

Karol GRANDEK, Grzegorz DYSARZ, Robert POLANIK

ANALIZA SYSTEMÓW PROGRAMOWANIA STEROWNIKÓW PLC WG NORMY IEC 1131-3

Streszczenie. W pracy przedstawiono procedurę pozwalającą na zbadanie zgodności dowolnych narzędzi programowania sterowników PLC z normą IEC 1131-3. Zwrócono również uwagę na samą normę pokazując genezę jej powstania oraz części składowe. Dokonano także przykładowej analizy systemu programowania ACCON Prosys 1131.

ANALYSIS OF PLC'S PROGRAMMING SYSTEMS WITH THE NORM IEC 1131-3

Summary. Dynamic evolution of microprocessor technic's causes the spread of Programmable Logic Controllers in applied automatic. In this paper European Norm IEC 1131-3, which relates to programming languages of PLC, is considered. At the beginning we present the genesis IEC 1131-3 and its composition. Test tables and test algorithm, verifying accordance with the norm are also presented. However, the greatest interest is put into the comparison of ACCON Prosys 1131 with the norm.

1. Rozwój narzędzi programowania sterowników PLC

W roku 1969 w USA wprowadzono do przemysłu samochodowego Programowalne Sterowniki Logiczne (PLC) [12] Poprawiło to niezawodność i umożliwiło dokonywanie zmian algorytmów sterowania procesami technologicznymi.

Sterowniki PLC były programowane na niskim poziomie, przy użyciu kodu numerycznego lub assemblera. Ponieważ takie programowanie wymagało specjalistycznej wiedzy, było ono bardzo uciążliwe dla użytkowników tych urządzeń. Podstawowym językiem programowania była alfanumeryczna reprezentacja kodu numerycznego, zwana Listą Instrukcji. Programo-

wanie odbywało się za pomocą programatora ręcznego „Hand Held Programmer - HHP”. Pozwalał on na wprowadzanie i uaktualnianie programu ON LINE bez przerywania jego działania.

Ważnym krokiem w rozwoju programowania sterowników logicznych było pojawienie się w 1982 komputera osobistego IBM PC. Duża moc obliczeniowa tego urządzenia pozwoliła na wykorzystanie przez narzędzia programowania środowiska graficznego bardziej przyjaznego dla programisty.

Pod koniec lat 80. stało się oczywiste, że należy ujednoczyć narzędzia programowania w taki sposób, aby były zgodne z różnymi rodzajami sterowników pochodzących od różnych producentów (do tej pory każdy producent opracowywał swoje narzędzie programowania, które nie było kompatybilne z narzędziami innych producentów). Rezultatem tego było opracowanie trzeciej części międzynarodowej normy IEC 1131. Norma zaleca, żeby firmy zajmujące się projektowaniem i produkcją sterowników mikroprocesorowych zapewniały możliwość programowania sterowników w pięciu różnych językach programowania. Standard nie wymaga, by wszystkie modele sterowników obsługiwały wszystkie pięć języków, lecz konieczne jest, aby więcej niż jeden język programowania był obsługiwany przez dany sterownik. Ideą standaryzacji oprogramowania jest to, aby dany typ aplikacji mógł być przetwarzany przez różne narzędzia projektowania oprogramowania. Potęgą standardu jest natomiast to, że pozwala on łączyć różne metody programowania i używać ich na wspólnej platformie jednego programu [2].

2. Charakterystyka normy IEC 1131-3

Norma IEC 1131-3 została zatwierdzona przez Europejski Komitet Norm Elektrotechnicznych (CENELEC) 06.07 1993 roku. W normie tej korzystano z następujących norm pomocniczych:

IEC 50	IEV - International Electrotechnical Vocabulary
IEC 559:1989	Binary floating-point arithmetic for microprocessor systems
IEC 617-12:1991	Graphical symbols for diagrams -Part 12:Binary logic elements
IEC 617-13:1978	Graphical symbols for diagrams-Part 13:Analogue elements
IEC 848:1988	Preparation of function charts for control systems
ISO/AFNOR:1989	Dictionary of computer science, ISBN 2-12-486911-6
ISO/IEC 646:1991	Information technology - ISO 7-bit coded character set for information processing interchange
ISO 8601:1988	Data elements and interchange formats - Information interchange - Representations of dates and times

