Rejestruj moduł” wskazując własny plik oraz klasę, którą należy zarejestrować (musi to być klasa publiczna, a także implementować interfejs CPDev.CPCom.ICommDev). Zatwierdzona konfiguracja będzie konfiguracją domyślną dla nowych projektów korzystających z tego sprzętu. Pewną niedoskonałością takiej rejestracji jest wybór według nazwy klasy, a nie po nazwie sprzętu (widocznej dopiero po rejestracji). Jest to jednak spowodowane względami bezpieczeństwa – nie są tworzone instancje wszystkich klas, jedynie tej, którą wskazał użytkownik. Inne klasy mogą bowiem okazać się kodem obcego pochodzenia z niepożądanym działaniem.

Interfejs CPDev.CPCom.ICommDev ma na celu ujednoczenie sposobu programowania kart wejścia/wyjścia obsługiwanych przez program CPCtrl. Funkcje wymagające implementacji przy tworzeniu własnych modułów zostały podzielone na trzy kategorie:

1. **Funkcje informacyjne** – udostępniające informacje dla CPCtrl o wejściach i wyjściach urządzenia, np.: GetNumberBinaryInputs, GetNumberAnalogOutputs, o błędach napotkanych przez bibliotekę komunikacji ze sterownikiem: GetErrorsCount, GetTextLastError, czy nazwie identyfikującej typ urządzenia sprzętowego – GetDeviceName.
2. **Funkcje sterujące** – zezwalające programowi CPCtrl na manipulację zasobami, jak np.: SetBinaryOutput, GetAnalogInput, ClearErrorsCounter.
3. **Funkcje konfiguracyjne** – pozwalające na zapamiętywanie i odczytywanie konfiguracji pracy: SaveCurrentConfiguration, LoadSavedConfiguration, oraz wyświetlanie okienka konfiguracji urządzenia DoConfigDialog.

Listing 1. Kod programu kompilowany pakietem CPDev

Przykład – generator przebiegu trójkątnego

W celu zademonstrowania możliwości programu CPCtrl współpracującego z kartami wejścia/wyjścia, utworzono programowy generator wolnozmiennego przebiegu trójkątnego przestrajany napięciem. Schemat ideowy generatora pokazano na rys. 3a, a przykładowy przebieg na rys. 3b.

Kod programu, napisany w języku ST, kompilowany pakietem CPDev, podano w listingu 1. W zależności od wartości napięcia ustawionego potencjometrem (rys. 3a) następuje zmiana napięcia wyjściowego VOLT. Szybkość zmian jest proporcjonalna do ustawionego potencjału. Zmienne POT i VOLT przed przekazaniem do przetworników C/A muszą zostać ograniczone do przedziału 0...5 V funkcją LIMIT(). Diody DI0... DI3 sygnalizują aktualny kierunek zbocza sygnału wyjściowego poprzez wędrowkę zapalającego się punktu. Przyciskiem można modyfikować sposób wyświetlania pomiędzy jednym zapalonym punktem, a jednym zgaszonym.

```

PROGRAM PRG
VAR _EXTERNAL (*$AUTO*)
END_VAR VAR
STEP : REAL := 0.25;
STEP2 : REAL;
LB, PSW : BOOL;
END_VAR;

IF SW AND NOT PSW THEN
DI0 := NOT DI0; DI1 := NOT DI1;
DI2 := NOT DI2; DI3 := NOT DI3;
END_IF;
PSW := SW;

POT := LIMIT(POT, 0.0, 5.0);

IF STEP > 0.0 THEN
STEP2 := STEP + POT*0.1 - 0.25;
ELSE
STEP2 := STEP - POT*0.1 + 0.25;
END_IF
VOLT := VOLT + STEP2;
IF VOLT > 5.0 THEN STEP := NEG(STEP);
ELIF VOLT < 0.0 THEN STEP := NEG(STEP);
END_IF
VOLT := LIMIT(VOLT, 0.0, 5.0);
IF STEP > 0.0 THEN
LB := DI0; DI0 := DI1; DI1 := DI2;
DI2 := DI3; DI3 := LB;
ELSE
LB := DI3; DI3 := DI2; DI2 := DI1;
DI1 := DI0; DI0 := LB;
END_IF
END_IF
END_PROGRAM

```

Zmienne globalne prezentowane są na rys. 3a w postaci nazw otoczonych przerywaną ramką. W lewej części znajdują się zmienne odczytywane z otoczenia – do nich trafiają informacje o stanie przełącznika i ustawieniu potencjometru, natomiast w prawej zmienne zapisywane – ustawienie ich wartości powoduje wymuszenie żadanego stanu na wyjściu przetworników. Zmienne są zadeklarowane jak w listingu 2:

```
VAR _GLOBAL
VOLT : REAL;
DI0  : BOOL := TRUE;
DI1  : BOOL := FALSE;
DI2  : BOOL := FALSE;
DI3  : BOOL := FALSE;
POT  : REAL;
SW   : BOOL;
END _VAR
```

