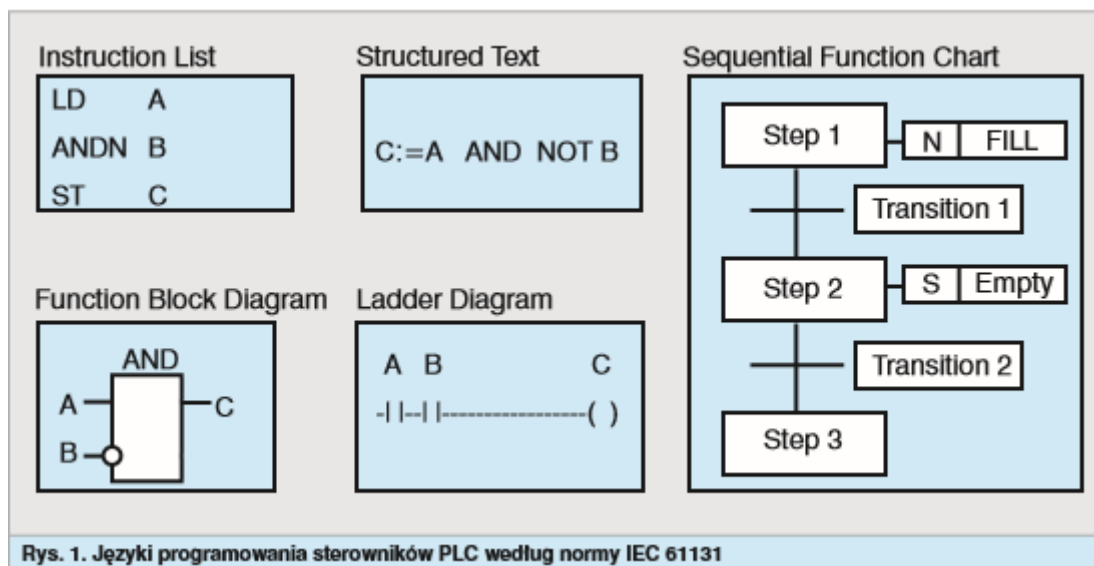


Języki programowania sterowników PLC.

Pierwszy sterownik programowalny PLC – **MODICON** (ang. **Modular Digital CONTroller**), opracowany przez niemiecką firmę AEG, pojawił się na rynku w 1969 roku. Rozpoczął się jednocześnie konkurencyjny i nie kontrolowany rozwój sprzętu i oprogramowania. Wymusiło to konieczność standaryzacji rozpoczętej w USA w latach siedemdziesiątych. Znaczącą datą jest lipiec 1992 roku, kiedy to pod auspicjami **Międzynarodowej Komisji Elektroniki** (ang. **IEC – International Electronical Commission**) opracowano **normę IEC 1131** (tab. 1).

Tab. 1. Układ normy IEC 1131		
Część nr	Nazwa	Treść
Część 1	Ogólny przegląd, definicje	<ul style="list-style-type: none">- definicje, słownik pojęć,- wykaz powiązanych i zalecanych standardów IEC,- cechy funkcjonalne PLC,
Część 2	Sprzęt (sterowniki i urządzenia peryferyjne) i zasady testowania	<ul style="list-style-type: none">- wymagania elektryczne, mechaniczne i funkcjonalne odnośnie sprzętu,- wymagania odnośnie serwisu, magazynowania i transportu,- informacje użytkowe od producentów,- metody testowania i procedury weryfikacji zgodności sprzętu,
Część 3	Języki programowania	<ul style="list-style-type: none">- model oprogramowania i komunikacji,- definicja pięciu języków programowania,- składnia logiczna (syntaktyka) i znaczenie (semantyka) elementów dwóch języków tekstowych (IL, ST) i dwóch języków graficznych (LD, FBD),- język SFC do programowania strukturalnego,
Część 4	Wskazówki dla użytkownika	<ul style="list-style-type: none">- sposób wykorzystywania pozostałych części standardu sterowników programowalnych,- wybór i wdrażanie systemów,- wymagania dotyczące zastosowań (aplikacji),
Część 5	Komunikacja, wymiana danych, bazująca na MMS (ang. <i>Manufacturing Message Specifications</i> – opis komunikatów od producentów)	<ul style="list-style-type: none">- komunikacja pomiędzy sterownikami różnych producentów,- komunikacja sieciowa,- wymiana danych,- stany alarmowe.

Norma ta, dopracowana w 1996 roku i wydana pod zmienioną nazwą – **IEC 61131**, podaje informacje ogólne i zalecenia dotyczące sprzętu, języków programowania, zasad testowania, sposobów wymiany danych, określa wytyczne dla użytkowników oraz zalecenia w zakresie uruchamiania i wdrażania systemów. Nie ma ona charakteru wiążącego i dopuszcza utrzymanie firmowych rozwiązań w celu zachowania możliwości przenoszenia starszych algorytmów sterujących do nowszych wersji systemowych (**kompatybilność wstecz**).