ISO 7185:1990	Information technology - Programming languages - Pascal
ISO 7498:1984	Information processing systems - Open systems interconnection - Basic reference model

Norma podzielona jest na 4 rozdziały i 9 dodatków. W rozdziale 1 podano obszar zastosowania normy i założenia, definicje podstawowych pojęć, opisano modele programowy i komunikacyjny systemu oraz model programu. Scharakteryzowano również pojęcie stopnia zgodności systemów PLC z normą. W rozdziale 2 podano elementy wspólne języków programowania, w tym alfabet, jednostki leksykalne, zewnętrzny format danych, typy danych (elementarne, pochodne). Następnie opisano zmienne, jednostki organizacyjne programu, elementy języka SFC (sekwencyjnej sieci działań) takie, jak: kroki, tranzycje, czynności oraz reguły wykonywania programu SFC, wymagania co do stopnia spełnienia normy przez implementację SFC. Rozdział ten kończy opis elementów konfiguracyjnych. Rozdział 3 opisuje języki tekstowe: listę instrukcji oraz tekst strukturalny. Rozdział 4 opisuje języki graficzne: język drabinkowy oraz język funkcyjny logicznych, ich elementy wspólne oraz sposób i kolejność wartościowania.

Dodatki A-E zawierają zalecenia normatywne dotyczące syntaktyki i semantyki języków tekstowych, używanych znaków graficznych i słów kluczowych, parametrów zależnych od implementacji oraz opisów przyczyn błędów.

Dodatki F, G, H zawierają zalecenia informacyjne takie, jak: przykłady, alfabetyczny spis haseł, programowy test wypełnienia normy[3].

3. Badanie zgodności systemów programowania sterowników PLC z normą IEC 1131-3 za pomocą tablic sprawdzających

Przy wyborze odpowiedniego systemu programowania brakuje często istotnych dla użytkownika wiadomości, jakimi są dane o wymaganiach sprzętowych, możliwości pracy i sterowania w sieci, dostępnych funkcjach, ich zakresie, możliwościach wymiany danych, rozbudowy systemu oraz o elastyczności całego systemu w sensie dostępności różnych środowisk programowania. Dla potencjalnego użytkownika, szukającego właściwego dla swoich potrzeb systemu, mogą być pomocne tzw. tablice sprawdzające. W tablicach tych, oprócz określenia ważności (dla użytkownika) poszczególnych elementów i funkcji, zamieszczono pozycję określającą stopień zgodności danej cechy (funkcji) z normą IEC 1131-3. Podstawą do wprowadzenia jednolitej normy odnośnie do systemów programowania sterowników PLC było zwiększające się podobieństwo obecnych systemów. To pozwala łatwiej oceniać kompleksowo różne systemy od strony użytkownika. Rośnie znaczenie analizy wszelkiego rodzaju odstępstw, ograniczeń, rozszerzeń i odchyień od przyjętego standardu. Z punktu widzenia użyt-

kownika ocena kompleksowa systemu ma sens, jeśli chodzi o sposób definiowania oraz wykorzystania funkcji oferowanych przez system.

Norma IEC 1131-3 standaryzuje wszystkie języki programowania systemów programowania sterowników PLC. Realizacja języków programowania przez dany system wywiera istotny wpływ na cechy sprawności i wydajności wszystkich komponentów systemowych oraz tworzy skalę określającą zdolności wszystkich systemów programowania. Tablice sprawdzające bazują na zaleceniach normy IEC 1131-3, takich jak konwersja języków programowania oraz metody programowania w rzeczywistym środowisku. Tablice sprawdzające mogą być oczywiście przez indywidualnego użytkownika uzupełnione innymi istotnymi dla niego elementami [2].