Listing 2. Deklaracje zmiennych

Odczyt i zapis zmiennych konfiguruje się w oknie *soft controllera* (rys. 4). W pliku *.dcp z informacjami od kompilatora znajduje się lista zmiennych globalnych. Należy wybrać sprzęt docelowy (tutaj kartę NI-DAQ USB 6008) i skonfigurować jego parametry za pomocą przycisku „Ustawienia”. Po zatwierdzeniu ustawień dokonuje się powiązania zmiennych z poszczególnymi portami urządzenia za pomocą menu kontekstowego lub pól wyboru znajdujących się nad tabelami zmiennych (rys. 4). Zmienna SW używa jako typu kanału „Zanegowany binarny”. Związane jest to z budową wewnętrzną portu binarnego w karcie i zastosowanego normalnie otwartego przełącznika (rys. 3). W chwili, gdy przełącznik jest otwarty wyjście zwraca TRUE, zaś gdy zamknięty – FALSE. Kod programu zakłada, że przełącznik zgłasza TRUE, gdy jest aktywny, czyli odwrotnie niż jest to zrealizowane sprzętowo. Aby temu zaradzić bez modyfikacji kodu, wprowadzono typ kanału „Zanegowany binarny”, który po odczycie portu przekazuje do zmiennej zanegowaną wartość. Typ ten działa analogicznie podczas zapisu zmiennej do portu binarnego.

Skonfigurowany w ten sposób przydział zmiennych do zasobów sprzętowych należy wprowadzić do *soft controllera* przyciskiem „Zastosuj”, a następnie uruchomić program. Przydział zasobów sprzętowych wraz z konfiguracją sprzętową można zapisać do pliku w ce-

lu późniejszego odtworzenia. Informacja o ewentualnym niepowodzeniu operacji zapisu, przekroczeniu czasu cyklu, czy błędzie w konfiguracji karty I/O pojawiają się w polu u dołu okna. W systemie MS Windows występują ograniczenia na czas cyklu, związane z wbudowanym algorytmem szeregowania zadań (nie jest to system czasu rzeczywistego i nie gwarantuje determinizmu czasowego). W związku z tym zaimplementowano w CPCTRL mechanizm automatycznego wydłużania czasu cyklu, w przypadku jego notorycznego przekroczenia (o czym użytkownik jest informowany komunikatem na belce statusu). Przedstawiony przykład zaimplementowano na stanowisku laboratoryjno-dydaktycznym pokazanym na rys. 5.

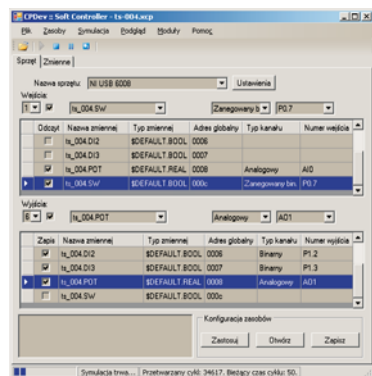
Podsumowanie

Przedstawiono sposób implementacji obsługi modułów wejścia/wyjścia w pakiecie inżynierskim CPDev na przykładzie karty National Instruments NI-DAQ USB 6008. Opracowano stanowisko laboratoryjno-dydaktyczne demonstrujące możliwości pakietu CPDev i sposób obsługi kart. Na stanowisku uruchomiono program realizujący przetwornik typu napięcie/częstotliwość, napisany w języku ST. W ramach dalszych prac planowana jest implementacja kart o lepszych parametrach, jak np. RT-DAC/USB I/O produkcji krakowskiej firmy Inteco, co zwiększy możliwości zastosowań, np. w mechatronice.

Praca częściowo finansowana jako projekt rozwojowy MNiSzW nr R0205803.

Bibliografia

1. Rzońca D., Sadolewski J., Stec A., Świder Z., Trybus B., Trybus L.: *Środowisko programistyczne dla rozproszonych minisystemów kontrolno-pomiarowych*, Metody Informatyki Stosowanej 1/2008 (tom 13).
2. Rzońca D., Sadolewski J., Stec A., Świder Z., Trybus B., Trybus L.: *Mini-DCS System Programming in IEC 61131-3 Structured Text*, *Journal of Automation, Mobile Robotics & Intelligent Systems*, vol. 2 no. 3, 2008.
3. PN-EN 61131-3:2004 *Sterowniki programowalne. Część 3: Języki programowania*.
4. Kasprzyk J.: *Programowanie sterowników przemysłowych*. WNT, Warszawa, 2006.
5. *Sterownik programowalny typu SMC*. Instrukcja programowania „Szybki start”, LUMEL S.A. i Politechnika Rzeszowska, Zielona Góra/Rzeszów, 2008.
6. *User Guide and Specifications NI USB-6008/6009*, National Instruments 2008.
7. *MS .NET Framework Developer's Guide* <http://msdn2.microsoft.com/en-US/library/aa720433.aspx>. ■



Rys. 4. Konfiguracja wiązań zmiennych globalnych do kanałów karty I/O



Rys. 5. Stanowisko laboratoryjno-dydaktyczne z kartą NI-DAQ USB 6008 i pakietem CPDev