

## Norma IEC 61131-3

Norma *IEC 61131-3* definiuje pojęcia podstawowe, zasady ogólne, model programowy i model komunikacyjny (wymiana danych między elementami oprogramowania) oraz podstawowe typy i struktury danych. Określono w niej dwie grupy języków programowania: języki tekstowe i graficzne.

W grupie *języków tekstowych* zdefiniowane zostały następujące języki:

- *Język IL* (ang. *Instruction List – Lista rozkazów*), będący odpowiednikiem języka typu *assembler*, którego zbiór instrukcji obejmuje operacje logiczne, arytmetyczne, operacje relacji, jak również funkcje przerzutników, czasomierzy, liczników itp.
- *Język ST* (ang. *Structured Text – Tekst strukturalny*), który jest odpowiednikiem języka algorytmicznego *wysokiego poziomu*, zawierającego struktury programowe i polecenia podobne do występujących w językach typu PASCAL lub C.

Do grupy *języków graficznych* opisanych w normie IEC 61131-3 należą:

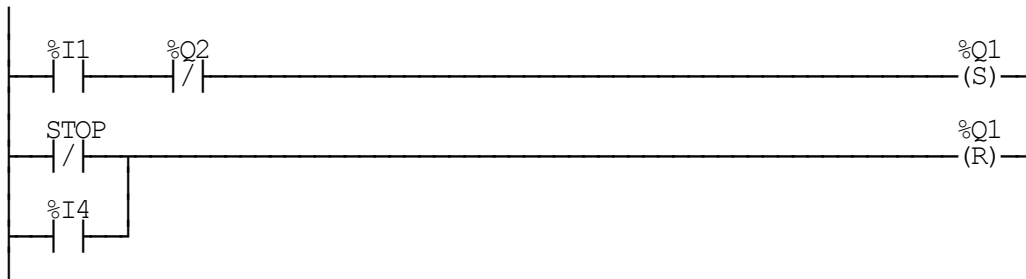
- *Język LD* (ang. *Ladder Diagram – Schemat drabinkowy*), podobny do stykowych obwodów przekaźnikowych, w którym oprócz symboli styków, cewek i połączeń między nimi, dopuszcza się także użycie funkcji (np. arytmetycznych, logicznych, porównań, relacji) oraz bloków funkcjonalnych (np. przerzutniki, czasomierze, liczniki).
- *Język FBD* (ang. *Function Block Diagram – Funkcjonalny schemat blokowy*), będący odpowiednikiem schematu przepływu sygnału dla obwodów logicznych przedstawionych w formie połączonych bramek logicznych oraz funkcji i bloków funkcjonalnych, takich jak w języku LD.

Ponadto przedstawiono sposób tworzenia struktury wewnętrznej programu w postaci *schematu funkcji sekwencyjnej SFC* (ang. *Sequential Function Chart*), który pozwala na opisywanie zadań sterowania sekwencyjnego za pomocą grafów zawierających kroki (etapy) i warunki przejścia (tranzycji) między tymi krokami. W celu otrzymania odpowiedniej struktury programu można wykorzystać SFC, w którym definicje akcji dla poszczególnych kroków oraz warunki przejścia programuje się w jednym z czterech wymienionych wyżej języków.

IEC 61131-3 specyfikuje syntaktykę i semantykę wymienionych języków programowania. *Syntaktyka* (inaczej *składnia*) opisuje elementy języka i sposób ich użycia, natomiast *semantyka* ich znaczenie.

Definiowane są również elementy *konfiguracji*, które wspomagają instalację oprogramowania w sterownikach PLC, oraz możliwości *komunikacyjne* w celu ułatwienia połączenia sterowników z innymi elementami automatycznego systemu sterowania.