3.1. Tablice sprawdzające

Użytkownik ma do dyspozycji 16 tablic, dzięki którym może określić stopień zgodności systemu programowania z normą IEC 1131-3. Tablice te dotyczą następujących własności systemu:

- tablica zgodności: stopień wypełnienia normy,
- cechy zgodności z IEC 1131-3,
- zestaw realizowanych języków oraz badanie uzyskanej funkcjonalności,
- możliwość wzajemnej konwersji języków programowania,
- edytory i narzędzia tłumaczenia języków programowania,
- narzędzia do konfiguracji,
- narzędzia do testowania i uruchamiania,
- zarządzanie projektem,
- dokumentacja,
- dalsze narzędzia: symulacje i analiza logiczna,
- realizacja narzędzi,
- instalacja narzędzi,
- środowisko pracy systemu programowania,
- otwartość systemu,
- jakość dostarczanych użytkownikowi dokumentów,
- koszty zakupu i konserwacja systemu.

W zależności od dostępnych informacji na temat sprawdzanego systemu użytkownik wybiera spośród ww. tablic tylko te, które jest w stanie wypełnić. Pierwsza z ww. tablic, tzn. tablica zgodności: Stopień wypełnienia normy, może mieć postać jak tabela 1.

Tabela 1

Tablica zgodności: Stopień wypełnienia normy

Funkcja / Cecha	Waga	Zgodność	Punkty
System ma opracowane tablice cech zgodności z normą IEC 1131-3 oraz ich pełny opis.			
System posiada pełną dokumentację własności zależnych od implementacji.			
Dokumentacja zawiera pełny opis własności systemu nie objętych normą.			
Dokumentacja zawiera opis błędów (i komunikatów z nimi związanych), jakie mogą powstać przy tworzeniu programu użytkownika.			
System reaguje w odpowiedniej formie na błędy.			
System definiuje formalną syntaktykę wszystkich elementów języków tekstowych.			
System posiada certyfikacje organizacji niezależnych (np. org. PLCopen).			
Występujący zakres ograniczeń względnie odchyleń od normy jest możliwy do zaakceptowania.			
Cechy inne (wprowadzone przez użytkownika)			
Wartość jednostkowa tablicy			
Ważność tablicy			

3.2. Korzystanie z tablic sprawdzających

Tablice sprawdzające służą do obiektywnej oceny oraz porównania tych samych właściwości różnych systemów programowania. Przed przystąpieniem do oceny systemów należy przyjąć pewne ustalenia. Następnie, postępując wg poniższych wskazówek, można przystąpić do oceny systemu programowania:

1. Wybrać wszystkie interesujące użytkownika tablice (wg indywidualnych wymagań co do porównywanych systemów).

- Ustalić cechy systemu, istotne dla użytkownika w kontekście przyszłej funkcji systemu programowania.

2. Określić wagę (priorytety) poszczególnych tablic.

- Należy ustalić ważność, z punktu widzenia użytkownika, każdej tablicy zawierającej funkcje i cechy systemu ułożone tematycznie, np.: gdy nie grają roli koszty zakupu i konserwacji systemu, możemy przydzielić tablicy kosztów wartość 1 (jest to najniższa wartość, jaką możemy przydzielać). Dopuszczalne jest oczywiście przydzielenie takich samych wartości różnym tablicom. Wartości te należy wpisać na końcu tablicy w rubrykę „Ważność tablicy”.

3. Oszacować wagę kolejnych cech i funkcji w ramach każdej tablicy.

- Należy tutaj ustalić w ramach każdej tablicy, które z cech mają dla użytkownika systemu największe znaczenie. Przy pozycjach, które (z punktu widzenia funkcji, jaką ma spełniać system) nie mają znaczenia, wpisujemy wartość „0”, natomiast przy punktach najistotniejszych piszemy liczbę „10”. To są oczywiście skrajne przypadki. Liczby od „1” do „9” przydzielamy odpowiednio wg naszej oceny.

4. Ocenić stopień realizowalności danych pozycji tablicy dla każdego porównywanego systemu programowania.

- Po zapoznaniu się z opisem systemu, wersją demonstracyjną lub pełną, należy w tym punkcie ocenić stopień realizowalności wybranej funkcji (lub cechy) przez dany system. Ocena zawiera się w przedziale liczbowym od „0” do „10” i tak: wartość „0” przydzielamy w przypadku, gdy system nie realizuje danej funkcji (cechy) w ogóle, natomiast wartość „10” przydzielamy pozycji w pełni realizowanej przez dany system. W sytuacjach częściowej realizacji interesującej nas cechy przydzielamy wartości od „1” do „9”. Wartości te wpisujemy w rubrykę „Zgodność” poniższych tablic.

