Programowany monitor stanów sygnałów obiektowych i zmiennych systemowych sterowników PLC

Jacek Dunaj, Dariusz Grabowski
Przemysłowy Instytut Automatyki i Pomiarów PIAP, Al. Jerozolimskie 202, 02-486 Warszawa

Streszczenie: Podczas uruchamiania stanowisk przemysłowych zawierających sterowniki PLC, roboty i komputery przemysłowe dużym ułatwieniem jest możliwość podglądu stanu sygnałów obiektowych i zmiennych programów aplikacyjnych. Producenci urządzeń sterujących w swoim oprogramowaniu do tworzenia aplikacji często udostępniają dodatkowe narzędzia do realizacji takich funkcji. Ciekawym rozwiązaniem jest oprogramowanie MX Components firmy Mitsubishi Electric oferujące nie tylko możliwość podglądu tego, co dzieje się w sterowniku, ale też dostarczające narzędzi do tworzenia własnego programu diagnostycznego. W artykule przedstawiono możliwości funkcjonalne programowalnego monitora stanów sygnałów obiektowych i zmiennych systemowych sterowników Mitsubishi, który uruchomiono w celu ułatwienia testowania i diagnostyki aplikacji przemysłowych wykonywanych w Instytucie.

Słowa kluczowe: sterownik PLC, aplikacja komputera PC, oprogramowanie, podgląd stanu sygnałów obiektowych i zmiennych systemowych sterowników

1. Wprowadzenie

Podczas prac montażowych, uruchamiania i testowania stanowisk przemysłowych, wyposażonych w programowalne sterowniki PLC dużym ułatwieniem jest możliwość podglądu aktualnego stanu i ręcznego wysterowywania różnych sygnałów obiektowych. Dodatkowym wyposażeniem niektórych sterowników są diody świecące sprzężone z pojedynczymi wejściami i wyjściami. Innym sposobem przekazywania informacji o sygnałach obiektowych jest oprogramowanie umożliwiające monitorowanie ich stanu za pomocą standardowego interfejsu. Diody przekazują informację szybciej, ale specjalizowane oprogramowanie oferuje znacznie większe możliwości.

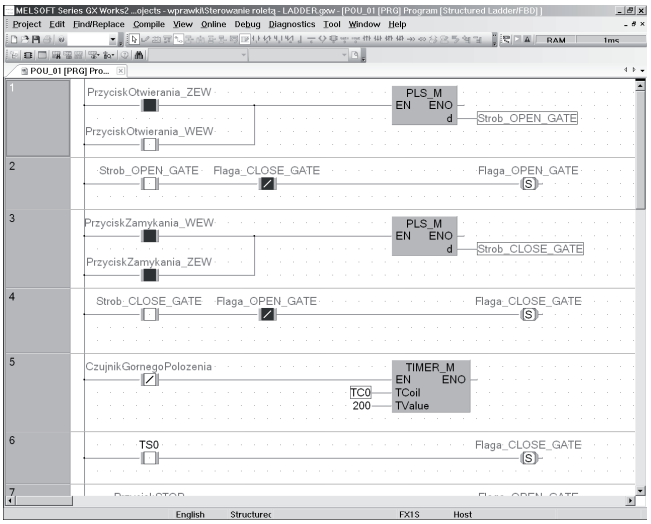
Aplikacja sterownika PLC najczęściej jest tworzona na zewnętrznym komputerze PC za pomocą dedykowanego oprogramowania. Oprogramowanie takie zawiera funkcje tworzenia i edycji programu aplikacyjnego oraz zapewnia pełną obsługę samego sterownika – zapis i odczyt aplikacji do/z pamięci, uruchamianie, zatrzymywanie i debugging programu, podgląd stanu wejść, ustawianie zmiennych systemowych odnoszących się np. do nastaw zegara i kalendarza, adresacji i parametryzacji portów transmisyjnych. W zależnosci od sprzętowego wyposażenia sterownika, wszystkie te czynności są wykonywane za pośrednictwem standardowych interfejsów komunikacyjnych (RS-232, Ethernet, USB, IEEE 1394). Również oprogramowanie systemowe sterownika powinno realizować funkcje umożliwiające współpracę z odpowiednim programem komputera PC.

Za wykonywanie programu aplikacyjnego odpowiada oprogramowanie systemowe sterownika. System odczytuje stany wszystkich wejść sterownika i zapamiętuje je w odpowiednich rejestrach pamięci, następnie wykonuje cały program aplikacyjny. Jeśli kolejne instrukcje programu zawierają odwołania do stanu wejść, ich wartości są pobierane z rejestrów pamięci. Jeśli efektem działania aplikacji ma być zmiana stanu wyjść, to informacja o tym jest zapamiętywana w rejestrach sterownika, a właściwe wysterowanie ma miejsce dopiero po wykonaniu całego programu aplikacyjnego. Specyfika programu aplikacyjnego sterownika PLC wymaga ograniczonego czasu wykonania, więc program nie może zostać „zapętlony" w oczekiwaniu na spełnienie warunku umożliwiającego wyjście z pętli. Po wykonaniu aplikacji i wysterowaniu wyjść, cały cykl jest powtarzany. Jeśli programu aplikacyjnego nie wprowadzono do pamięci sterownika lub po wczytaniu nie został uruchomiony, oprogramowanie systemowe cały czas cyklicznie realizuje wymianę informacji między wejściami i wyjściami a przypisanymi im rejestrami pamięci. Przez cały czas prowadzony jest też nasłuch kanału transmisyjnego wykorzystywanego do komunikacji z oprogramowaniem komputera PC przeznaczonym do obsługi sterownika. W przypadku, gdy sterownik ma dwa kanały (np. do sterownika Mitsubishi z jednostką centralną Q02HCPU wbudowano port RS-232 i port USB) to nasłuch prowadzony jest w obu kanałach. Nasłuch nie wymaga

Autor korespondujący:
Jacek Dunaj, jdunaj@piap.pl

Artykuł recenzowany
nadesłany 09.05.2016 r., przyjęty do druku 25.05.2016 r.

tworzenia dodatkowego kodu w programie aplikacyjnym, nie ma też bezpośredniego wpływu na jego wykonywanie (poza komendami START i STOP), jest wykonywany przez oprogramowanie systemowe sterownika.