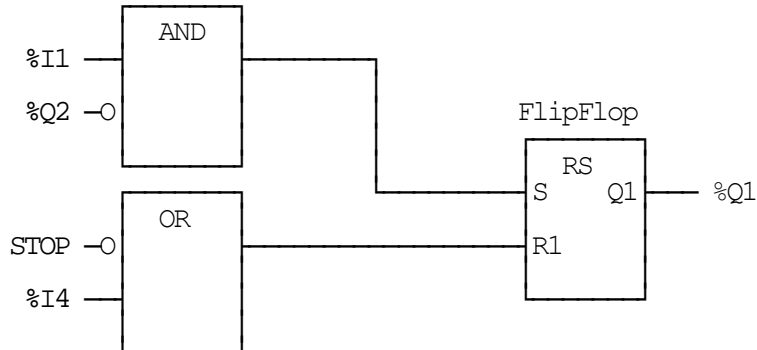
Przykład w języku LD:



Przykład w języku IL:

```
(* pierwszy szczebel drabinki *)
LD    %I1
ANDN  %Q2
S     %Q1
(* drugi szczebel drabinki *)
LDN   STOP
OR    %I4
R     %Q1
```

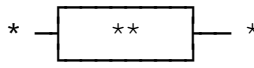
Przykład w języku FBD:



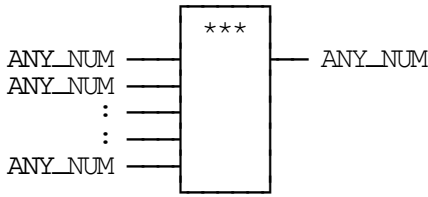
Przykład w języku ST:

```
(* wywołanie przerzutnika *)
FlipFlop( S:= %I1 AND NOT %Q2, R1:=NOT STOP OR %I4 );
(* ustawienie wyjścia *)
%Q1 := FlipFlop.Q1
```

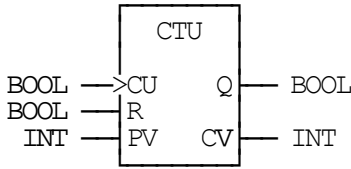
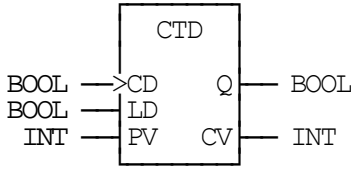
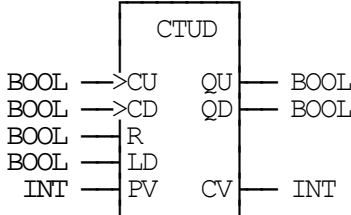
Standardowe funkcje liczbowe jednej zmiennej:

Forma graficzna		Przykłady użycia w językach ST i IL	
 <p>* – oznacza typ wejścia/wyjścia (I/O) ** – oznacza nazwę funkcji</p>		<p><math>A := SIN(B);</math> (* język ST *)</p> <p>LD B (* język IL *) SIN ST A</p>	
Lp.	Nazwa funkcji	Typ I/O	Opis
Funkcje podstawowe			
1	<i>ABS</i>	<i>ANY_NUM</i>	Wartość bezwzględna
2	<i>SQRT</i>	<i>ANY_REAL</i>	Pierwiastek kwadratowy
Funkcje logarytmiczne			
3	<i>LN</i>	<i>ANY_REAL</i>	Logarytm naturalny
4	<i>LOG</i>	<i>ANY_REAL</i>	Logarytm dziesiętny
5	<i>EXP</i>	<i>ANY_REAL</i>	Funkcja wykładnicza o podstawie $e$
Funkcje trygonometryczne			
6	<i>SIN</i>	<i>ANY_REAL</i>	Sinus kąta w radianach
7	<i>COS</i>	<i>ANY_REAL</i>	Cosinus kąta w radianach
8	<i>TAN</i>	<i>ANY_REAL</i>	Tangens kąta w radianach
9	<i>ASIN</i>	<i>ANY_REAL</i>	Arcus sinus
10	<i>ACOS</i>	<i>ANY_REAL</i>	Arcus cosinus
11	<i>ATAN</i>	<i>ANY_REAL</i>	Arcus tangens