Zgodnie z normą IEC 61131-3 wyróżnia się dwie grupy języków programowania sterowników PLC (rys. 1):

- **języki tekstowe:**
 - **IL** (ang. **I**nstruction **L**ist) – lista instrukcji,
 - **ST** (ang. **S**tructured **T**ext) – tekst strukturalny,
- **języki graficzne:**
 - **LD** (ang. **L**adder **D**iagram) – schemat drabinkowy,
 - **FBD** (ang. **F**unction **B**lock **D**iagram) – schemat bloków funkcyjnych, schemat blokowy,
 - **SFC** (ang. **S**equential **F**unction **C**hart) – schemat sekwencji funkcji, graf sekwencyjny.

Należy podkreślić, że nie wszystkie wymienione powyżej nazwy języków, metody programowania i opisu są aprobowane przez wszystkich producentów sterowników PLC. Przykładowo, firma Siemens zamiast IL używa nazwy **STL** (ang. **S**tatement **L**ist) oraz zamiast LD – **LAD** (ang. **L**adder **D**iagram).

Lista instrukcji IL jest językiem tekstowym najniższego poziomu, typu *assembler*, zależnym w dużym stopniu od typu procesora zastosowanego w sterowniku PLC. Przykładowo, w sterowniku Simatic S7 (firmy Siemens) wykorzystywany jest stos bitów systemowych, natomiast w sterowniku PCDx (firmy SAIA Burgess) sposób wykonywania instrukcji zależy od stanu pojedynczego bitu systemowego, zwanego akumulatorem ACCU.

Tekst strukturalny ST jest językiem tekstowym wyższego poziomu (typu Pascal, C itp.), w którym, oprócz prostych instrukcji assemblerowych języka IL, występują również instrukcje złożone, takie jak pętle iteracyjne (REPEAT-UNTIL, WHILE-DO) wykonanie warunkowe (IF-THEN-ELSE, CASE). Język ten w zdecydowany sposób przyspiesza i ułatwia realizację algorytmów sterujących.




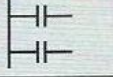

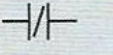

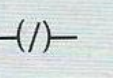
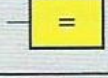
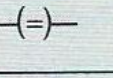

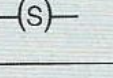

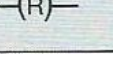
Schemat drabinkowy LD, opracowany został z myślą o projektantach elektrycznych układów przekaźnikowych, realizujących funkcje logiczne poprzez odpowiednie łączenie styków i cewek przekaźników.

Schemat bloków funkcyjnych FBD, opracowany dla projektantów realizujących funkcje logiczne przez odpowiednie łączenie półprzewodnikowych elementów funkcjonalnych (bramki logiczne, kodery, dekodery, multipleksery, komutatory, liczniki itp.) na płytkach drukowanych. Ze względu na to, że język FBD nie dopuszcza kreślenia pętli wokół bloków funkcyjnych, w ostatnich latach opracowany został język **CFC** (ang. **C**ontinuous **F**low **C**hart), który taką możliwość udostępnia.

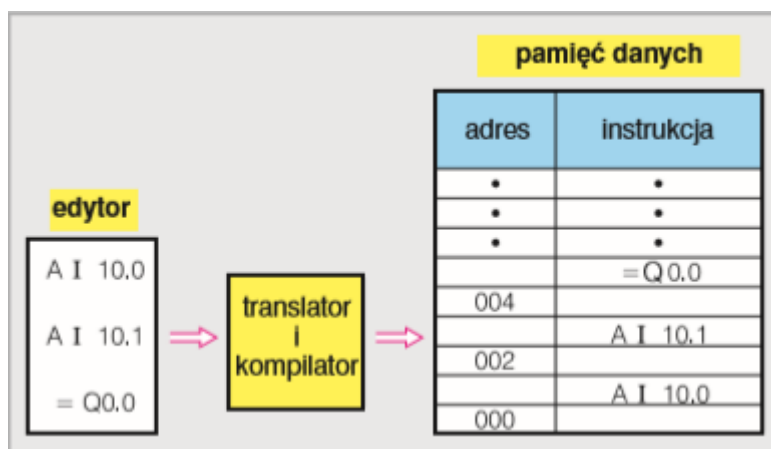
Schemat sekwencji funkcji SFC ułatwia realizację sekwencyjnych układów logicznych, które w odróżnieniu od prostszych układów kombinacyjnych, w których stany wyjść zależą wyłącznie od aktualnego stanu wejść, zawierają elementy pamięci. Sposób sterowania zależy bowiem również od stanu, w jakim znajduje się proces i system automatyki.