Rys. 1. Okno programu GX Works2 do podglądu stanów niektórych sygnałów obiektowych
Fig. 1. GX Works2 window to preview the states of some signals of object

Sterowniki PLC firmy Mitsubishi są programowane za pomocą oprogramowania GX Developer, GX IEC Developer lub GX Works2 pracującego na komputerze PC pod kontrolą systemu operacyjnego Windows. Wymienione programy współpracują z oprogramowaniem systemowym sterowników, umożliwiają nie tylko tworzenie aplikacji, ale także realizują wymienione funkcje obsługi. Z punktu widzenia osoby montującej stanowisko lub poszukującej uszkodzenia w układzie sterowania (niebędącej autorem programu aplikacyjnego), oprogramowanie to ma kilka wad:

- wymaga zainstalowania na komputerze operatora i potwierdzenia praw licencyjnych,
- operator musi umieć posługiwać się programami narzędziowymi i znać specyfikę programowania sterowników PLC,
- sprawdzenie działanie elementów stanowiska zwykle wymaga znajomości postaci źródłowej programu, jego struktury, funkcji realizowanych przez różne moduły i ich wzajemne powiązania – w praktyce jest to zwykle utrudnione,
- sięganie do źródeł aplikacji za pomocą oprogramowania narzędziowego zwykle grozi nieumyślnym wprowadzeniem niepożądanych modyfikacji.

Na rys. 1 pokazano przykładowe okno programu GX Works2 z otwartym podglądem fragmentu aplikacji sterownika Mitsubushi FX1S w trybie cyklicznego monitorowania stanu niektórych sygnałów i zmiennych programu. PrzyciskOtwierania_ZEW będzie wypełniony kolorem niebieskim, jeśli operator wciśnie przycisk dołączony do odpowiedniego wejścia dwustanowego, a kolorem białym – jeśli przycisk zostanie zwolniony. Aby informacja ta pojawiła się w oknie programu GX Works2 program wysyła do sterownika odpowiednie zapytanie, a oprogramowanie systemowe sterownika PLC odsyła odpowiedź. Treść odpowiedzi nie jest generowana na podstawie bezpośredniego odczytu wejścia dwustanowego, lecz informacji zapisanej w rejestrze pamięci sterownika. Zawartość tego rejestru jest cyklicznie uaktualniana przez system sterownika.

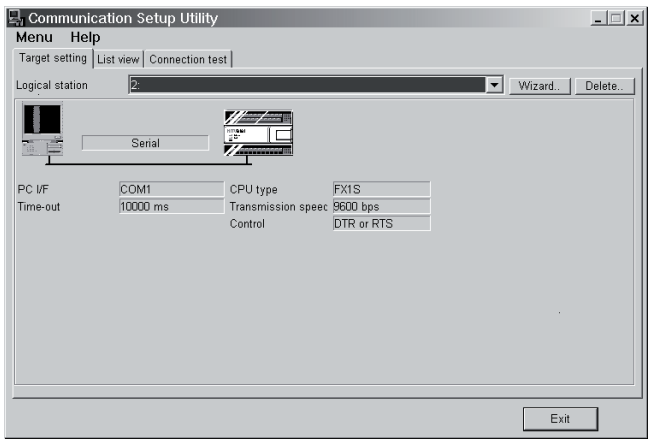
2. Oprogramowanie MX Components

Funkcja monitorowania stanu sygnałów i zmiennych za pomocą pakietu GX Works2 jest bardzo przydatna, jednak brak znajomości tego narzędzia oraz struktury i wewnętrznych oznaczeń zmiennych aplikacji uniemożliwia praktyczne jej wykorzystanie. Potrzebne jest rozwiązanie, które upraszcza sposób odczytu informacji bezpośrednio ze sterownika i umożliwia dowolnej aplikacji komputera PC komunikację z oprogramowaniem systemowym sterownika tak jak GX Works2.

Pakiet MX Components firmy Mitsubishi oferuje możliwość znacznie prostszego podglądu stanu nie tylko sygnałów obiektowych (wejść, wyjść), ale także zmiennych systemowych (m.in. timery, liczniki, markery, rejestry danych). Dostarcza również moduły, które umożliwiają zbudowanie aplikacji komputera PC do bezpośredniej komunikacji z oprogramowaniem systemowym sterownika PLC.

2.1. Communication Setup Utility

Aplikacja umożliwia definiowanie kanału transmisyjnego między komputerem PC i sterownikiem Mitsubishi. Definiowanie kanału polega na nadaniu numeru (0–1023), określeniu rodzaju interfejsu do jego obsługi (RS-232, USB, Ethernet) oraz określeniu parametrów transmisji, jeśli są wymagane. Jednym z dodatkowych parametrów jest typ jednostki centralnej sterownika, nie ma on jednak wpływu na połączenie ze sterownikiem, ponieważ kanał o wybranym numerze można wykorzystywać do połączenia z inną jednostką centralną niż zadeklarowano podczas definiowania kanału. Służy on do weryfikacji, czy wybrany moduł jednostki centralnej zawiera wybrany interfejs komunikacyjny. Po zdefiniowaniu kanału i fizycznym połączeniu komputera ze sterownikiem można wykonać test transmisji (Connection test) sprawdzający, czy współpraca komputera ze sterownikiem jest prawidłowa.

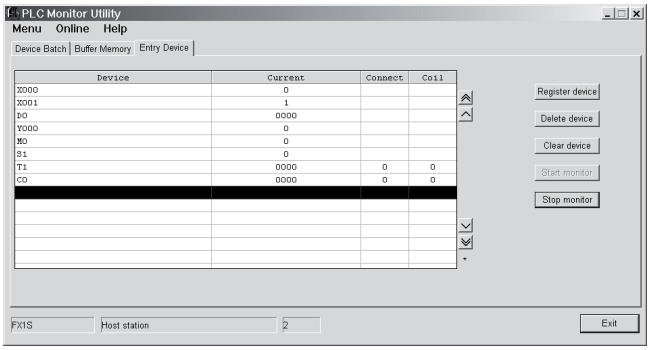


Rys. 2. Okno dialogowe aplikacji Communication Setup Utility do definiowania kanału transmisyjnego do/z sterownika
Fig. 2. A dialog window applications Communication Setup Utility to define the transmission channel to/from controller

Definiowanie kanału jest istotne dla innych elementów oprogramowania MX Components. Na jednym komputerze PC może być zdefiniowanych wiele kanałów transmisyjnych.