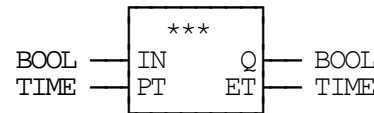
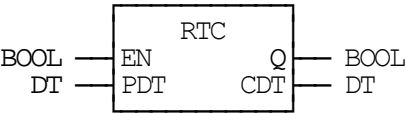
Standardowe funkcje arytmetyczne:

Forma graficzna			Przykład użycia w językach ST i IL	
 <p>*** – oznacza nazwę funkcji lub symbol</p>			<p><math>A := ADD(B, C, D);</math> (* język ST *)  lub  <math>A := B + C + D;</math></p> <p><i>LD B</i> (* język IL *)  <i>ADD C</i>  <i>ADD D</i>  <i>ST A</i></p>	
Lp.	Nazwa	Symbol	Opis	
Funkcje arytmetyczne rozszerzalne (o zmiennej liczbie wejść)				
12	<i>ADD</i>	+	Dodawanie	$OUT := IN_1 + IN_2 + \dots + IN_n$
13	<i>MUL</i>	*	Mnożenie	$OUT := IN_1 * IN_2 * \dots * IN_n$
Funkcje arytmetyczne o stałej liczbie wejść				
14	<i>SUB</i>	-	Odejmowanie	$OUT := IN_1 - IN_2$
15	<i>DIV</i>	/	Dzielenie	$OUT := IN_1 / IN_2$
16	<i>MOD</i>		Reszta z dzielenia	$OUT := IN_1 \text{ modulo } IN_2$
17	<i>EXPT</i>	**	Potęgowanie	$OUT := IN_1^{IN_2}$
18	<i>MOVE</i>	:=	Przepisanie	$OUT := IN$

## Standardowe liczniki

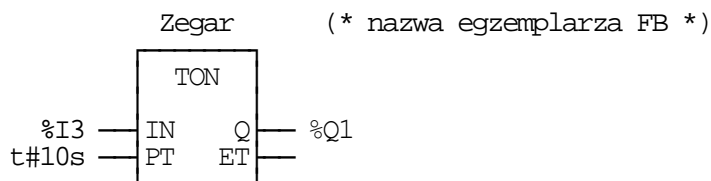
Lp	Forma graficzna	Opis
1		<p>Licznik dodający</p> <p><i>CU</i> – wejście, którego zmiany z 0 na 1 są zliczane</p> <p><i>R</i> – wejście zerujące licznik</p> <p><i>PV</i> – wartość zadana</p> <p><i>Q</i> – wyjście załączane gdy <i>CV</i> osiągnie wartość <i>PV</i></p> <p><i>CV</i> – liczba zliczonych impulsów</p>
2		<p>Licznik odejmujący</p> <p><i>CD</i> – wejście, którego zmiany z 0 na 1 są zliczane</p> <p><i>LD</i> – wejście ustawiające <i>CV</i> na wartość <i>PV</i></p> <p><i>PV</i> – wartość zadana</p> <p><i>Q</i> – wyjście załączane gdy <i>CV</i> osiągnie wartość 0</p> <p><i>CV</i> – liczba zliczonych impulsów</p>
3		<p>Licznik dodająco-odejmujący</p> <p><i>CU</i> – wejście, jego zmiany z 0 na 1 są zliczane w górę</p> <p><i>CD</i> – wejście, jego zmiany z 0 na 1 są zliczane w dół</p> <p><i>R</i> – wejście zerujące licznik</p> <p><i>LD</i> – wejście ustawiające <i>CV</i> na wartość <i>PV</i></p> <p><i>PV</i> – wartość zadana</p> <p><i>QU</i> – wyjście załączane gdy <i>CV</i> osiągnie wartość <i>PV</i></p> <p><i>QD</i> – wyjście załączane gdy <i>CV</i> osiągnie wartość 0</p> <p><i>CV</i> – liczba zliczonych impulsów</p>