5. Określić zależność ważności każdego elementu tablicy od stopnia jego realizowalności w ramach tego systemu.

- Aby określić wartość współzależności między wagą cechy a jej realizowalnością przez system, należy pomnożyć obie te wartości (waga* zgodność), a otrzymany wynik podzielić przez 10. W ten sposób otrzymaną wartość wpisujemy w rubrykę „Punkty” odpowiedniej tablicy. W ramach każdej tablicy należy następnie dodać wartości z rubryki „Punkty” oraz podzielić ich sumę przez liczbę pozycji sumowanych (obliczyć średnią wartość arytmetyczną) i tak otrzymany wynik wpisać w rubrykę „Wartość pojedynczej tablicy”. Według tej pozycji możemy dokonywać oceny przydatności danej grupy cech i funkcji każdego systemu w kontekście postawionych wcześniej wymagań (wartość określająca ważność danej tablicy).

6. Obliczyć pojedynczą wartość dla kolejnych systemów.

- Po wcześniej wymienionych operacjach należy obliczyć iloczyn wartości liczbowych umieszczonych w rubrykach „Ważność tablicy” oraz „Wartość pojedynczej tablicy”. W ten sposób można określić, który z testowanych systemów jest najodpowiedniejszy (największy iloczyn) dla określonego użytkownika.

4. Wykorzystanie tablic sprawdzających dla badania zgodności z normą IEC 1131-3 przykładowego systemu

Do analizy zgodności systemu programowania ACCON-Prosyls 1131 [1], [2] z normą IEC 1131-3 wybrano następujące tablice:

- cechy zgodności z IEC 1131-3,
- możliwość konwersji języków programowania,
- edytory i narzędzia tłumaczenia języków programowania,

- narzędzia do testowania i uruchamiania,
- dalsze narzędzia: symulacje i analiza logiczna,
- realizacja narzędzi,
- otwartość systemu.

Sposób wykorzystania tabeli sprawdzających przedstawiono na przykładzie tablicy „Edytory i narzędzia tłumaczenia języków programowania” (tab.2).

Tabela 2

Edytory i narzędzia tłumaczenia języków programowania

Funkcja / Cecha	Ważność	Zgodność	Punkty
Edytory graficzne z dowolnością położenia elementów	7	5	3,5
Edytory graficzne z automatycznym ustalaniem położenia elementów	6	10	6
Automatyczne naprowadzanie linii łączących („tasiemka gumowa”)	10	6	6
Edytory graficzne z natychmiastowym sprawdzaniem syntaktyki	8	8	6,4
Edytory tekstowe z natychmiastowym sprawdzaniem syntaktyki	6	0	0
Możliwość zmiany tekstu źródłowego za pomocą edytora standardowego ASCII	6	10	6
Wspomaganie systemowe przy wprowadzaniu bloków funkcyjnych i wywoływaniu funkcji (wyświetlenie listy dostępnych bloków funkcyjnych, przedstawienie ich parametrów formalnych)	10	10	10
Filtrowanie proponowanych informacji (operatory, funkcje, itp.) dla zasobów ustalonych w konfiguracji	6	10	6
Możliwy do przyjęcia czas reakcji na wprowadzanie danych oraz ich sprawdzenie, nawet dla dużych jednostek programowych POE (liczba instrukcji >1000)	5	10	5
Możliwy do przyjęcia czas reakcji na wprowadzanie danych oraz ich sprawdzenie, nawet dla dużych jednostek programowych POE (liczba obwodów stykowych Network > 100)	5	10	5
Możliwy do przyjęcia czas od przetłumaczenia do załadowania systemu nawet w przypadku dużych projektów	5	10	5
Inne	0	0	0
Wartość jednostkowa tablicy		5,3	
Ważność tablicy		6	

Wszystkie wartości wpisane w tablicy są wartościami, które zostały przydzielone przez konkretnego użytkownika. Inny użytkownik mógłby wpisać tu zupełnie inne wartości, w zależności od swoich wymagań stawianych badanemu systemowi programowania.

Analizę systemu opartą na wszystkich tablicach sprawdzających zamieszczono w pracy [2].

5. Zakończenie

Projektowane i produkowane po 1993 roku systemy PLC są charakteryzowane przez swych twórców jako zgodne z zaleceniami normy IEC 1131-3. Niektóre narzędzia programowania zawierają nawet w nazwie symbol „IEC 1131-3”. Jednakże do stwierdzenia faktycznej zgodności z normą upoważnione są agendy organizacji PLCopen [5]. Sama norma definiuje wymagania, jakie muszą spełniać systemy PLC oraz programy PLC, dla których można używać określenia, że są zgodne z IEC 1131-3. Dokumentacja musi zawierać między innymi deklarację: „Ten system spełnia wymagania IEC 1131-3 w zakresie następujących cech języka programowania”, po której następują tablice szczegółowo opisujące własności języka i odnośniki do poszczególnych jego własności/cech. Dla użytkownika taka deklaracja może być niewystarczająca przy wyborze systemu PLC. Pomocne mogą być wtedy tablice sprawdzające.

Biorąc pod uwagę coraz większe wymagania co do otwartości oraz elastyczności projektowanych systemów sterowania PLC oraz ich zgodności z normami europejskimi, w interesie każdego projektanta oraz producenta ww. systemów jest zapoznanie się z obowiązującymi zasadami oraz używanie narzędzi zgodnych z normą IEC 1131-3.

LITERATURA

1. ACCON-PROSYS 1131-3 - das zertifizierte IEC 1131-3 Programiersystem , DELTA-LOGIC, 1995.
2. Dysarz G., Polanik R.: Analiza systemów programowania sterowników PLC wg normy IEC 1131-3 Praca dyplomowa magisterska, Politechnika Opolska, 1997.
3. John K.-H., Tiegelkamp M.: SPS Programmierung mit IEC 1131-3. Springer-Verlag, 1995.
4. Programmable controllers - Part3: Programming languages (IEC 1131-3 : 1993) German version EN 61131-3: 1993.
5. Rabe G., Glöe G.: „Atestacja, certyfikacja i licencjonowanie systemów komputerowych pełniących funkcje związane z bezpieczeństwem.”, Informatyka, 8/1995.

6. Kwasnowski P., Wolcz K.: STYKI - przyjazne oprogramowanie sterowników PLC zgodne z normą IEC 1131-3, Pomiary Automatyka Kontrola, 8/1996.
7. Meier H., Matz B.: „Kompatibel oder normgerecht ? Die Simatic S7: Spagat zwischen Simatic 5 und IEC - norm”, Elektronik, 24/1996.
8. „Trägheitsmoment überwinden. S7 - Programmiersysteme.”: IEE Automatisierung + Datentechnik, 41, 7/1996.
9. „IEC 1131-3: Schnell in allen Sprachen.” IEE Automatisierung+Datentechnik ,40, 11/1995.
10. „Programmierung unter IEC 1131-3: Den Anschluß nicht verpassen.”, IEE Automatisierung+Datentechnik, 41, 1/1996.
11. Beyrle P.: „IEC 1131-3 und/oder Step5/7.”, IEE Automatisierung+Datentechnik, 41, 11/1996.
12. Gilles Michel "Programmable Logic Controllers Architecture and Application" John Wiley & sons,1990.

Recenzent: Dr hab.inż. Edward Hryniewicz

Wpłynęło do Redakcji 14.10. 1997 r.

Abstract

In the late 80 ies it became clear obvious that programming tools should be made uniform in the way that they should be accordance with diffrent kinds of Programmable Logic Controllers. As a resoult the norm IEC 1131 has been made.

The norm is divided into 4 chapters and 9 appendixes (see [2]).

In this paper the autors present test tables and their application to ACCON Prosys 1131 (see chapter 3).

PLC systems, which are being designed and produced after 1993 are described by their autors as corresponding to the IEC 1131 norm. However only the approved PLCopen [5] agenda can certify the actual accordance to the norm.

The norm itself defines the requirements which PLC systems and PLC programs have to fulfil.