2.2. PLC Monitor Utility

Aplikacja umożliwia podgląd i monitorowanie aktualnego stanu sygnałów obiektowych oraz zmiennych systemowych sterownika. Na rys. 3 przedstawiono przykład okna dialogowego z zaprogramowanym podglądem stanów następujących zmiennych: wejść X0 i X1, rejestru danych D0, wyjścia Y0,



Rys. 3. Okno dialogowe aplikacji PLC Monitor Utility do podglądu stanu sygnałów obiektowych i zmiennych systemowych sterowników
Fig. 3. A dialog window applications PLC Monitor Utility to view the status of object signals and system variables controllers

wewnętrznych znaczników binarnych M0 i S1, timera T1 oraz licznika C0. Listę monitorowanych urządzeń (sygnałów) ustala się za pomocą przycisków: Register device (dopisz urządzenie do listy), Delete device (usuń urządzenie z listy), Clear device (usuń wszystkie urządzenia z listy). Aplikację tę można stosować podczas prac montażowych, uruchamiania i testowania stanowiska z programowalnym sterownikiem PLC.

2.3. Kontrolki Active-X

Podczas instalacji pakietu MX Components zostają zainstalowane kontrolki Active X, które w tym przypadku dostarczają komponentów umożliwiających zbudowanie aplikacji komputera PC do współpracy z oprogramowaniem systemowym dowolnego sterownika Mitsubishi. Współpraca ta odbywa się za pomocą kanału transmisyjnego, zdefiniowanego programem Communication Setup Utility. Od autora aplikacji komputera PC nie jest wymaga wiedza na temat ramki protokołu ani sposobu jej wysyłania i odbioru odpowiedzi, ponieważ dostępne są gotowe funkcje zawarte w bibliotekach do obsługi kontrolek. Do komunikacji z komputerem PC nie jest także potrzebna żadna dodatkowa modyfikacja programu użytkowego sterownika PLC, co więcej, sterownik nie musi zawierać takiego programu. Kontrolki te są narzędziem znacznie rozszerzającym możliwości funkcjonalne sterownika i całego stanowiska.

Jeśli aplikacja komputera PC jest tworzona za pomocą oprogramowania Microsoft Visual C++ .NET, to proces dołączania elementów do współpracy ze sterownikiem Mitsubishi PLC wymaga wykonania czynności:

- do projektu należy dołączyć dwa predefiniowane pliki ActEasyIf.cpp i ActEasyIf.h, które są fragmentami przykładów instalowanych wraz z pakietem MX Components – zawierają one deklarację i definicję klasy CActEasyIf;
- klikając prawym przyciskiem myszy na projekt okna dialogowego aplikacji należy wybrać opcję Insert Active X Control a następnie na liście zainstalowanych kontrolek wskazać MITSUBISHI ActEasyIF Control
- korzystając z definicji klasy CActEasyIf zadeklarować zmienne:

```
CActEasyIf    m_ActEasyIF;
long          ErrorCode.;
```

Zbiór dostępnych funkcji opisano w [1, 3]. Najważniejsze z nich i najczęściej stosowane to:

1. Otwarcie kanału transmisyjnego.
```
m_ActEasyIF.put_ActLogicalStationNumber
(NumerKanaluTransmisyjnego);
ErrorCode = m_ActEasyIF.Open();
```
gdzie: NumerKanaluTransmisyjnego – numer kanału zdefiniowany za pomocyą aplikacji Communication Setup Utility.
2. Zamknięcie kanału transmisyjnego:
```
ErrorCode = m_ActEasyIF.Close();
```
3. Uruchomienie aplikacji sterownika PLC.
```
ErrorCode = m_ActEasyIF.SetCpuStatus(0);
```
4. Zatrzymanie aplikacji sterownika PLC.
```
ErrorCode = m_ActEasyIF.SetCpuStatus(1);
```
5. Odczyt wartości zmiennej systemowej sterownika:
```
ErrorCode = m_ActEasyIF.GetDevice
(DeviceName,&GetDeviceValue);
```
gdzie: CString DeviceName określa, jakiej zmiennej systemowej sterownika dotyczy odczyt, np. X0, D0, Y0, M0, S1, TC1, CC0; long GetDeviceValue określa zmienną aplikacji komputera PC gdzie zostanie zapisana wartość odczytana ze sterownika.
6. Ustawienie (zapis) wartości zmiennej systemowej sterownika.
```
ErrorCode = m_ActEasyIF.SetDevice
(DeviceName,SetDeviceValue);
```
gdzie: CString DeviceName określa, jakiej zmiennej systemowej sterownika dotyczy zapis, np. X0, D0; long SetDeviceValue określa zmienną aplikacji komputera PC której wartość będzie wpisana do wskazanej zmiennej systemowej sterownika PLC

Każda z tych funkcji zwraca kod błędu (ErrorCode). Wartość 0x0000 oznacza, że wykonanie funkcji zakończyło się sukcesem. Oprócz wymienionych funkcji odczytu/zapisu pojedynczych zmiennych, kontrolki Mitsubishi ActEasyIF Control dopuszczają odczyt/zapis całych bloków pamięci sterownika PLC.

Zastosowanie kontrolek Mitsubishi ActEasyIF Control nie ogranicza się do wymiany informacji między aplikacją PC a zmiennymi systemowymi sterownika (X, Y, M, S, T, C). Każda aplikacja ma dostęp do tzw. rejestrów danych (oznaczanych D), które mogą być wykorzystane jako bufor wymiany danych między programami PLC a komputera PC. W Przemysłowym Instytucie Automatyki i Pomiarów PIAP zastosowano je we wdrożeniach przemysłowych, takich jak: urządzenie testujące mechanizmy sprężynowe [8], urządzenie testujące rozłączniki SN, wykonanych dla Zakładów ABB Sp. z o.o z Przasnysza. W obu przypadkach programy użytkowe sterowników PLC, testując wybrane urządzenie, zapamiętywały kolejne punkty charakterystyki w rejestrach danych D. Po zakończeniu badania aplikacja PC kanałem RS-232 odczytywała zapamiętaną informację i wykonywała końcową ocenę wyrobu.

Ze względu na stosowanie elementów MX Components bardziej zaawansowaną aplikacją było sterowanie dwoma robotami rehabilitacyjnymi Renus opracowanymi w PIAP [9, 10]. W tym przypadku oprogramowanie PC pełniło rolę panelu operatorskiego dla sterowników obu robotów, inicjując wykonywanie czynności oraz pośredniczenie w wymianie informacji między sterownikiem PLC a osadzoną na komputerze PC bazą danych trajektorii. Podczas definiowania trajektorii (uczenia robota) oraz podczas wykonywania ćwiczeń rehabilitacyjnych kanałem transmisyjnym wykorzystującym interfejs USB na bieżąco przekazywano odczyty położeń wałów trzech silników manipulatora i oddziaływujących na nie sił.