## Standardowe czasomierze

№p	Forma graficzna	Opis
1 2 3	 <p>*** oznacza: <i>TP, TON, TOF</i></p>	<i>TP</i> – generator impulsu <i>TON</i> – opóźnione załączenie <i>TOF</i> – opóźnione wyłączenie <i>IN</i> – wejście uruchamiające czasomierz <i>PT</i> – wartość zadana czasu <i>ET</i> – czas mierzony
4		Zegar czasu rzeczywistego <i>PDT</i> – ustalone data i czas (ładowane przy zboczu narastającym na wejściu <i>EN</i> ) <i>CDT</i> – bieżące data i czas (gdy <i>EN = 1</i> ) <i>Q</i> – kopia <i>EN</i>

Przykład użycia czasomierza TON w języku FBD, ST i IL:

(\* język FBD \*)



(\* języki tekstowe \*)

*VAR*

*Zegar : TON;*

(\* deklaracja egzemplarza FB \*)

*END\_VAR*

(\* język ST \*)

*Zegar(IN:=%I3, PT:=t#10s);*

(\* wywołanie \*)

*%Q1:= Zegar.Q;*

(\* nadanie wartości wyjściu %Q1 \*)

(\* język IL \*)

*CAL Zegar(IN:=%I3, PT:=t#10s)*

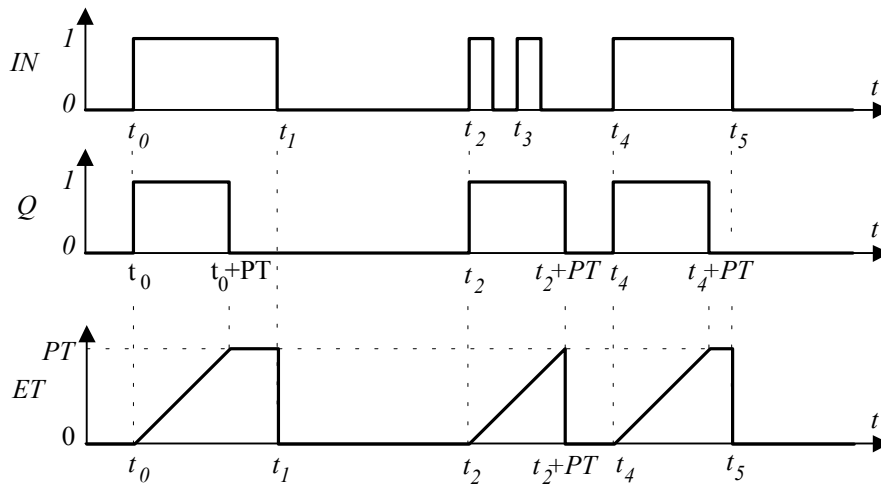
(\* wywołanie \*)

*LD Zegar.Q*

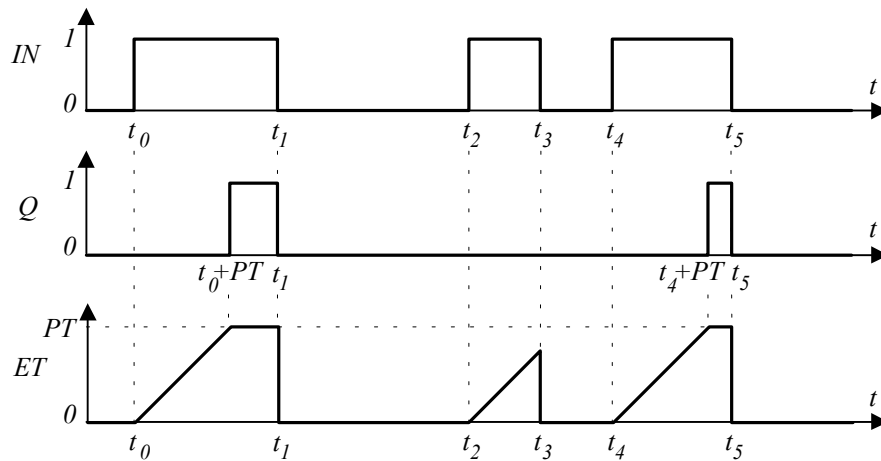
*ST %Q1*

(\* nadanie wartości wyjściu %Q1 \*)