Pierwowzorem języka SFC jest **język GRAFCET**, zdefiniowany w 1977 roku i ujęty w **normie IEC 848** w 1987 roku. Obydwa języki, SFC i Grafcet, stanowią narzędzia wykorzystywane do tworzenia struktury algorytmu sterowania, składającej się z sekwencji kroków (ang. *step*) i warunków przejścia pomiędzy krokami (ang. *transition*), określonej logicznymi zależnościami przyczynowo - skutkowymi przebiegu sterowanego procesu (analogia do koncepcji **sieci Pertiego** typu **P/T** – ang. *Position/Transition*). Wszystkie kroki i warunki przejść są podprogramami napisanymi w dowolnie wybranym, jednym z wyżej wymienionych języków programowania.

Do najczęściej stosowanych języków należą: schemat drabinkowy (LD), schemat bloków funkcyjnych (FBD) oraz lista instrukcji (IL) (tab. 1 poniżej).

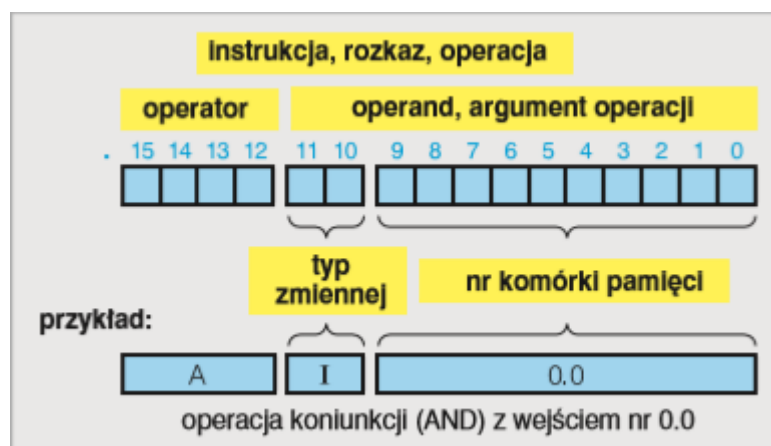
Tab. 1. Przykłady operacji wykonywanych przez sterowniki PLC wg normy IEC 61131-3			
Operacja	Zapis		
	IL	FBD	LD
AND (I)	A		
OR (LUB)	O		
NOT (NIE) wejście	N		
NOT (NIE) wyjście	N		
przypisanie	=		
ustawianie (Set)	S		
zerowanie (Reset)	R		

Stosowane są w nich powszechnie znane, proste instrukcje i symbole, wykorzystywane także w innych technikach sterowania oraz w językach programowania komputerów. Natomiast programy napisane w języku IL, optymalne pod względem czasu przebiegu i zajętości pamięci, są jednocześnie bardzo nieczytelne i trudne w realizacji oraz modyfikowaniu.



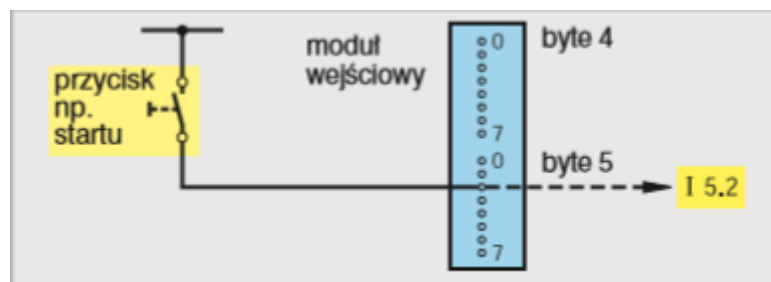
Rys. 2. Wprowadzanie programu użytkownika do sterownika PLC.

Przed wprowadzeniem programu użytkownika do sterownika (rys. 2) wszystkie podprogramy (napisane w językach graficznych) muszą zostać przetłumaczone (przez **translator**) na instrukcje oraz muszą zostać złożone (przez **kompilator**) w jeden moduł wsadowy w postaci **kodu maszynowego**, zrozumiałego dla procesora zastosowanego w sterowniku PLC.



Rys. 3. Format instrukcji (rozkażu, operacji).

Instrukcja (rys. 3) składa się z **operatora** (symbol operacji z ewentualnymi modyfikatorami) oraz **operandu** (argument operacji). Operator określa działanie arytmetyczne, logiczne, transmisyjne, systemowe, organizacyjne itp., wykorzystywane przez instrukcję, natomiast operand stanowi argument tej operacji. Operandem może być **stała** lub **adres**, pod którym zapamiętana jest wartość argumentu. Adres (rys. 4) składa się z typu zmiennej oraz numeru komórki pamięci przechowującej wartość zmiennej – xx.yy (xx – nr byte, yy – nr bitu w byte).



Rys. 4. Przykład operandu – adresu wejścia cyfrowego.