3. Monitor stanów sygnałów obiektowych i zmiennych systemowych sterowników PLC

Podczas uruchamiania i testowania stanowisk przemysłowych bardzo pomocny jest podgląd aktualnych wartości i modyfikacja zmiennych systemowych bez konieczności wnikania w strukturę aplikacji sterownika PLC. Ułatwia to aplikacja PLC Monitor Utility, ale w praktyce jego możliwości funkcjonalne nie zawsze idą w parze z aktualnymi potrzebami. Dlatego

przy okazji realizacji linii produkcyjnej matryc ciekłokrystalicznych urządzenia wykrywającego raka piersi, gdzie zastosowano sterownik firmy Mitsubishi, zdecydowano się opracować własny uniwersalny program diagnostyczny, wykorzystujący elementy oprogramowania MX Components. W założeniu program ten powinien umożliwiać programowanie tego, co operator chce monitorować, łatwą zmianę aktualnego podglądu, być niezależnym od programu aplikacyjnego sterownika PLC oraz udostępniać funkcje podstawowe: „Uruchom program", „Zatrzymaj program", „Ustaw lub odczytaj zegar i kalendarz sterownika", „Odczytaj typ jednostki centralnej".

Program napisano w języku C++ i uruchomiono w środowisku MS Visual Studuio .NET 2003. Nosi on nazwę Mitsubishi_IO_Test.exe i nie wymaga instalacji, lecz skopiowania do pamięci komputera.

3.1. Pliki konfiguracyjne monitora sygnałów obiektowych

Podstawą działania programu monitora sygnałów są tekstowe pliki konfiguracyjne, które określają listy monitorowanych sygnałów obiektowych i zmiennych systemowych (urządzeń), sposób ich prezentacji oraz inne parametry modyfikujące warunki pracy programu.

Każdy wiersz pliku konfiguracyjnego jest oddzielnie analizowany przez aplikację. Jeśli wiersz nie zawiera znaku „=" jest traktowany jako komentarz. Jeśli jednak aplikacja wykryje w wierszu znak równości, to wszystkie znaki wiersza poprzedzające znak „=" są traktowane jako nazwa zmiennej, a wszystkie znaki na prawo od tego znaku są traktowane jako wartość zmiennej. Długość każdego wiersza nie może przekraczać 825 znaków. Podczas analizy pliku konfiguracyjnego aplikacja rozróżnia duże i małe litery.

3.1.1. Zmienne pliku konfiguracyjnego do ustalania list monitorowanych sygnałów obiektowych i zmiennych systemowych sterownika

W programie źródłowym monitora sygnałów do odczytu/zapisu wartości pojedynczej zmiennej systemowej sterownika PLC wykorzystywane są dwie funkcje:

```
ErrorCode = m_ActEasyIF.GetDevice
   (DeviceName,&GetDeviceValue);
ErrorCode = m_ActEasyIF.SetDevice
   (DeviceName,SetDeviceValue);
```

Obie funkcje jako parametru o mnemoniku „DeviceName" używają łańcucha znaków określającego daną zmienną. W poniższej tabeli pokazano kilka przykładów ilustrujących prawidłową postać tego parametru:

Parametr DeviceName	Znaczenie	Parametr DeviceName	Znaczenie
X00	wejście dwustanowe X0	D0	rejestr danych D0
Y10	wyjście dwustanowe Y10	M0	rejestr wskaźników M0
TN0	timer T0 – aktualna wartość	CN0	licznik C0 – aktualna wartość
TC0	timer T0 – cewka	CN0	licznik C0 – cewka
TS0	timer T0 – wyjście (styk)	CS0	licznik C0 – wyjście (styk)

Monitor sygnałów wykorzystuje dwie listy monitorowanych sygnałów obiektowych i zmiennych systemowych (urządzeń), umownie określanych jako lista wejściowa i lista wyjściowa. Wartości sygnałów z obu list są cyklicznie odczytywane za pomocą funkcji m_ActEasyIF.GetDevice(), natomiast wartości sygnałów z listy wyjściowej mogą być też ustawiane funkcją m_ActEasyIF.SetDevice().

Obie listy są definiowane w pojedynczym pliku konfiguracyjnym za pomocą czterech instrukcji:

WED=DeviceName Komentarz
WEH=DeviceName Komentarz
WYD=DeviceName Komentarz
WYH=DeviceName Komentarz

gdzie: DeviceName – określa monitorowaną zmienną, Komentarz – to tekstowy opis sygnału wyświetlany obok jego aktualnej wartości.

W nazwach instrukcji przed znakami „=" występują dodatkowe oznaczenia literowe D i H. Określają one czy wartość danej zmiennej systemowej ma być prezentowana w postaci liczby dziesiętnej (D) czy liczby szenastkowej (H, hexadecymalnej). Separatorem między parametrami DeviceName i Komentarz jest pierwszy znak spacji lub znak tabulacji występujący po znaku „=" (parametr DeviceName nie może zatem zawierać żadnego z tych znaków).

Kolejność, w jakiej aktualne wartości sygnałów obiektowych będą prezentowane przez program monitora odpowiada kolejności ich występowania w pliku konfiguracyjnym. Interpreter tego pliku nie nakłada żadnych ograniczeń, co do obecności tego samego sygnału na obu listach, ani jego kilkakrotnej obecności na tej samej liście. W założeniach ma to ułatwić wygodniejszą prezentację informacji, np. gdy wartość sygnału ma być przedstawiana zarówno w postaci dziesiętnej jak i szesnastkowej, oraz synchronizację podglądu sygnałów z obu list.