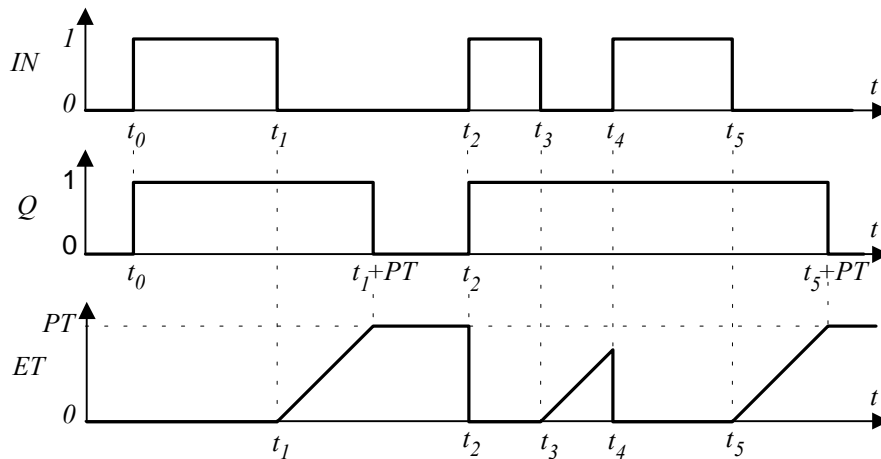
Wykresy czasowe sygnałów dla generatora impulsu TP:



Wykresy czasowe sygnałów dla czasomierza załączającego TON:



Wykresy czasowe sygnałów dla czasomierza wyłączającego TOF:



## Język LD (Schemat drabinkowy)

Symbole styków:

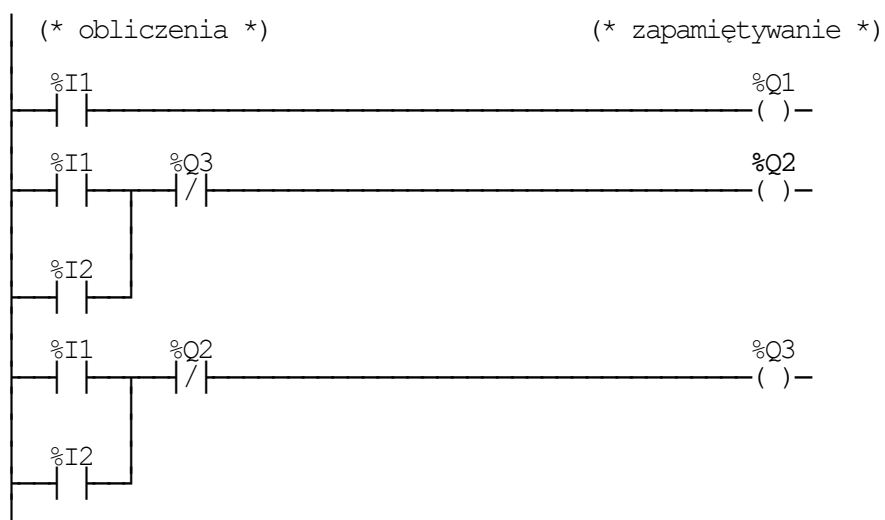
Styki	Symbol	Opis
Styki statyczne	$\begin{array}{c} *** \\ \text{--- ---} \end{array}$	<p><i>Styk zwierny (normalnie otwarty, ang. normally open contact)</i></p> <p>Stan połączenia z lewej strony styku jest przenoszony na prawą stronę, jeżeli skojarzona zmienna boolowska ma wartość <i>1</i>. W przeciwnym razie prawe połączenie jest w stanie <i>OFF</i>.</p>
	$\begin{array}{c} *** \\ \text{--- / ---} \end{array}$	<p><i>Styk rozwierny (normalnie zamknięty, ang. normally closed contact)</i></p> <p>Stan połączenia z lewej strony styku jest przenoszony na prawą stronę, jeżeli skojarzona zmienna boolowska ma wartość <i>0</i>. W przeciwnym razie prawe połączenie jest w stanie <i>OFF</i>.</p>
Styki impulsowe (wrażliwe na zbocze)	$\begin{array}{c} *** \\ \text{--- P ---} \end{array}$	<p><i>Styk wrażliwy na zbocze narastające (ang. Positive transition-sensing contact)</i></p> <p>Połączenie z prawej strony styku jest w stanie <i>ON</i> w czasie jednego wykonania, jeśli połączenie z lewej strony jest w stanie <i>ON</i>, a skojarzona zmienna boolowska zmieniła wartość z <i>0</i> na <i>1</i>. Poza tym stan połączenia z prawej strony jest <i>OFF</i>.</p>
	$\begin{array}{c} *** \\ \text{--- N ---} \end{array}$	<p><i>Styk wrażliwy na zbocze opadające (ang. Negative transition-sensing contact)</i></p> <p>Połączenie z prawej strony styku jest w stanie <i>ON</i> w czasie jednego wykonania, jeśli połączenie z lewej strony jest w stanie <i>ON</i>, a skojarzona zmienna boolowska zmieniła wartość z <i>1</i> na <i>0</i>. Poza tym stan połączenia z prawej strony jest <i>OFF</i>.</p>