Przykłady instrukcji:

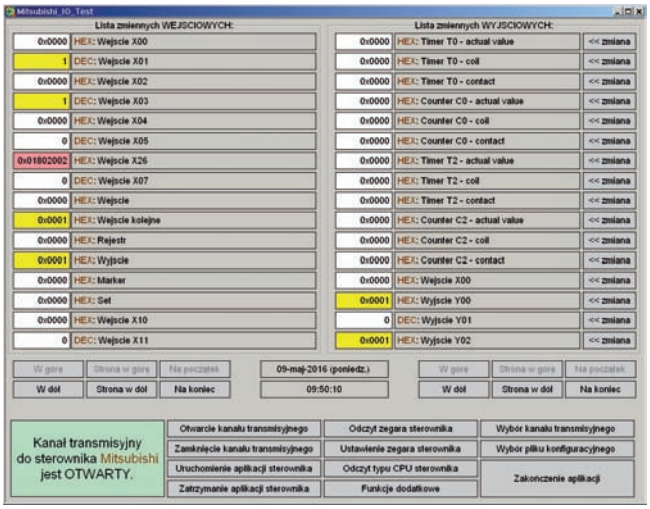
WEH=X00 wejście dwustanowe (przycisk zielony)
WYH=Y10 wyjście dwustanowe (lampka czerwona)
WYH=TN0 zawartość timera T0 zliczającego opóźnienie
WYH=TC0 timer T0 – cewka
WYH=TS0 timer T0 – wyjście (styk)
WED=D0 rejestr danych D0
WEH=D0 rejestr danych D0
WYD=D0 rejestr danych D0
WEH=M0 rejestr wskaźników M0
WEH=CN0 aktualna zawartość licznika C0
WEH=CN0 licznik C0 – cewka
WEH=CS0 licznik C0 – wyjście (styk)

3.1.2. Zmienne określające niektóre parametry modyfikujące warunki pracy programu

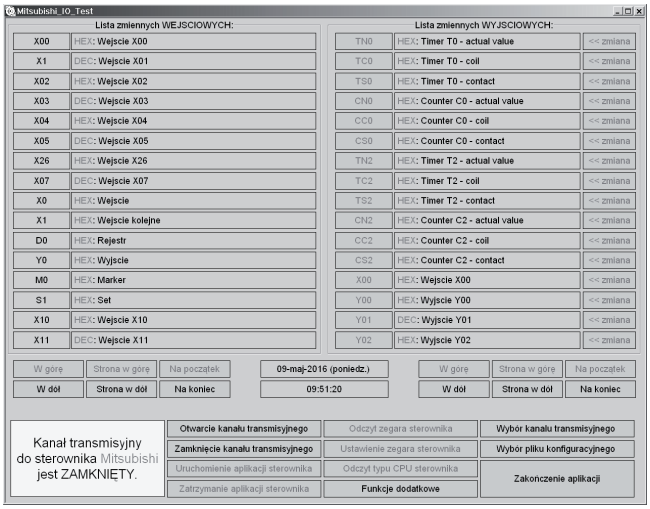
Grupa zmiennych, niezwiązanych z tworzeniem list monitorowanych sygnałów pozwala ustalać niektóre warunki pracy programu monitora. Jeśli nazwa danej zmiennej nie zostanie umieszczona w pliku konfiguracyjnym, to aplikacja przyjmie jej wartość domyślną.

CzasWyswietlaniaKomunikatuInformacyjnego=xxxx
— czas wyświetlania niektórych komunikatów informacyjnych wyrażony w sekundach. Wartość parametru „xxxx" musi zawierać się w granicach od 0 do 60, wartością domyślną jest 5. Jeśli za wartość parametru przyjęto 0, to komunikat musi zostać potwierdzony przez operatora.

CzasWyswietlaniaKomunikatu_oBledzie=xxxx
— czas wyświetlania komunikatów o błędach fatalnych wyrażony w sekundach. Wartość parametru „xxxx" musi zawierać się w granicach od 0 do 60, wartością domyślną jest 15. Jeśli



Rys. 4. Główne okno dialogowe programu monitora przy otwartym kanale transmisyjnym
Fig. 4. The main dialog box of the monitor program with an open transmission channel



Rys. 5. Główne okno dialogowe programu monitora przy zamkniętym kanale transmisyjnym
Fig. 5. The main dialog box of the monitor program with a closed transmission channel

za wartość parametru przyjęto 0, to komunikat musi zostać potwierdzony przez operatora.

MitsubishiCpuType=zzzzzzzzzz

— zmienna wykorzystywana do weryfikacji typu jednostki centralnej sterownika Mitsubishi współpracującej z programem monitora sygnałów. Weryfikacja polega na odczytaniu ze sterownika typu CPU i porównaniu z wzorcem. Parametr „zzzzzzzzzz", będący ciągiem znakowym o długości nieprzekraczającej 20 znaków określa wzorzec, wartością domyślną jest Q02HCPU. Błędny wynik weryfikacji nie ma wpływu na wykonywanie aplikacji, powoduje tylko wyświetlenie dodatkowego komunikatu, który ma zwrócić uwagę na niezgodność deklaracji ze stanem faktycznym i przyjrzenie się innym parametrom wykorzystywanym w procedurach współpracy aplikacja–sterownik.

NumerKanaluTransmisyjnego=xx

— aby współpraca monitora sygnałów z jednostką centralną sterownika PLC była możliwa, należy zdefiniować kanał transmisyjny wykorzystując aplikację Communication Setup Utility. Podczas definiowania kanału określa się m.in. jego numer. Z tak zdefinowanego kanału może korzystać dowolna aplikacja, która odwołując się do tego kanału podaje tylko jego numer. Zmienna NumerKanaluTransmisyjnego pozwala na sparametryzowanie numeru kanału wykorzystywanego przez monitor sygnałów. Parametr „xx" może być liczbą całkowitą z zakresu od 1 do 1023, wartością domyślną jest 1. Program monitora sygnałów ma wbudowaną funkcję umożliwiającą zmianę numeru kanału bez potrzeby modyfikacji pliku konfiguracyjnego.

3.1.3. Wybór pliku konfiguracyjnego i interpretacja jego zawartości

Pliki konfiguracyjne mogą mieć dowolną nazwę i rozszerzenie, a wybór pliku następuje po wybraniu odpowiedniego przycisku w głównym polu dialogowym aplikacji. Dodatkowo, podczas restartu programu monitora przeglądana jest zawartość katalogu, w którym zapisano program w poszukiwaniu pliku noszącego tę samą nazwę co on sam, ale o rozszerzeniu .ini (tu Mitsubishi_IO_Test.ini). Jest to plik konfiguracyjny, którego zawartość jest automatycznie interpretowana. Jeśli we wskazanej lokalizacji brak jest takiego pliku, albo zawiera błędy, to aplikacja przyjmuje domyślne wartości niektórych parametrów, ale listy monitorowanych sygnałów nie będą zawierać żadnych elementów. Dotyczy to także plików ręcznie wskazanych przez operatora. Interpretacja zawartości pliku konfiguracyjnego prowadzona jest aż do ostatniego wiersza, ale w przypadku wykrycia błędu program kończy analizę na wierszu, w którym znaleziono błąd. W takim przypadku aplikacja wyświetla komunikat zawierający także numer wiersza.

3.2. Okno dialogowe monitora sygnałów obiektowych i zmiennych systemowych sterowników Mitsubishi

Główne okno dialogowe monitora sygnałów obiektowych i zmiennych systemowych pokazano na rys. 4 i rys. 5. W oknach tych można wyróżnić trzy części:

— w dwóch skrajnych kolumnach znajdujących się w lewej, górnej części okna prezentowane są informacje dotyczące sygnałów i zmiennych systemowych (urządzeń) z listy wejściowej. Lewa kolumna zawiera ich aktualne wartości odczytane ze sterownika, prawa kolumna – komentarze.

— w kolejnych dwóch kolumnach w podobny sposób aplikacja wyświetla informacje dotyczące sygnałów i zmiennych systemowych (urządzeń) z listy wyjściowej. Po prawej stronie kolumny zawierającej komentarze umieszczono szereg przycisków opisanych jako „<< zmiana". Ich funkcje zostaną opisane w dalszej części tego artykułu.

— w dolnej części okna dialogowego znajduje się dziesięć przycisków służących do obsługi sterownika, a w lewym dolnym rogu umieszczono okienko, w którym aplikacja wyświetla aktualny stan kanału transmisyjnego do sterownika Mitsubishi.

Każda lista może zawierać od 0 do 1000 elementów, ale w polu dialogowym jest miejsce na wyświetlenie informacji dotyczących tylko 16 elementów z każdej listy. Listy niezależnie od siebie można przewijać za pomocą dwóch grup przycisków, po 6 przycisków w grupie.

Wymiana informacji ze sterownikiem jest możliwa tylko wtedy, gdy otwarty jest kanał transmisyjny komputer–sterownik. Automatyczne otwarcie kanału ma miejsce podczas restartu programu monitora albo może zostać ręcznie wymuszone przez operatora – po wybraniu stosownego przycisku w głównym oknie dialogowym aplikacji. Ręczna zmiana kanału lub wskazanie innego pliku konfiguracyjnego powoduje automatycznie zamknięcie otwartego kanału transmisyjnego. Jeśli kanał transmisyjny jest zamknięty (rys. 5), to:
- zawartości obu list nadal można przeglądać, ale zamiast aktualnych wartości sygnałów i zmiennych program wyświetla parametry DeviceName tych elementów obu list, które mieszczą się w polu dialogowym.
- zablokowane są funkcje zmiany wartości sygnałów i zmiennych systemowych, a także większość funkcji obsługi sterownika.

3.2.1. Odczyt i sposób prezentacji wartości sygnałów obiektowych i zmiennych systemowych odczytanych ze sterownika

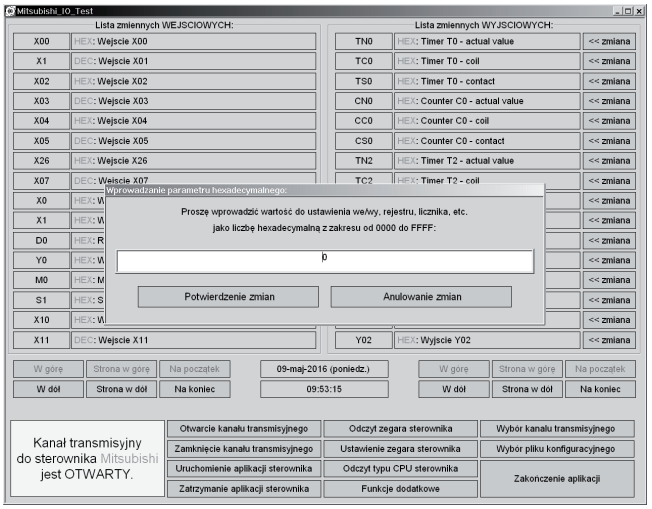
Odczyt sygnałów obiektowych i zmiennych systemowych ze sterowników Mitsubishi program monitora realizuje cyklicznie, z przerwą 10 ms między kolejnymi cyklami odczytu. Ponieważ obie listy sygnałów i zmiennych łącznie mogą zawierać do 2000 elementów, odczyt tylu wartości ze sterownika w trybie pytanie – odpowiedź powodowałby, że aktualizacja informacji postępowałaby bardzo wolno i była mało przydatna. Dlatego każdy cykl odczytu dotyczy tylko tych elementów z każdej listy, które aktualnie są prezentowane w głównym oknie dialogowym aplikacji.

Wyświetlanie informacji o wartościach sygnałów i zmiennych systemowych jest realizowane w dwóch kolumnach okien. W zależności od ustawienia w pliku konfiguracyjnym każda wartość prezentowana jest jako liczba hexadecymalna z przedrostkiem „0x” lub jako liczba dziesiętna. Ponadto jej komentarz także jest poprzedzony określeniem „HEX:” lub „DEC:” (rys. 4 i 5). Dla ułatwienia podglądu, jeśli aktualna wartość sygnału lub zmiennej systemowej jest większa od 0 to zostaje wyświetlona na żółtym tle, w przeciwnym razie na tle białym.

Funkcja odczytu m_ActEasyIF.GetDevice() wykorzystywana w programie może zgłaszać błąd. Najczęstszą jego przyczyną jest nieprawidłowa definicja parametru DeviceName instrukcji WED, WEH, WYD, WYH w pliku konfiguracyjnym albo parametr formalnie zdefiniowano poprawnie, ale sterownik fizycznie nie ma dostępu do odpowiedniego sygnału. Wówczas zamiast aktualnej wartości sygnału lub zmiennej systemowej na czerwonym tle wyświetlany jest kod błędu w postaci ośmiocyfrowej liczby hexadecymalnej. Pełną listę kodów błędów wraz z ich opisem i instrukcją usuwania zamieszczono w literaturze [1 (rozdz, 6), 3 (rozdz. 7)].

3.2.2. Zmiana wartości sygnałów obiektowych i zmiennych systemowych

Program monitora umożliwia zmianę wartości wskazanego sygnału obiektowego lub zmiennej systemowej z listy wyjściowej. Jednak wysyłanie nowej wartości do sterownika nie odbywa się cyklicznie, jak w przypadku odczytu, ale jest wykonywane tylko raz – po potwierdzeniu zmiany. Należy pamiętać, że:
- w przypadku sygnałów wejściowych zmiana nie polega na fizycznym wysterowaniu wejścia, tylko na zmianie zawartości rejestru przechowującego stan wejścia; zawartość takiego rejestru jest cyklicznie uaktualniana przez oprogramowanie systemowe sterownika, zmiana wykonana programem monitora od razu zostaje skorygowana przez to oprogramowanie,



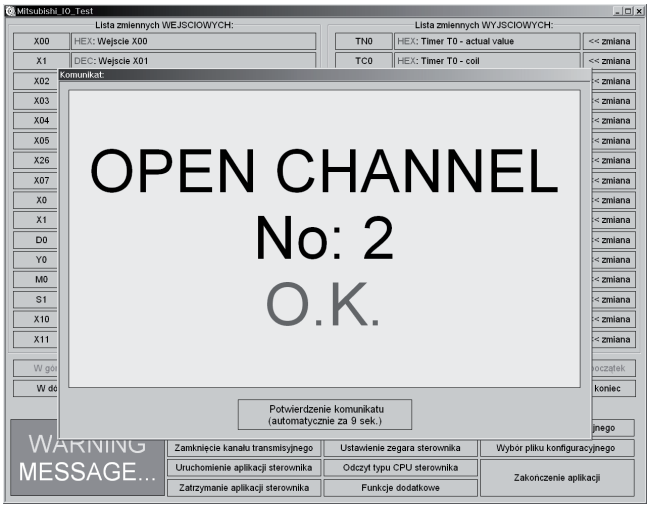
Rys. 6. Okienko dialogowe do wprowadzania parametru w postaci liczby szesnastkowej
Fig. 6. A dialog box for entering a parameter in the form of a hexadecimal number

- w przypadku sygnałów wyjściowych program monitora zmienia tylko zawartość rejestru pośredniczącego; jeśli program aplikacyjny sterownika nie zawiera instrukcji wysterowującej dane wyjście lub zawiera taką instrukcję, ale nie został ustawiony w stan RUN, to wyjście będzie wysterowane zgodnie z nastawą wykonaną programem monitora,
- w przypadku niektórych zmiennych systemowych sterownika, np. dotyczących timerów lub liczników funkcja zapisu m_ActEasyIF.SetDevice() nie zgłosi błędu, ale wartość zmiennej nie ulegnie zmianie.

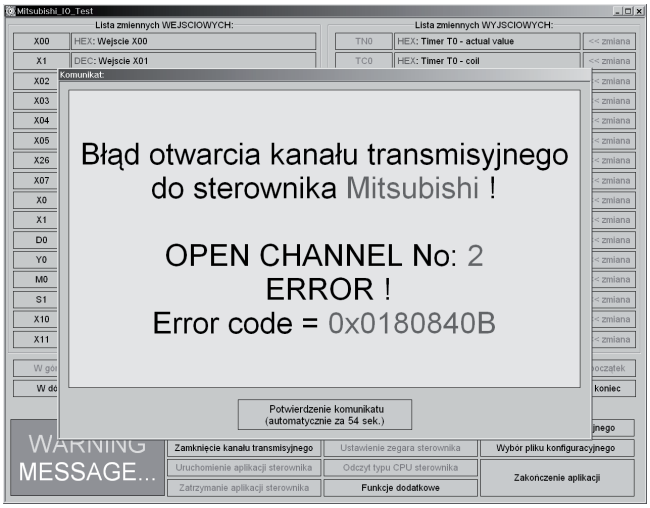
Zmianę wartości sygnału obiektowego lub zmiennej systemowej można wykonać dwoma sposobami:
- klikając wskaźnikiem myszki okno z informacją o aktualnej wartości wybranego sygnału lub zmiennej systemowej. Ta metoda ma jednak ograniczenie, wynikające z tego, że jest to zmiana zero-jedynkowa. Jeśli aktualna wartość obiektu jest różna od 0 (True), to zostanie ona wyzerowana (False), jeśli wartość jest równa 0 (False), to zostaje wysterowana (True). Wysłanie przesyłki do sterownika następuje bezpośrednio po kliknięciu w okienko i nie wymaga dodatkowego potwierdzenia. Metoda jest szczególnie przydatna do testowania połączeń sterownik–urządzenie wykonawcze, np. do sterowania zaworami.
- klikając wskaźnikiem myszki jeden z przycisków „<< zmiana”. Powoduje to otwarcie dodatkowego okna dialogowego (rys. 6), umożliwiającego wprowadzenie nowej wartości. W zależności od ustawienia w pliku konfiguracyjnym dotyczącym wskazanego obiektu nową wartość można wprowadzać albo w postaci liczby szesnastkowej bez przedrostka „0x”, albo w postaci liczby dziesiętnej. Wysłanie przesyłki do sterownika następuje po potwierdzeniu zmiany (przycisk „Potwierdzenie zmiany”).

3.3. Obsługa serwisowa sterownika

Do celów uruchomieniowo-diagnostycznych oraz jako wsparcie normalnej eksploatacji stanowiska ze sterownikiem Mitsubishi aplikację monitora stanu sygnałów i zmiennych systemowych wyposażono w niektóre funkcje do jego obsługi serwisowej. Umożliwiają one wykonanie następujących czynności:
- otwarcie i zamknięcie kanału transmisyjnego do sterownika,



Rys. 7. Komunikat potwierdzający prawidłowe otwarcie kanału transmisyjnego
Fig. 7. The message confirming the correct opening of the transmission channel



Rys. 8. Komunikat o błędzie otwarcia kanału transmisyjnego
Fig. 8. Error message opening the transmission channel

– uruchomienie i zatrzymanie wykonywania aplikacji sterownika, a więc zmianę jego trybu pracy RUN → STOP i STOP → RUN,
– odczyt i ustawianie nastaw zegara i kalendarza sterownika. Ustawienie zegara i kalendarza sterownika polega na wysłaniu wskazań zegara i kalendarza współpracującego komputera PC,
– odczyt typu jednostki centralnej,
– zmiana numeru kanału transmisyjnego do sterownika,
– wybór innego pliku konfiguracyjnego.

Funkcje wymienione w punktach od 2 do 4 można realizować tylko wtedy, gdy pozostaje otwarty kanał transmisyjny do sterownika. W przypadku zmiany numeru kanału i wyboru innego pliku konfiguracyjnego zawsze następuje automatyczne zamknięcie otwartego kanału (trzeba go ręcznie otworzyć). Na rys. 7 i 8 pokazano przykładowe komunikaty generowane przez funkcję otwierającą kanał transmisyjny do sterownika.

4. Podsumowanie

Jednym z obszarów działalności Przemysłowego Instytutu Automatyki i Pomiarów PIAP jest realizacja zleceń z przemysłu na budowę stanowisk, których elementami są sterowniki PLC, komputery przemysłowe i roboty. Ze względu na krótkie terminy realizacji różne prace, m.in. kompletowanie mechaniki, połączenia elektryczne i sygnałowe, testowanie i regulacja stanowiska, tworzenie oprogramowania aplikacyjnego są zwykle prowadzone równolegle. Dużym ułatwieniem są programowe narzędzia diagnostyczne ułatwiające podgląd sygnałów obiektowych i zmiennych związanych z programami użytkowymi. W przypadku użycia elementów sterujących firmy Mitsubushi, można zastosować opisany monitor stanu sygnałów i zmiennych systemowych. Program ten powstał w 2016 r. jako wsparcie prac rozwojowych podczas realizacji linii produkcyjnej matryc ciekłokrystalicznych urządzenia do testów medycznych. Jego zaletą jest łatwa konfiguracja podglądu i czytelne monitorowanie sygnałów. Stosowanie programu nie wymaga dodatkowej wiedzy z zakresu programowania sterowników PLC ani narzędzi do tworzenia takich aplikacji. Nie grozi też wprowadzeniem przypadkowych modyfikacji do działającego programu sterownika. Wadą monitora pozostaje to, że wymaga zainstalowania komercyjnego oprogramowania MX Components firmy Mitsubishi. Jednak MX Components, w porównaniu z np. GX Works2 zajmuje mniej zasobów komputera PC, a obsługa zawartych w nim aplikacji jest bardzo prosta.

Należy podkreślić duże możliwości tworzenia własnych aplikacji na bazie elementów MX Components. Oprogramowanie to, szczególnie w najnowszej wersji 4.x, umożliwia obsługę nie tylko sterowników PLC, ale także innych urządzeń automatyki, ze sterownikami robotów włącznie.

Bibliografia

1. Mitsubishi Electric Corporation 2002 – MX Component Version 3 Programming Manual.
2. Mitsubishi Electric Corporation 2002 – MX Component Operating Manual.
3. Mitsubishi Electric Corporation 2012 – MX Component Version 4 Programming Manual.
4. Microsoft Visual C++ 6.0 MFC Library Reference, Microsoft Press 1998.
5. Bates J., Tompkins T., *Poznaj Visual C++*, Wydawnictwo MIKOM, 1999.
6. Petzold Ch., *Programowanie Windows*, Wydawnictwo RM Warszawa 1999.
7. Leinecker R.C., Archer T., *Visual C++ 6 Vademecum profesjonalisty*, Wydawnictwo HELION 2000.
8. Dunaj J., Stempniak P., Syryczyński A., *Urządzenie testujące mechanizmy sprężynowe do napędu rozłączników średnich napięć*, „Pomiary Automatyka Robotyka”, Vol. 14, Nr 9/2010, 68–72.
9. Dunaj J., *Opis działania, konfiguracji i obsługi aplikacji „Renus” do sterowania robotami rehabilitacyjnymi Renus-1 i Renus-2*, materiały Przemysłowego Instytutu Automatyki i Pomiarów PIAP, czerwiec 2014.
10. Dunaj J., Klimasara W., *Rozwiązania sprzętowe i programowe w sterowaniu robotami rehabilitacyjnymi Renus*, „Pomiary Automatyka Robotyka”, Vol. 18, Nr 12/2014, 100–115, DOI: 10.14313/PAR_214/100.

Programmable Monitor of Signal Object's and System Variables States for PLC Controllers

Abstract: During development of post industrial containing PLC controllers, robots and industrial PC a great convenience is the ability to view the status signals, object and variable of application programs. Manufacturers of controls in its software for creating applications often provide additional tools to perform these functions. An interesting solution is the MX Components software of Mitsubishi Electric, which offer not only the ability to see what is happening in the controllers, but also providing tools to create their own diagnostic program. This article contains information as the basis of the elements of the MX Components to create such a program. The possibilities of functional programmable monitor signal states object and system variables Mitsubishi controllers are presented. This monitor was built in PIAP in order to facilitate the testing and diagnosis of industrial applications performed at the Institute.

Keywords: PLC, PC application, software, view the status of object signals and system variables

mgr inż. Jacek Dunaj
jdunaj@piap.pl

W 1980 r. ukończył studia na Wydziale Elektrycznym Politechniki Warszawskiej, od 1985 r. jest zatrudniony w Przemysłowym Instytucie Automatyki i Pomiarów PIAP. Specjalizuje się w programowaniu mikroprocesorów, kontrolerów, sterowników i robotów przemysłowych, systemów wizyjnych a także komputerów PC programowanych w języku asemblera i C/C++ w środowisku różnych systemów operacyjnych. Współautor oprogramowania kilku urządzeń opracowanych w PIAP oraz wielu wdrożeń przemysłowych, w szczególności wymagających współpracy różnych urządzeń automatyki i wykorzystania oprogramowania biurowego (baz danych, arkuszy kalkulacyjnych).

mgr inż. Dariusz Grabowski
dgrabowskij@piap.pl

W 2001 r. ukończył studia na Wydziale Mechatroniki Politechniki Warszawskiej. Karierę zawodową rozpoczął w służbach utrzymania ruchu w przemyśle. Jednocześnie prowadził działalność dydaktyczną w Zespole Szkół Technicznych w Legionowie. Pracę w Przemysłowym Instytucie Automatyki i Pomiarów PIAP podjął w 2002 roku. Na stanowisku konstruktora. W latach 2006–2007 pracował w firmie konsultingowej Epsilon świadczącej usługi doradztwa technicznego w Szwecji, Danii i Norwegii. Od 2010 r. kierownik Zespołu Zrobotyzowanych Instalacji Przemysłowych PIAP. Specjalizuje się w projektowaniu konstrukcji mechanicznych aplikacji przemysłowych wykorzystując oprogramowanie CAD. Współautor i Kierownik Projektu kilkudziesięciu wdrożeń przemysłowych Instytutu, gdzie zastosowano sterowniki PLC i roboty przemysłowe.