Symbole cewek:

Cewki	Symbol	Opis
Cewki zwykłe	*** — ( ) —	<i>Cewka (ang. coil)</i> Stan połączenia z lewej strony cewki jest przenoszony na prawą stronę i zapamiętywany w skojarzonej zmiennej boolowskiej.
	*** — (/) —	<i>Cewka negująca (ang. negated coil)</i> Stan połączenia z lewej strony cewki jest przenoszony na prawą stronę, a jego odwrotność jest zapamiętywana w skojarzonej zmiennej boolowskiej.
Cewki zatrzas kiwane	*** — (S) —	<i>Cewka ustawiająca (ang. Set coil, Latch coil)</i> Skojarzona zmienna przyjmuje wartość 1, jeżeli połączenie z lewej strony jest w stanie ON. Wartość ta pozostanie niezmienną, aż do chwili wyzerowania przez cewkę kasującą ( R ).
	*** — (R) —	<i>Cewka kasująca (ang. Reset coil, Unlatch coil)</i> Skojarzona zmienna przyjmuje wartość 0, jeżeli połączenie z lewej strony jest w stanie ON. Wartość ta pozostanie niezmienną, aż do chwili ustawienia przez cewkę ustawiającą ( S ).
Cewki podtrzymywane, cewki z pamięcią	*** — (M) —	<i>Cewka z zapamiętaniem stanu (ang. Retentive coil, Memory coil)</i>
	*** — (SM) —	<i>Cewka ustawiająca z zapamiętaniem stanu (ang. Set retentive coil)</i>
	*** — (RM) —	<i>Cewka kasująca z zapamiętaniem stanu (ang. Reset retentive coil)</i>
Cewki impulsowe (wrażliwe na zbocze)	*** — (P) —	<i>Cewka wrażliwa na zbocze narastające (ang. Positive transition-sensing coil)</i> Skojarzona zmienna przyjmuje wartość 1 na czas jednego wykonania, jeśli połączenie z lewej strony zmieniło stan z OFF na ON. Stan połączenia z lewej strony jest zawsze przenoszony na prawą.
	*** — (N) —	<i>Cewka wrażliwa na zbocze opadające (ang. Negative transition-sensing coil)</i> Skojarzona zmienna przyjmuje wartość 1 na czas jednego wykonania, jeśli połączenie z lewej strony zmieniło stan z ON na OFF. Stan połączenia z lewej strony jest zawsze przenoszony na prawą.

Przykład przejrzystej struktury obwodów w języku LD:



Przykład nieprzejrzystej struktury obwodów w języku LD:

