

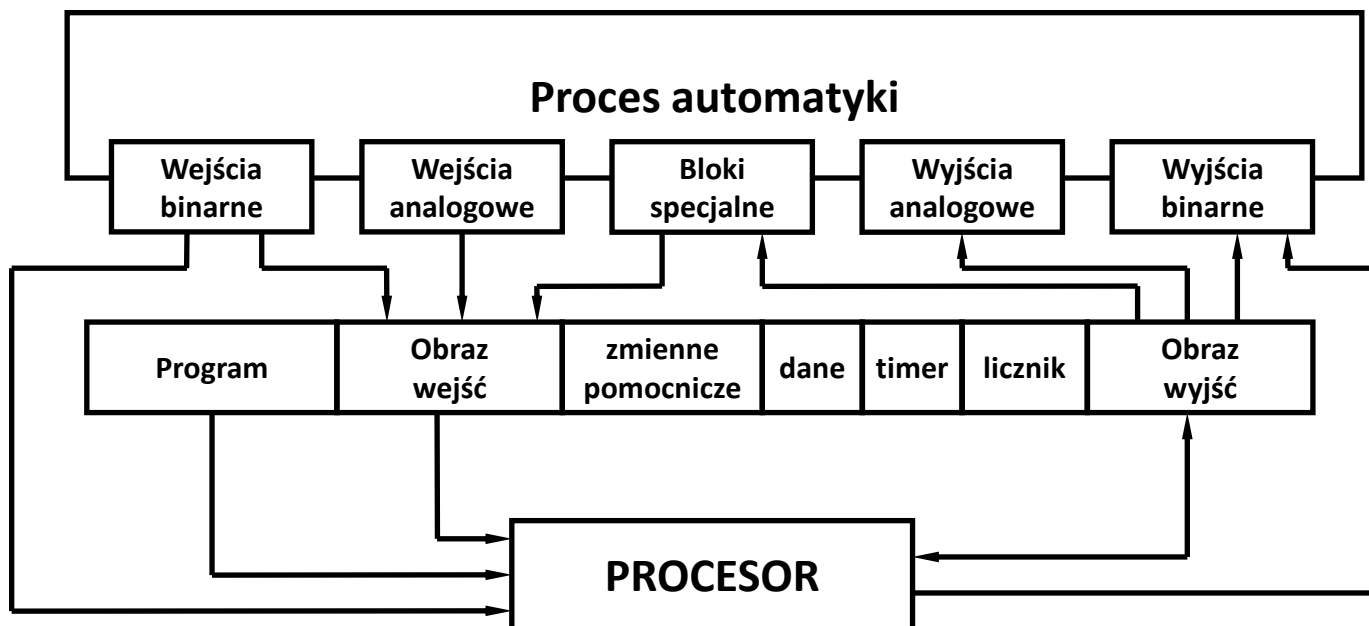


PROJEKT FINANSOWANY  
ZE ŚRODKÓW UNII EUROPEJSKIEJ I BUDŻETU PAŃSTWA

# Sterowniki PLC - Konfiguracja, diagnostyka i programowanie na przykładzie sterowników firmy Siemens



**PLC (Programowalny Sterownik Logiczny)** (ang. *Programmable Logic Controller*) – uniwersalne urządzenie mikroprocesorowe przeznaczone do sterowania pracą maszyny lub urządzenia technologicznego. Sterownik PLC musi zostać dopasowany do określonego obiektu sterowania poprzez wprowadzenie do jego pamięci żądanego algorytmu działania obiektu. Cechą charakterystyczną sterowników PLC odróżniającą ten sterownik od innych sterowników komputerowych jest cykliczny obieg pamięci programu. Algorytm jest zapisywany w dedykowanym sterownikowi języku programowania.



Rys.1 Struktura sterownika.

# Sterowniki programowalne - podział

Podział sterowników jest umowny, jednak znajduje on odbicie w ofercie producentów sterowników PLC.

W odniesieniu do sprzętu rodziny S7 podział obejmuje

- **S7-200** - sterowniki małe typu kompakt
- **S7-300** - sterowniki średnie
- **S7-400** - sterowniki średnie
- **S7-1200** - sterowniki duże

Wraz z rodziną S7 pojawiły się też systemy **SIMATIC M7** i **SIMATIC C7**:

**SIMATIC M7**-jest to system automatyzacji na bazie komputera PC AT dostarczający użytkownikowi możliwość programowania urządzenia sterującego w typowym dla komputera środowisku.

**SIMATIC C7**- stanowi połączenie jednostki centralnej S7-300 z panelem operatorskim, modułami we/wy oraz modułem interfejsowym

System sterowania bazujący na sterownikach programowalnych obejmuje również inne elementy stanowiące zwykle integralną jego część:

- Oprogramowanie systemowe i wspomagające
- Struktury hierarchiczne- połączenie między sterownikami /systemami
- Urządzenie peryferyjne
- Panele operatorskie/urządzenie wizualizacji

# S7-300 Jednostki centralne CPU

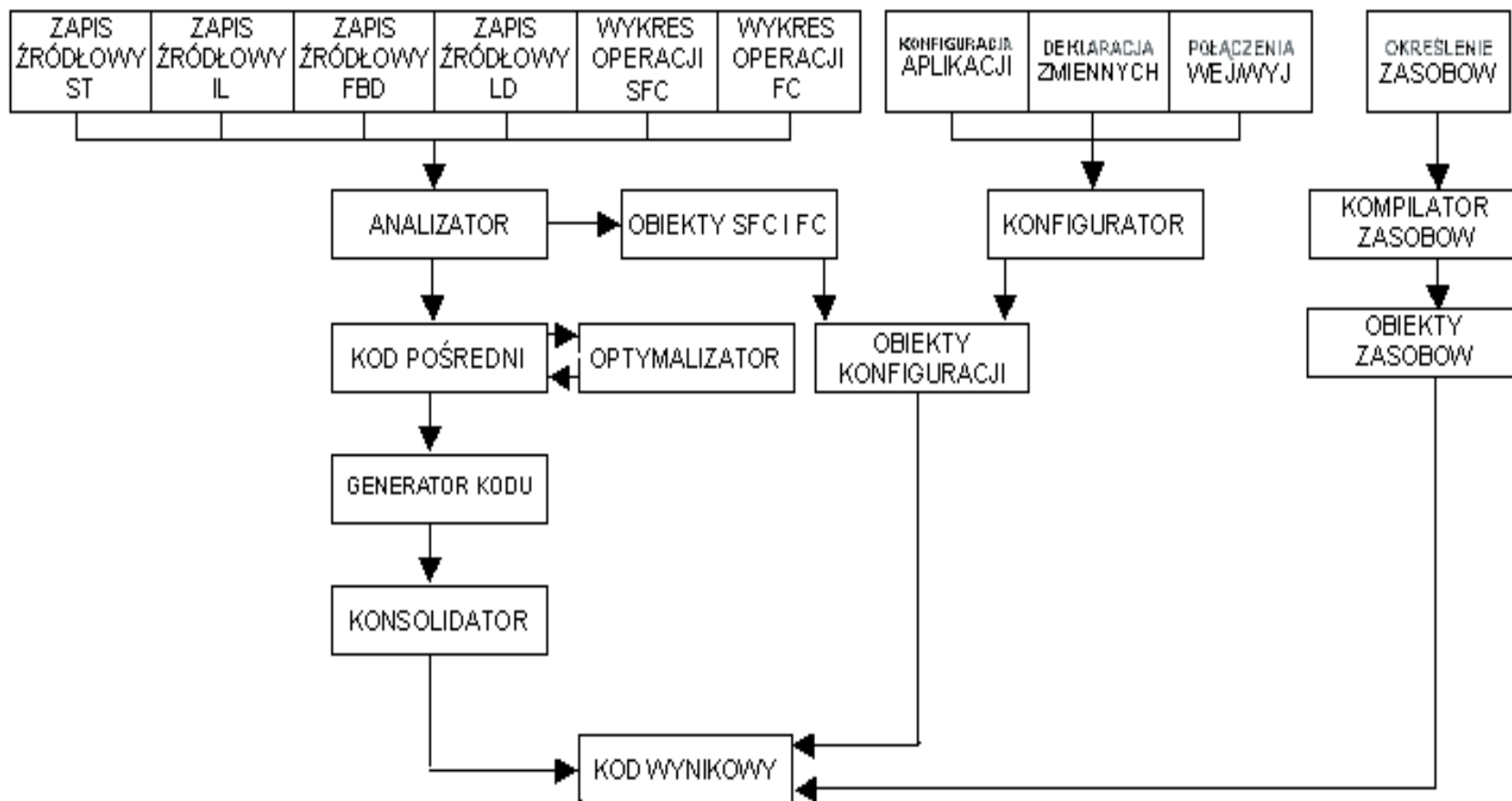
## Cechy sterownika :

- Możliwość konfigurowania CPU ( różna klasa obliczeniowa) oraz doboru modułów zewnętrznych do aplikacji ( również moduły komunikacyjne CP)
- Możliwości komunikacyjne MPI, Profibus, AS-I oraz Ethernet
- Oprogramowanie STE 7/STEP 7 Mini
- Konfiguracja programowa *Hardware Configuration*
- Montaż na szynie DIN
- Obsługa tzw. błędów synchronicznych i asynchronicznych,
- Programowanie i diagnostyka danych lokalnych
- Adresacja stała i zmienna modułów wejścia/wyjścia

## Możliwości funkcjonalne:

- Sterowanie wejść/wyjść
- Zegar czasu rzeczywistego
- Zintegrowane wyjścia analogowe
- Programowanie w blokach OB./FC/FB/DB
- Tworzenie konfiguracji wielopanelowej

# S7-300 Cykl pracy sterownika (CPU)



Rys.4 Generowanie programu i cykl pracy sterownika.

# Elementy systemu sterownika S7-300

Do rodziny sterowników S7-300 oprócz CPU zaliczamy różnego rodzaju moduły rozszerzające :

**Moduł sygnałowy SM** - grupuje sygnały zewnętrzne i dopasowuje je do różnych poziomów sygnałów wewnętrznych sterownika

**Moduł interfejsowy IM** – umożliwia konfigurację wielopanelową. Szyna jest przedłużona do kolejnych rzędów:

- IM 360 instalowany jest w CR ( Central Rack) szynie centralnej
- IM 361 instalowany jest w ER ( Extension Rack) szynie rozszerzającej
- IM 365 jako korzystne cenowo rozwiązanie dla budowy dwurzędowej, w drugim rzędzie tylko moduły sygnałowe

**Moduł funkcyjny FM** – umożliwia realizację „funkcji specjalnych” ( wymagających dużej mocy obliczeniowej lub krótkich czasów reakcji, odciążających CPU):

- Zliczanie szybkich impulsów (np. FM 350)
- Pozycjonowanie (np.. FM 351/353/354)
- Regulacja (np.. FM 355)

**Moduł komunikacyjny CP**- udostępnia następujące możliwości przyłączenia do sieci:

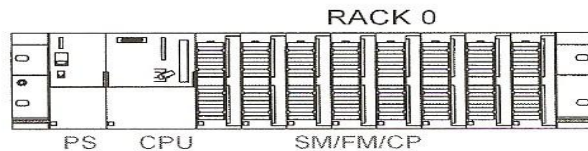
- Połączenie między dwoma urządzeniami (np. CP 340/341)
- AS-Interface (np. CP 342-2/342-3)
- Industrial Ethernet (np. CP 343-1)

# Zasady montażu – S7-300

- Zasilacz i jednostkę centralną CPU należy zainstalować po lewej stronie przy budowie poziomej lub u dołu przy budowie pionowej.
- Należy zachować odległości minimalne:
  - 20 mm na prawo i lewo od sterownika
  - 40 mm u dołu i na górze i min 60 mm między sterownikami
- Moduł rozszerzający IM należy umieścić obok jednostki CPU.
- Maksymalnie liczba modułów w jednym rzędzie wynosi 8.
- Konfiguracja wielorzędowa (wielopanelowa) jest możliwa dla jednostek CPU 314/315/316/318.
- Zapewnić niskoomowe połączenie pomiędzy szyną montażową a podłożem.
- Liczbą modułów (SM/FM/CP) podłączonych na panelu (rack'u) centralnym ( konfiguracja 1-panelowa) jest ograniczona max. poborem prądu przez moduły (0,8A dla CPU312IFM, 1,2A dla pozostałych)
- Moduły IM361/365 są instalowane na słocie 3 rack'a (ów) rozszerzenia
- Jednostka centralna CPU jest montowana zawsze na rack'u 0 (centralnym)

# Zasady montażu – S7-300

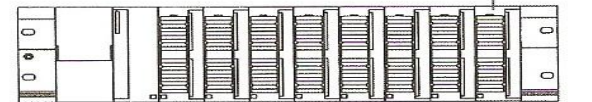
- Max. liczba modułów dołączonych do rack'a rozszerzenia wynosi 8 (SM/FM/CP) i są one instalowane z prawej strony modułu IM ( dla CPU314IFM slot 11 na rack'u 3 musi pozostać wolny)
- Moduły IM365 mogą być wykorzystane do połączenia w konfiguracji 2-panelowej; całkowity pobór prądu nie może przekroczyć wartości 1,2A (rack 1 (rozszerzenia) 0,8A)



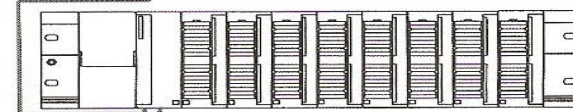
Konfiguracja 1-panelowa

Konfiguracja 4-panelowa

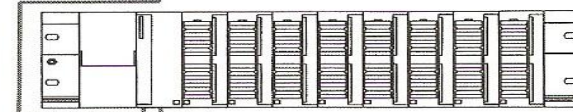
RACK 3



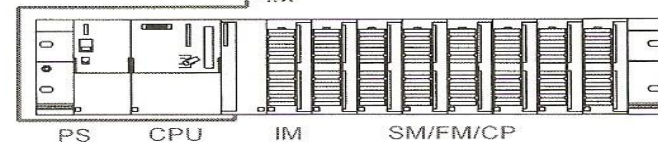
RACK 2



RACK 1



RACK 0

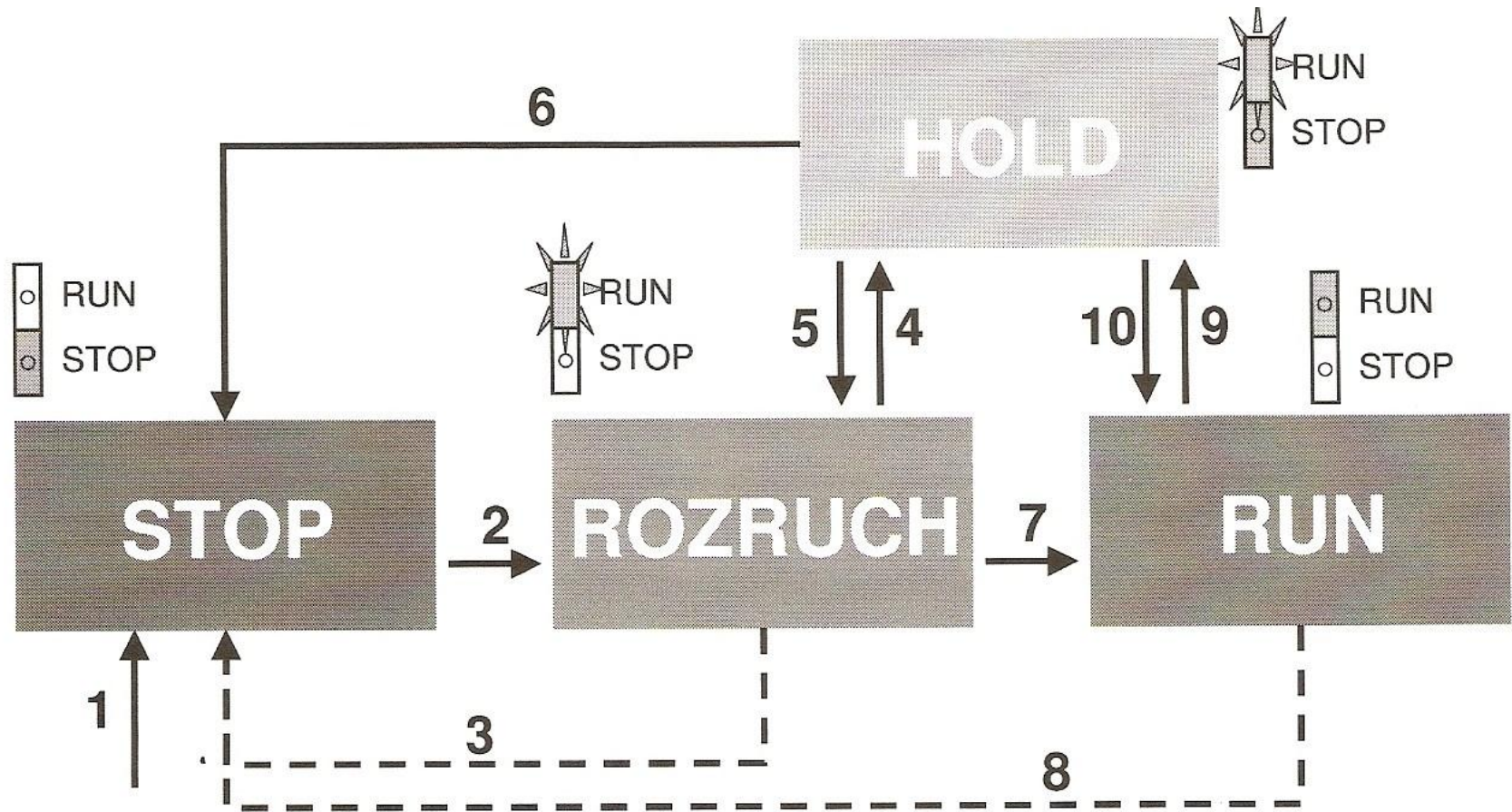




# Kryteria doboru sterownika

- Rodzaj obiektu:
  - pojedyncza maszyna
  - linia technologiczna
- Wielkość obiektu określana liczbą punktów/kanałów:
  - liczba wejść/wyjść cyfrowych, napięcia operacyjne, obciążalność prądowa
  - liczba wejść/wyjść analogowych, wielkość pomiarowa, zakres
  - liczba modułów określona na podstawie liczby punktów/kanałów obsługiwanych przez sterownik
  - liczba sterowników, ich konfiguracja
- Główne zadania stawiane sterownikowi:
  - zbieranie danych o obiekcie, monitorowanie tych danych, wizualizacja
  - sterowanie w układzie otwartym, sterowanie w układzie zamkniętym
  - obliczenia statystyczne, raportowanie
- Warunki eksploatacyjne:
  - temperatura, wilgotność, wibracje, środowiska wybuchowe
- Kryteria ekonomiczne:
  - cena sterownika i osprzętu, możliwości rozbudowy systemu,
  - koszty projektu części elektrycznej i automatyki, koszty montażu i rozruchu
  - czas realizacji

# Tryb pracy CPU i przejścia między nimi

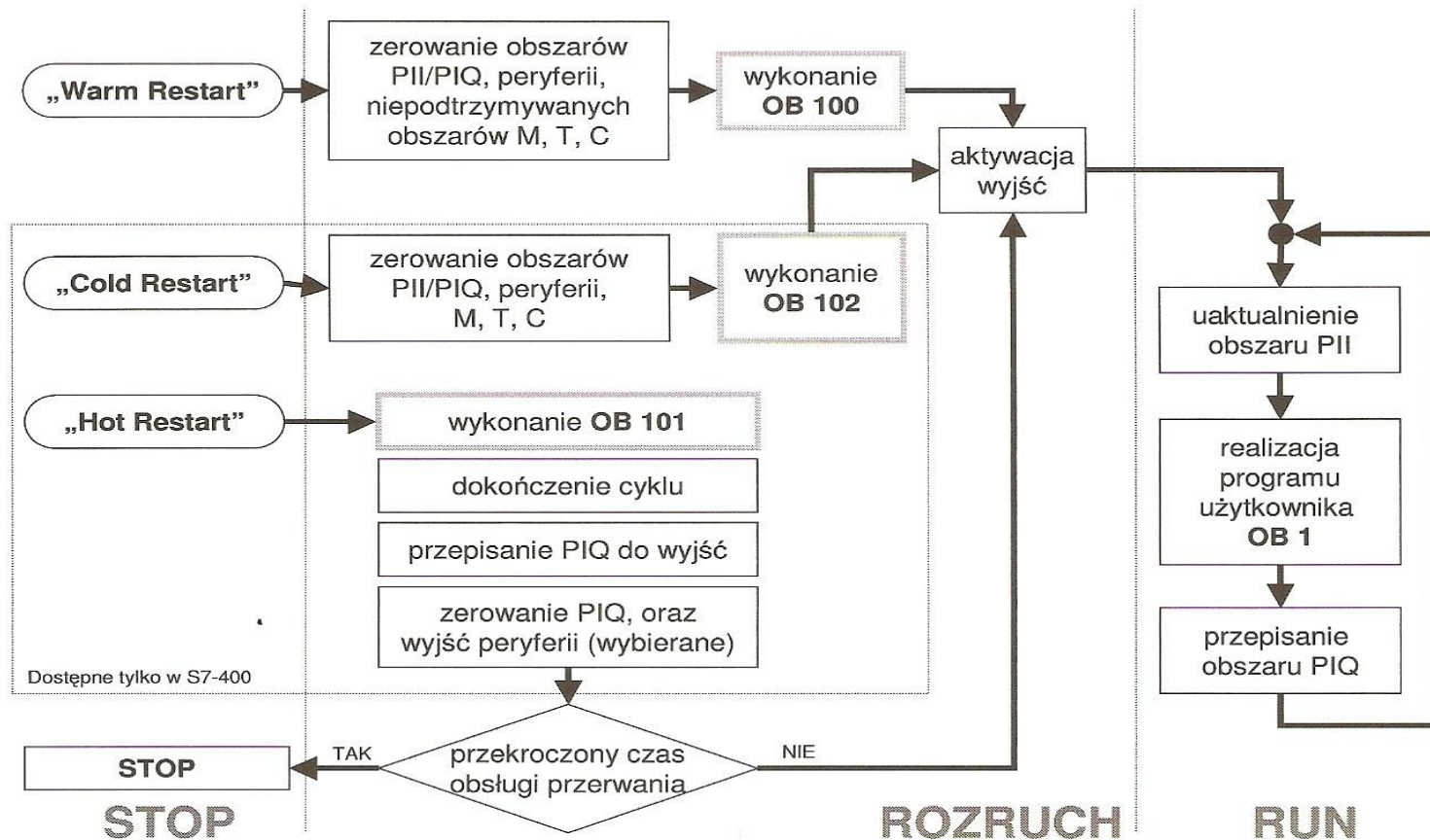


# Rozruch – rodzaje oraz wykonywane przez CPU funkcje

W jednostkach centralnych serii S7-300 wyróżniane są trzy typy rozruchu:

- „**Warm restart**” – po wykonaniu, którego program realizowany jest od początku. Obszary pamięci, które nie zostały zadeklarowane jako podtrzymywane, zostają wyzerowane. Jeżeli użytkownik załadował do sterownika blok OB 100 to zadanie zapisane w nim zostaną wykonane.
- „**Hot restart**” – pozwala na rozpoczęcie wykonywania programu od momentu w którym jego realizacja została przerwana, np. w wyniku zaniku napięcia zasilającego. Obszary pamięci pomocniczej, układy czasowe i liczniki nie zmieniają swojej zawartości. „Hot restart” jest dostępny tylko w S7-300 serii 318. Jeżeli użytkownik załadował do sterownika OB 101 to zadania zapisane w nim zostaną wykonane przed dokończeniem pozostałej części cyklu.
- „**Cold restart**” – program jest wykonywany od początku. Zerowane są wszystkie obszary pamięci, również obszary zadeklarowane jako podtrzymane. Warunkiem parametryzacji jest wykorzystanie oprogramowania STEP 7 w wersji 5 lub wyższej. Jeżeli użytkownik załadował do sterownika blok OB 102, to zadania w nim zapisane zostaną wykonane przed rozpoczęciem realizacji zadań zapisanych w bloku OB1.

# Rozruch – rodzaje oraz wykonywane przez CPU funkcje



# S7-300 Hardware Configuration

HW Config - [SIMATIC 300(L) (Configuration) -- Voss\_Butyl\_6]

Station Edit Insert PLC View Options Window Help

(U) UK

1	
2	CPU 313C-2 DP
X2	DP
2.2	DI16/DO16
2.4	Zähler
3	
4	CP 343-1
5	
6	
7	
8	
9	
10	
11	

Panel: DP master system (1)

(3) Festo CP

(0) UF

Slot	Module	Order number	Fi...	vl...	I...	Q...	Comment
2	CPU 313C-2 DP	6ES7 313-6CE01-0AB0	V2.0	2			
X2	DP				1223		
2.2	DI16/DO16				0...1	0...1	
2.4	Zähler				762...	762...	
3							
4	CP 343-1	6GK7 343-1EX13-0XE0	V2.0	3	6...21	6...21	
5							
6							
7							
8							
9							
10							

PROFIBUS-DP slaves for SIMATIC (distributed rack)

# Parametryzacja CPU – zakładka General

Properties - CPU 313C-2 DP - (R0/S2)

Time-of-Day Interrupts | Cyclic Interrupts | Diagnostics/Clock | Protection | Communication  
General | Startup | Cycle/Clock Memory | Retentive Memory | Interrupts

Short Description: CPU 313C-2 DP

Work memory 32KB; 0.1ms/1000 instructions; DI16/DO16 integrated; 3 pulse outputs (2.5kHz); 3 channel counting and measuring with incremental encoders 24V (30kHz); MPI+ DP connector (DP master or DP slave); multi-tier configuration up to 31 modules; Send and receive

Order No./firmware 6ES7 313-6CE01-0AB0 / V2.0

Name: CPU 313C-2 DP

Interface

Type: MPI

Address: 2

Networked: Yes

Properties...

Comment:

OK Cancel Help

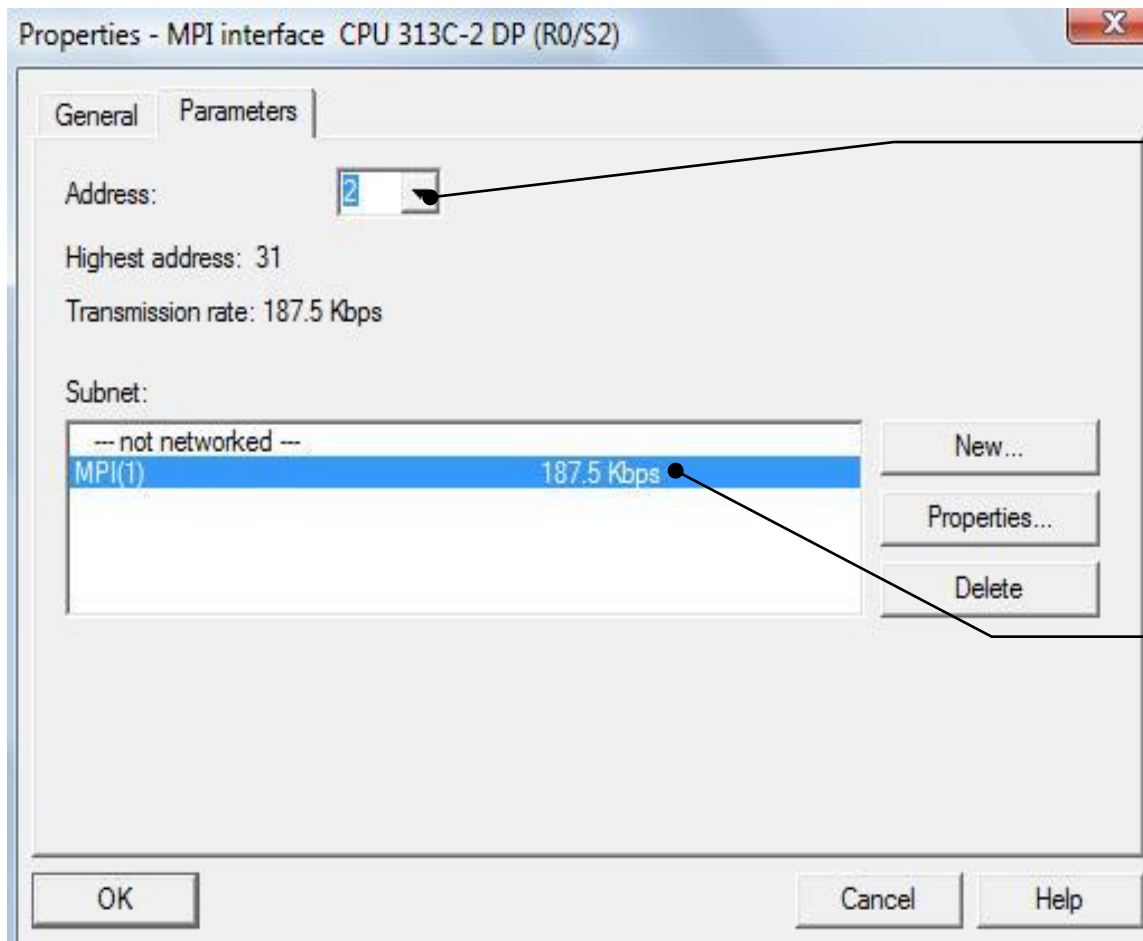
Ogólne informacje o module

Nazwa modułu wykorzystanego w projekcie

Parametry wbudowanego interfejsu MPI

Komentarz tekstowy związany z modułem

# Parametryzacja CPU – parametry interfejsu MPI



Adres interfejsu MPI

Sieci MPI, do których można dołączyć CPU

# Parametryzacja CPU – zakładka Startup

Na zakładce tej możliwe jest zdeklarowanie:

- Zachowania się CPU w momencie wykrycia niezgodności pomiędzy zadaną w projekcie konfiguracją a konfiguracją rzeczywistą: **Startup at Preset Configuration Not Equal to Actual Configuration**
- Trybu rozruchu wykonywanego po włączeniu zasilania jeżeli przełącznik trybu pracy znajduje się w pozycji RUN lub RUN-P ( **Startup after Power On** )
- Kasowania wyjść po wykonaniu „Hot restart” ( opcja dostępna dla S7-400) – **Reset Outputs at Hot Restart**
- Zablokowanie wyzwolenia „Hot Restart” z programatora czy te przez inną stację poprzez sieć MPI (**Disable hot restart at startup by means...**  )
- Maksymalnego czasu oczekiwania na zgłoszenie gotowości do pracy przez moduły po włączeniu zasilania : „**Finished” Message by Means of Modules**, domyślne ustawienie – 650\*100ms czyli 65 sekund
- Monitorowanie czasu przesyłania parametrów do wszystkich modułów w konfiguracji: **Transfer of Parameters to Modules**, domyślne ustawienie – 100 (=10s)



# Parametryzacja CPU – zakładka Startup

Properties - CPU 313C-2 DP - (R0/S2)

Time-of-Day Interrupts | Cyclic Interrupts | Diagnostics/Clock | Protection | Communication  
General | **Startup** | Cycle/Clock Memory | Retentive Memory | Interrupts

Startup if preset configuration does not match actual configuration

Reset outputs at hot restart

Disable hot restart by operator (for example, from PG)  
or communication job (for example, from MPI stations).

Startup after Power On

Hot restart     Warm restart     Cold restart

Monitoring Time for

"Finished" message from modules [100 ms]:

Transfer of parameters to modules [100 ms]:

Hot restart [100 ms]:

OK    Cancel    Help

# Parametryzacja CPU – zakładka Cycle / Clock Memory

Zakładka ta pozwala na określenie:

- Maksymalnego „obciążenia” cyklu programowego przez zadania związane z komunikacją poprzez interfejs MPI: Cycle Load from Communication [%], domyślne ustawienie – 20. Ustawienie zbyt wysokiej wartości może spowodować „spowolnienie” jednostki centralnej (zwiększenie czasu obiegu pętli programowej). Możliwe jest ustawienie wartości z zakresu 10 do 50%.
- Minimalnego czasu obiegu pętli programowej: Scan Cycle Monitoring Time, domyślne ustawienie – 150ms.

Zaznaczając okienko Clock Memory możliwe jest skorzystanie z kilku generatorów o określonych częstotliwościach. Po wybraniu tej opcji należy podać numer bajtu w obszarze pamięci dodatkowej ( obszar M), w którym będą te częstotliwości dostępne.

# Parametryzacja CPU – zakładka Cycle / Clock Memory

Properties - CPU 313C-2 DP - (R0/S2)

Time-of-Day Interrupts | Cyclic Interrupts | Diagnostics/Clock | Protection | Communication  
General | Startup | **Cycle/Clock Memory** | Retentive Memory | Interrupts

Cycle

Update OB1 process image cyclically

Scan cycle monitoring time [ms]:

Minimum scan cycle time [ms]:

Scan cycle load from communication [%]:

Size of the process image:

OB85 - call up at I/O access error:

Clock Memory

Clock memory

Memory Byte:

OK Cancel Help

# Parametryzacja CPU – zakładka Retentive Memory

Properties - CPU 313C-2 DP - (R0/S2)

Time-of-Day Interrupts | Cyclic Interrupts | Diagnostics/Clock | Protection | Communication

General | Startup | Cycle/Clock Memory | Retentive Memory | Interrupts

Retentivity

Number of memory bytes starting with MB0:

Number of S7 timers starting with T0:

Number of S7 counters starting with C0:

Areas

	DB No.	Byte Address	Number of Bytes
Retentive Area 1:	<input type="text" value="1"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
Retentive Area 2:	<input type="text" value="1"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
Retentive Area 3:	<input type="text" value="1"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
Retentive Area 4:	<input type="text" value="1"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
Retentive Area 5:	<input type="text" value="1"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
Retentive Area 6:	<input type="text" value="1"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
Retentive Area 7:	<input type="text" value="1"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
Retentive Area 8:	<input type="text" value="1"/>	<input type="text" value="0"/>	<input type="text" value="0"/>

OK Cancel Help

Ilość bajtów obszaru M (począwszy od MB0), oraz numery układów czasowych i liczników (począwszy od T0/C0), których stan powinien być podtrzymywany

Podtrzymywane obszary bloków danych - DB

# Parametryzacja CPU – zakładka Interrupts

Zakładka ta pozwala na określenie priorytetów dla przerw ( bloków OB.), które mogą być obsługiwane przez daną jednostkę centralną. Dotyczy ona przerw sprzętowych (grupa OB 40), opóźnionych w czasie ( grupa OB 20), komunikacyjnych ( grupa OB 50), oraz obsługi błędów ( grupa OB 70/80).

Dzięki priorytetom możliwe jest określenie zachowania CPU w momencie jednoczesnego wystąpienia kilku przerw.

Priorytety zawierają się w granicach 1 ( najniższy - przypisany na stałe do OB 1) do 26 (29) ( priorytet najwyższy).

Jeżeli jednocześnie wystąpi kilka przerw to CPU najpierw zareaguje na przerwanie o wyższym priorytecie, później o niższym. Jeżeli w trakcie obsługi jednego zdarzenia wystąpi inne to jego klasa priorytetowa będzie decydowała o tym, czy zostanie ono obsłużone natychmiast, czy też zostanie dokończona obsługa przerwania bieżącego.

# Parametryzacja CPU – zakładka Interrupts

Properties - CPU 313C-2 DP - (R0/S2)

Time-of-Day Interrupts | Cyclic Interrupts | Diagnostics/Clock | Protection | Communication  
General | Startup | Cycle/Clock Memory | Retentive Memory | Interrupts

Hardware Interrupts		Time-Delay Interrupts		Async. Error Interrupts	
	Priority:		Process image partition:		Priority:
OB40:	16	OB20:	3	OB81:	26
OB41:	17	OB21:	4	OB82:	26
OB42:	18	OB22:	5	OB83:	26
OB43:	19	OB23:	6	OB84:	26
OB44:	20			OB85:	26
OB45:	21			OB86:	26
OB46:	22			OB87:	26
OB47:	23			OB70:	25
				OB72:	28
				OB73:	0

Interrupts for DPV1

	Priority:
OB55:	2
OB56:	2
OB57:	2

OK Cancel Help

# Parametryzacja CPU – Time of Day Interrupts

Na zakładce tej możliwe jest ustawienie następujących parametrów dla przerw czasu rzeczywistego ( grupa OB 10) tzn. generowanych w momencie wystąpienia zgodności czasu systemowego z zadany kryterium (czas początkowy oraz powtarzalność). Do parametrów tych przerw należą:

- Uaktywnienie bloku przerwa czasowych
- Częstotliwość wywołań, możliwe jest określenie częstotliwości z jaką ma być wywoływane przerwanie po wystąpieniu warunku wyzwolenia. Dostępne są opcje: jednokrotne ( Once), co minutę( Every), co godzinę (Every Hour),.....
- Daty wyzwolenia, parametr ten określa od jakiej daty ma być generowane przerwanie, z częstotliwością określoną w Execution
- Godziny do której przerwanie ma być generowane, przerwanie czasowe będzie generowane od momentu spełnienia warunku określonego przez pola Start Date oraz Time do godziny określonej w polu Time

# Parametryzacja CPU – Time of Day Interrupts

Properties - CPU 313C-2 DP - (R0/S2)

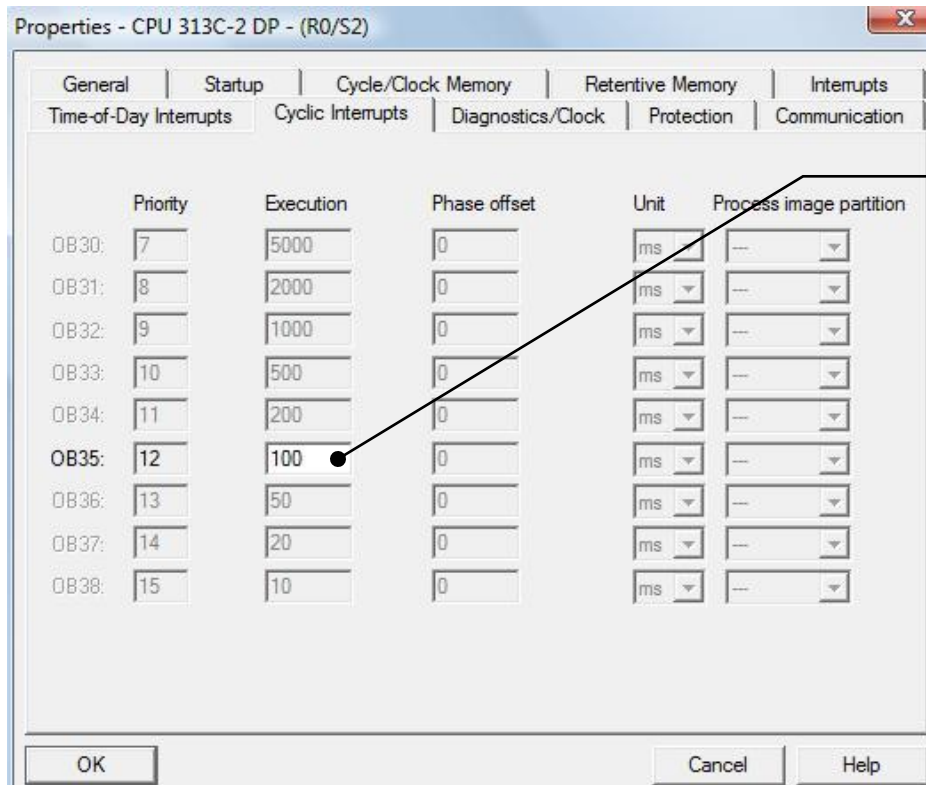
General | Startup | Cycle/Clock Memory | Retentive Memory | Interrupts  
Time-of-Day Interrupts | Cyclic Interrupts | Diagnostics/Clock | Protection | Communication

	Priority	Active	Execution	Start date	Time of day	PIP
OB10:	2	<input type="checkbox"/>	None	01/01/1994	00:00	--
OB11:	2	<input type="checkbox"/>	None	01/01/1994	00:00	--
OB12:	2	<input type="checkbox"/>	None	01/01/1994	00:00	--
OB13:	2	<input type="checkbox"/>	None	01/01/1994	00:00	--
OB14:	2	<input type="checkbox"/>	None	01/01/1994	00:00	--
OB15:	2	<input type="checkbox"/>	None	01/01/1994	00:00	--
OB16:	2	<input type="checkbox"/>	None	01/01/1994	00:00	--
OB17:	2	<input type="checkbox"/>	None	01/01/1994	00:00	--

OK Cancel Help



# Parametryzacja CPU – Cyclic Interrupt



Interwał z jakim wywoływany będzie OB 35 (blok obsługi przerwania cyklicznego)

W danej zakładce deklarowane są parametry przerwania cyklicznych ( grupa OB 30 tzn. zdarzeń generowanych wewnątrz CPU z określonym interwałem czasowym.

# Parametryzacja CPU – Diagnostics/ Clock

Properties - CPU 313C-2 DP - (R0/S2)

General | Startup | Cycle/Clock Memory | Retentive Memory | Interrupts  
Time-of-Day Interrupts | Cyclic Interrupts | Diagnostics/Clock | Protection | Communication

System Diagnostics

- Extended functions
- Report cause of STOP
- Acknowledgment-triggered reporting of SFB33-35

Clock

Synchronization	Synchronization Type	Time Interval
In the PLC:	None	None
On MPI:	None	None
On MFI:	None	None

Correction factor: 0 ms

OK Cancel Help

Przekazywanie informacji o przyczynie przejścia w stan STOP poprzez MPI do PG lub OP

System synchronizacji zegara systemowego

# Parametryzacja CPU – Protection

Zakładka ta pozwala na definicję poziomu zabezpieczenia dostępu do CPU ( pole **Level of Protection**). Możliwe poziomy to:

- Poziom 1 – zabezpieczenie przy pomocy klucza – dla stacji z wyjmowanym kluczykiem,
- Poziom 2 – zabezpieczenie przed zapisem przy pomocy hasła
- Poziom 3 – zabezpieczenie przed zapisem i odczytem przy pomocy hasła

Kolejnym parametrem definiowanym na tej zakładce jest tryb pracy CPU ( pole **Mode**). CPU może pracować w trybie:

- Process Mode – w tym wypadku obsługa komunikacji z programatorem jest ograniczona tak aby czas obiegu pętli programowej nie został wydłużony o więcej niż zostało to określone.
- Test Mode – w tym trybie obsługa komunikacji z programatorem nie jest ograniczona co może doprowadzić do znacznego wydłużenia czasu obiegu pętli programowej i w skrajnym wypadku do zatrzymania CPU ze względu na przekroczenie maksymalnego zadanego czasu obiegu pętli programowej.

# Parametryzacja CPU – Protection

Properties - CPU 313C-2 DP - (R0/S2)

General | Startup | Cycle/Clock Memory | Retentive Memory | Interrupts  
Time-of-Day Interrupts | Cyclic Interrupts | Diagnostics/Clock | Protection | Communication

Protection level

- 1: No protection
  - Can be bypassed with password
- 2: Write-protection
- 3: Write-/read protection

Password:

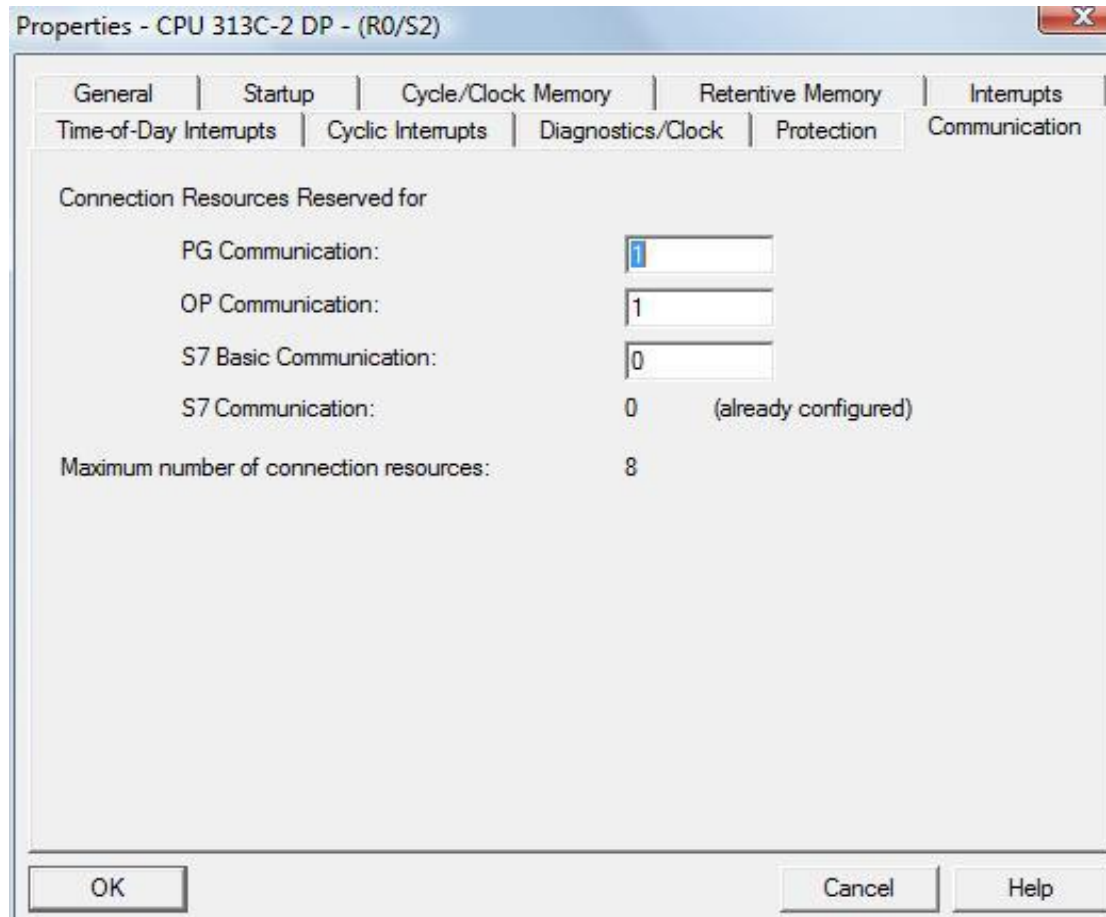
Enter again:

Mode

- Process mode  
Permissible cycle increase via test functions:  ms
- Test mode

OK Cancel Help

# Parametryzacja CPU – zakładka Communication

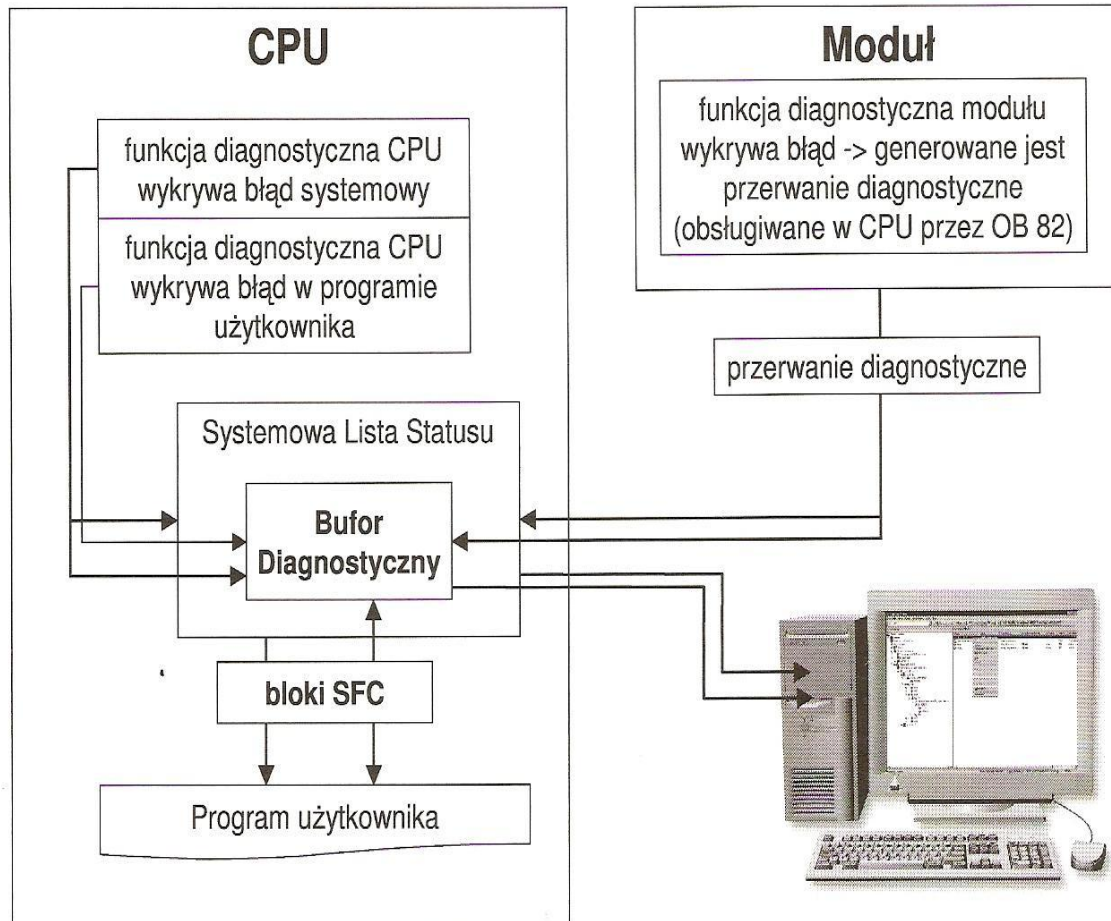


Zakładka Communication pozwala na rezerwację zasobów komunikacyjnych CPU

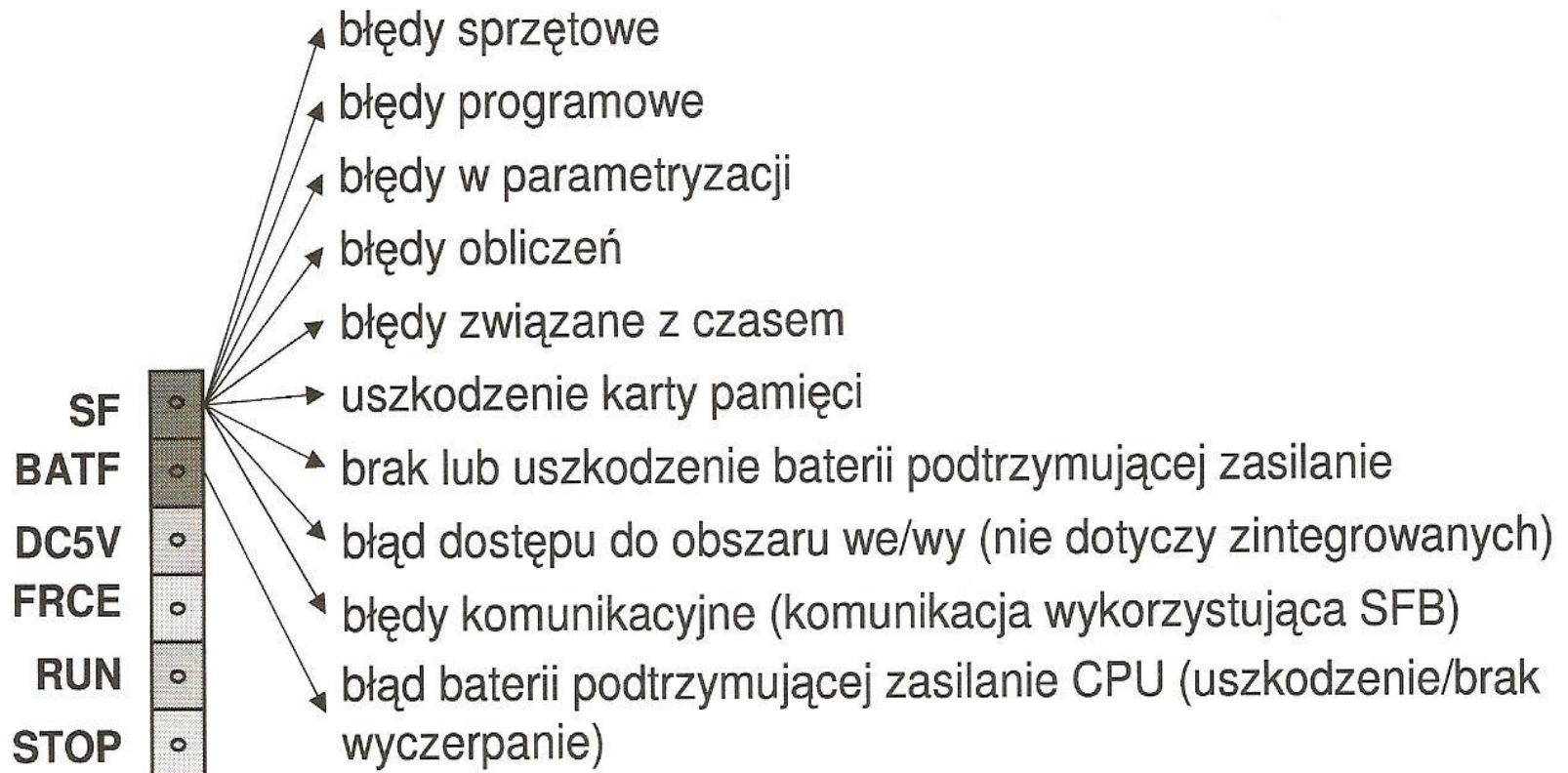
# Diagnostyka S7



# Przeptyw informacji diagnostycznych



# Sygnalizacja błędów na panelu S7-300





# Reakcja CPU na błędy w programie sterowania

*Tryb pracy CPU*

**STOP**

- Analiza LED na panelu CPU
- Analiza zawartości bufora diagnostycznego
- Analiza stosu bloków (BSTACK)
- Analiza stosu przerwań (ISTACK)
- Analiza programu sterowania

**RUN**

- Analiza LED na panelu CPU
- Analiza zawartości bufora diagnostycznego
- Analiza bloków OB reakcji na błędy (8x oraz 12x)
- Analiza programu sterowania

# Informacja o stanie modułów - HWC

HW Config - [SIMATIC 300(1) (Configuration) -- biurko]

Station Edit Insert PLC View Options Window Help
















(0) UR

1	
2	<b>CPU 313C-2 DP</b>
X2	DP
2.2	D16/D016
2.4	Count
3	
4	
5	
6	
7	
8	
9	
10	
11	

(0) UR

Slot	Module	Order number	Firmware	MPI ad...	I address	Q address	Comment
1							
2	<b>CPU 313C-2 DP</b>	<b>6ES7 313-6CF03-0AB0</b>	<b>V2.0</b>	<b>2</b>			
X2	DP				1023*		
2.2	D16/D016				0..1	0..1	
2.4	Count				768..783	768..783	
3							
4							
5							
6							
7							
8							
9							
10							
11							

# Informacje diagnostyczne dostępne dla modułów CPU

		Symbolizowana informacja
		CPU lub jeden z modułów, który został sparametryzowany przez CPU sygnalizuje błąd (np. przerwanie diagnostyczne)
		CPU znajduje się w trybie rozruchu ( <i>STARTUP</i> )
		CPU znajduje się w trybie pracy ( <i>RUN</i> )
		CPU znajduje się w trybie wstrzymania ( <i>HOLD</i> )
		CPU znajduje się w trybie zatrzymania ( <i>STOP</i> )
		CPU znajduje się w trybie zatrzymania ( <i>STOP</i> ) wyzwolonym przez zatrzymanie innego CPU (konfiguracja wieloprocesorowa)
		Brak informacji diagnostycznych lub dane niekompletne, STEP7 nie może jednoznacznie określić stanu CPU

# Ikony symbolizujące status modułu



informacje związane z modułem sygnałowym



informacje związane z CPU



informacje związane z CPU z aktywną funkcją FORCE



informacje związane z procesorem komunikacyjnym



informacje związane z modułem funkcyjnym



























informacje związane z szyną montażową



informacje związane z rozproszoną stacją (DP Slave)

# Informacje diagnostyczne dostępne dla pozostałych modułów

					Symbolizowana informacja
					Moduł jest niedostępny lub nie odpowiada skonfigurowanemu
					Brak informacji diagnostycznych lub dane niekompletne, STEP7 nie może jednoznacznie określić stanu modułu
					Wystąpił błąd w module
					Moduł znajduje się w trybie pracy ( <i>RUN</i> )
					Moduł znajduje się w trybie zatrzymania ( <i>STOP</i> )
					Moduł znajduje się w trybie wstrzymania ( <i>HOLD</i> )

# Informacja o stanie CPU – zakładka Diagnostic Buffer

W jednostkach centralnych systemu SIMATIC S7-300/400 został zaimplementowany Bufor Diagnostyczny.

W buforze diagnostycznym są informacje o: błędach modułu, błędach systemowych CPU, zmianach trybu pracy CPU, błędach w programie sterowania.

W buforze tym najnowsze zdarzenie zawsze zapisywane jest w rekordzie 1, cała reszta jest przesuwana i informacja o ostatnim jest tracona. Pojemność bufora zależy od wersji CPU i może wynosić od 100 do 500 zdarzeń dla CPU serii S7-300.

Jeżeli informacja zapisana w buforze diagnostycznym nie wystarczy do detekcji przyczyny przejścia CPU w stan STOP, wtedy można skorzystać z zakładki Stacks. Na danej zakładce wyświetlona jest zawartość stosu bloków( B Stack), na którym przechowywana jest informacja o blokach programowych, których realizacja nie została zakończona.

Na stosie bloków dostępna jest także informacja o otwartych w momencie wywołań kolejnych blokach danych.

# Informacja o stanie CPU – zakładka Diagnostic Buffer

Path: biurko\SIMATIC 300(1)\CPU 313C-2 DP    Operating mode of the CPU: RUN  
Status: OK    Not a force job

Time System    Performance Data    Communication    Stacks  
General    Diagnostic Buffer    Memory    Scan Cycle Time

Events:     Filter settings active     Time including CPU/local time difference

No.	Time of day	Date	Event
1	10:19:45.323 AM	12/01/2010	Mode transition from STARTUP to RUN
2	10:19:45.322 AM	12/01/2010	Request for automatic warm restart
3	10:19:44.909 AM	12/01/2010	Mode transition from STOP to STARTUP
4	10:19:33.453 AM	12/01/2010	Distributed I/Os: end of the synchronization with a D...
5	10:19:30.372 AM	12/01/2010	Power on backed up
6	05:23:46.730 PM	11/29/2010	Power failure
7	05:09:06.691 PM	11/29/2010	Mode transition from STARTUP to RUN
8	05:09:06.690 PM	11/29/2010	Request for manual warm restart

Details on Event: 1 of 10    Event ID: 16# 4302

Mode transition from STARTUP to RUN  
Startup information:  
- Time for time stamp at the last backed up power on  
- Single processor operation  
Current/last startup type:  
- Warm restart triggered via MPI: last power on backed up

Save As...    Settings...    Open Block    Help on Event

Close    Update    Print...    Help

Zawartość bufora  
diagnostycznego

Szczegóły zdarzenia

# Informacja o stanie CPU – zakładka Stacks

Path: biurko\SIMATIC 300(1)\CPU 313C-2 DP      Operating mode of the CPU: STOP

Status: Error      Not a force job

General      Diagnostic Buffer      Memory      Scan Cycle Time

Time System      Performance Data      Communication      Stacks

B Stack:

Block	Symbol	1st DB	2nd DB
OB1	...	DB10	...

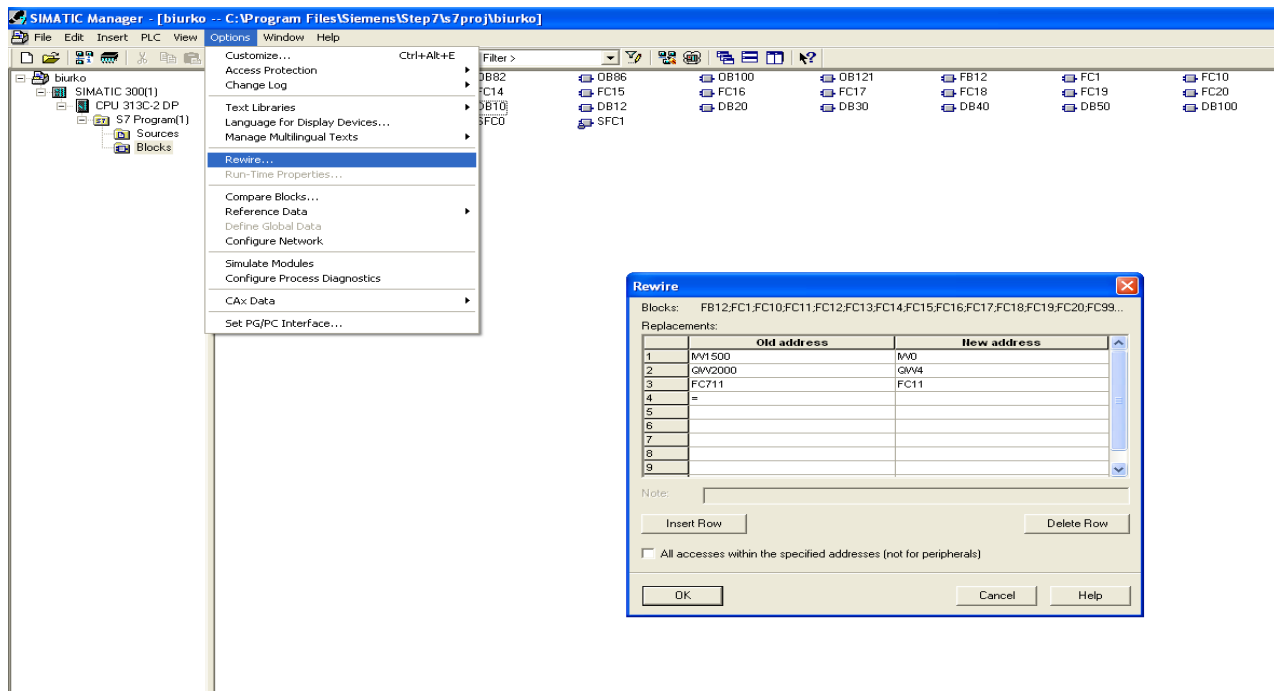
Callout: Zawartość stosu bloków

Buttons: Stack..., Stack..., Nesting Stack..., Open Block, Close, Update, Print..., Help



# Wykorzystanie funkcji „Rewire”

Korzystając z opcji Rewire dostępnej z menu **Options** w aplikacji SIMATIC Manager po zaznaczeniu pojemnika Blocks możliwe jest zastąpienie adresów komórek pamięci, numerów bloków programowych, liczników, układów czasowych dotychczas wykorzystywanych w programie przez nowe.



The screenshot shows the SIMATIC Manager interface with the 'Options' menu open and the 'Rewire...' option selected. The 'Rewire' dialog box is displayed, showing a list of replacements for memory addresses and program blocks. The dialog box has a title bar 'Rewire' and a close button. The main area contains a table with columns 'Old address' and 'New address'. Below the table are buttons for 'Insert Row' and 'Delete Row', a checkbox for 'All accesses within the specified addresses (not for peripherals)', and 'OK', 'Cancel', and 'Help' buttons.

	Old address	New address
1	MV1500	MV0
2	GMV2000	GMV4
3	FC711	FC11
4	=	
5		
6		
7		
8		
9		

# Aplikacja „Reference Data”

Aplikacja Reference Data przeznaczona jest:

- informowania o miejscach wykorzystania argumentu (np. wejście, wyjścia, timera) w programie
- informacja o wolnych i wykorzystanych komórkach pamięci
- podgląd struktury programu ( wywoływanie poszczególnych bloków programowych)

Dzięki tym informacją możliwe jest

- szybkie określenie warunku sterującego danym wyjściem
- bezpieczne uzupełnienie/modyfikacja istniejącego projektu

W kolumnie *Adress* wyświetlany jest adres zmiennej np. Q 5.3

Kolumna *Blocks* informuje o sposobie wykorzystania danego elementu

*Language* informuje o formie zapisu programu w jakiej zapisane zostało dane odwołanie

W kolumnie *Details* wyświetlane są szczegółowo informacje dotyczące wykorzystania danego elementu: numer segmentu( Networku), w którym element został wykorzystany np. Nw 1, numer instrukcji w określonym segmencie oraz rodzaj instrukcji odwołującej się do tego elementu

# Aplikacja „Reference Data”

Ref - [S7-Programm(1) (Cross-references): 2 -- Voss\_Butyl\_6\SIMATIC 300(1)\CPU 313C-2 DP]

Reference Data Edit View Window Help

Filtered

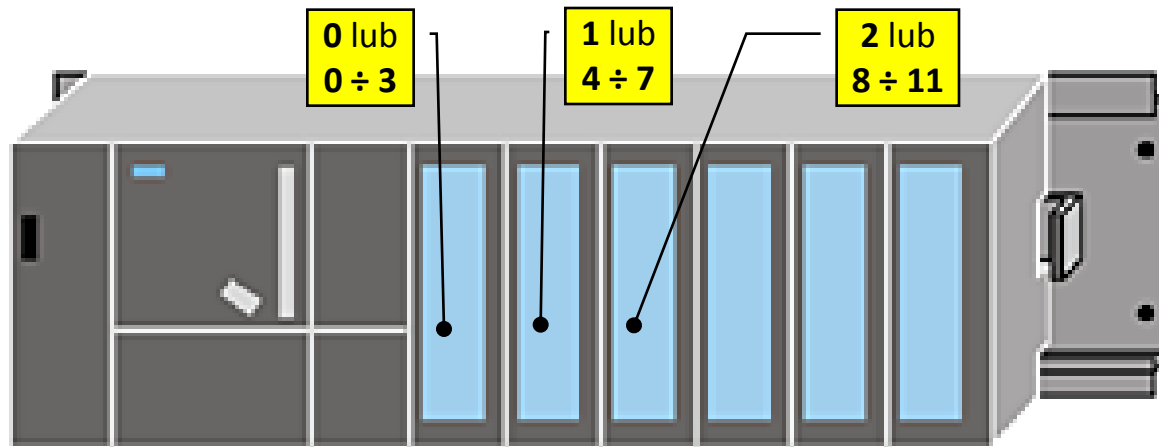
Address (symbol) /	Block (symbol)	Typ	Langua	Location	Location
Q 3.4	FC20 (Pneumatik)	W	LAD	Nw 24 /=	
Q 3.5	FC20 (Pneumatik)	W	LAD	Nw 25 /=	
Q 3.6	FC20 (Pneumatik)	W	LAD	Nw 26 /=	
Q 3.7	FC20 (Pneumatik)	W	LAD	Nw 27 /=	
Q 4.0	FC20 (Pneumatik)	W	LAD	Nw 2 /=	
Q 4.1	FC10 (Schrittkette)	R	LAD	Nw 6 /A	Nw 6 /A
Q 4.2	FC20 (Pneumatik)	W	LAD	Nw 4 /=	
Q 4.3	FC10 (Schrittkette)	R	LAD	Nw 6 /A	
Q 4.4	FC20 (Pneumatik)	W	LAD	Nw 6 /=	
Q 4.5	FC10 (Schrittkette)	R	LAD	Nw 6 /A	
Q 4.6	FC20 (Pneumatik)	W	LAD	Nw 20 /=	
Q 4.7	FC20 (Pneumatik)	W	LAD	Nw 21 /=	
Q 5.0	FC20 (Pneumatik)	W	LAD	Nw 17 /=	
Q 5.2	FC20 (Pneumatik)	W	LAD	Nw 19 /=	
Q 5.4	FC20 (Pneumatik)	W	LAD	Nw 16 /=	
Q 5.5	FC20 (Pneumatik)	W	LAD	Nw 17 /=	
Q 5.6	FC20 (Pneumatik)	W	LAD	Nw 18 /=	
Q 5.7	FC20 (Pneumatik)	W	LAD	Nw 19 /=	
T 1	FC110 (Stoerungen_1)	R	LAD	Nw 4 /A	
T 10	FC110 (Stoerungen_1)	R	LAD	Nw 5 /A	
T 11	FC110 (Stoerungen_1)	R	LAD	Nw 6 /A	
T 12	FC110 (Stoerungen_1)	R	LAD	Nw 7 /A	
T 13	FC110 (Stoerungen_1)	R	LAD	Nw 8 /A	
T 14	FC110 (Stoerungen_1)	R	LAD	Nw 9 /A	
T 15	FC110 (Stoerungen_1)	R	LAD	Nw 10 /A	
T 16	FC110 (Stoerungen_1)	R	LAD	Nw 11 /A	
T 17	FC110 (Stoerungen_1)	R	LAD	Nw 12 /A	
T 18	FC110 (Stoerungen_1)	R	LAD	Nw 13 /A	
T 19	FC110 (Stoerungen_1)	R	LAD	Nw 14 /A	
T 20	FC110 (Stoerungen_1)	R	LAD	Nw 15 /A	
T 21	FC110 (Stoerungen_1)	R	LAD	Nw 16 /A	
T 22	FC110 (Stoerungen_1)	R	LAD	Nw 17 /A	
T 23	FC110 (Stoerungen_1)	R	LAD	Nw 18 /A	



# Adresowanie PLC, typy zmiennych, typy danych

# Adresowanie modułów PLC

Adresowanie modułów przebiega podobnie jak adresowanie pamięci. W przypadku wejść i wyjść binarnych podajemy numer modułu i po kropce numer zacisku a w przypadku modułów analogowych tylko numer zacisku (adresowanie wyrazowe). Numer modułu zależy od jego umiejscowienia na szynie. Pierwszy moduł otrzymuje adres „0” a następne „1”, „2” itd. lub „4”, „8”, itd. w zależności od typu sterownika i modułów.



Np. aby odczytać czujnik podłączony do modułu nr „0” i zacisku „3”      wpisujemy: **0.3**

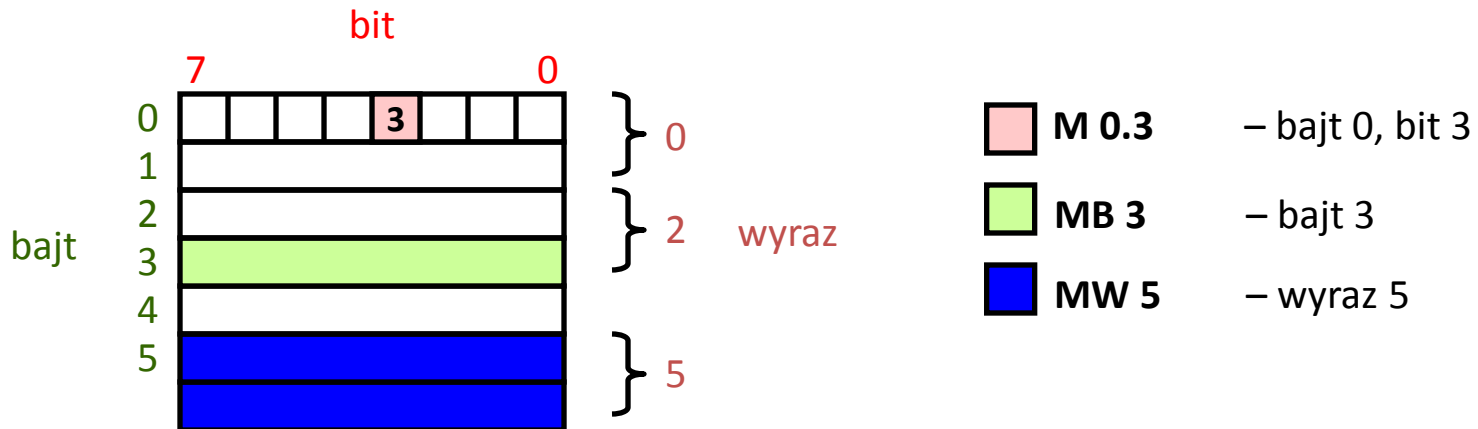
# Operandy

Przed opisem numerycznym wejścia, wyjścia lub pamięci dodaje się symbol literowy identyfikujący dany adres. Ponieważ dopuszczalny jest zapis w języku angielskim i niemieckim poniżej przedstawiono zapis niektórych operandów.

	oznaczenie		bitowo	bajtowo	wyrazowo
	ang.	niem.			
wejścia	I	E	x	x	x
wyjścia	Q	A	x	x	x
flagi	F	M	x	x	x
dane	D	D		x	x
timer	T	T		x	
licznik	C	Z		x	
stałe	K	K		x	x

# Adresowanie pamięci sterownika PLC

W pamięci sterownika wyodrębniona jest pewna ilość miejsc do przechowywania chwilowych wyników operacji. W sterownikach PLC rozróżniamy 4 tryby adresowania: bitowo, bajtowo, wyrazowo oraz przy pomocy dwóch słów. Adresując słownie operujemy na 16-tu bitach i przy pomocy dwóch słów na 32-ch bitach.



# Typy danych

Wszystkie informacje przychodzące do sterownika „I” ( wejścia cyfrowe, analogowe itd. ), wychodzące ze sterownika „Q” (wyjścia cyfrowe .... itd. ) oraz wewnętrzne sterownika „M” ( zmienne czyli markery ) .

Podstawową jednostką „miary” jest **bit** może on przyjmować tylko dwie wartości – 0 lub 1 kolejne jednostki składają się z odpowiedniej ilości bitów, jak w tabeli poniżej:

oznaczenie	typ danej	ilość bitów	zakres wartości	opis
<b>BOOL</b>	Boolean	1	0-1	Zmienne tego typu mogą przyjmować tylko dwie wartości czyli "0" lub "1" .
<b>BYTE</b>	ciąg 8 bitów	8	nie posiada wartości numerycznej	
<b>WORD</b>	ciąg 16 bitów	16	nie posiada wartości numerycznej	
<b>DWORD</b>	ciąg 32 bitów	32	nie posiada wartości numerycznej	
<b>LWORD</b>	ciąg 64 bitów	64	nie posiada wartości numerycznej	



# Typy danych

Typy zmiennych o wartości wyrażonej w systemie dziesiętnym, są to liczby całkowite (integer), których wartość wyliczana jest ze stanu odpowiednich bitów (konwersja z zapisu dwójkowego na zapis dziesiętny):

<b>SINT</b>	short integer	8	-128 : 127	"Krótka" liczba całkowita .
<b>INT</b>	integer	16	-32 768 : 32 767	Liczba całkowita .
<b>DINT</b>	double integer	32	$-2^{31} : (2^{31}-1)$	"Podwójna" liczba całkowita .
<b>LINT</b>	long integer	64	$-2^{63} : (2^{63}-1)$	"Długa" liczba całkowita .
<b>USINT</b>	unsigned Short integer	8	0 : 255	"Krótka" liczba całkowita nieznakowana (przyjmuje tylko wartości dodatnie).
<b>UINT</b>	unsigned integer	16	0 : 65535	Liczba całkowita nieznakowana (przyjmuje tylko wartości dodatnie).
<b>UDINT</b>	unsigned double integer	32	$0 : (2^{32}-1)$	" Podwójna "liczba całkowita nieznakowana (przyjmuje tylko wartości dodatnie).
<b>ULINT</b>	unsigned long integer	64	$0 : (2^{64}-1)$	" długa "liczba całkowita nieznakowana (przyjmuje tylko wartości dodatnie).

# Bloki dostępne w STEP 7

W STEP 7 programista ma do dyspozycji następujące rodzaje bloków:

- **Bloki organizacyjne – OB** – stanowią interfejs pomiędzy systemem operacyjnym sterownika a programem użytkowym. Program zapisany w OB zostanie wykonany w momencie wystąpienia odpowiedniego zdarzenia (przerwania), z którym dany blok OB jest skojarzony.  
Przykładowe bloki OB i ich zdarzenia:
  - **OB 35** - blok organizacyjny przerwania cyklicznych
  - **OB 100** - blok przerwania magistrali
  - **OB 121** - warm restart sterownika
  - **OB 100** - błąd programowy
- **Funkcje – FC** – realizują zadania sterownicze , określone przez zapisany w nich program. Funkcje mogą być parametryzowane (możliwe jest określenie parametrów, na których funkcja powinna wykonywać operacje). Dzięki temu funkcja może realizować te same zadania na różnych zmiennych. Aby funkcja była wykonywana, należy ją wywołać w bloku organizacyjnym lub w innym bloku programowym ( FC lub FB) wywoływanym bezpośrednio lub pośrednio w bloku OB

# Bloki dostępne w STEP 7

- **Bloki funkcyjne - FB** – właściwości tych bloków są identyczne jak funkcji. Różnicą jest to, że bloki funkcyjne posiadają pamięć w postaci bloków danych. Dzięki temu blok FB jest bardziej uniwersalny w użyciu niż funkcja.

Ponadto można skorzystać z:

- **Bloków danych – DB** - które, nie zawierają programu sterowania, lecz mogą służyć do przechowywania danych, zbioru parametrów dla programu, wyników realizacji programu, wyników pomiarów zmiennych procesowych itp.
- **Systemowych bloków funkcyjnych/funkcji – SFC/SFB** – są to zintegrowane, stworzone przez producenta bloki ( są one częścią systemu operacyjnego), które mogą być wykorzystane w programie użytkowym. Są one zapisane w jednostce centralnej i dlatego też nie ma potrzeby ładowania ich do sterownika wraz z programem sterowania. Nie można ich także zmieniać ich zawartości.



# Tworzenie i edycja bloku danych w programie

Blok danych „widziany” jest przez program jako ciąg bajtów. Podział bloku danych na pola pozwala na lepszą identyfikację zapisanych w nim danych. W trakcie edycji bloku danych w kolumnie *Address* pojawia się adres danego pola, który powinien być wykorzystany przez rozkazy odwołujące się do elementów zapisanych w bloku danych.

Blok danych ( lub pole w DB) może zostać zadeklarowane jako tablica. W takim wypadku jako typ pola należy wpisać *Array*, a następnie w nawiasie należy określić ilość elementów tablicy, np. [1..30] – oznacza 30 elementów. Zakres indeksów w nawiasach nie jest powiązany z adresami w DB( te zawsze zaczynają się od 0), następnie w kolejnej linii określany jest typ pola w tablicy np. *Real*.

Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	stan_DI	WORD	W#16#0	stan słowa wejściowego
+2.0	stan_AI	INT	0	stan wejścia analogowego
+4.0	napięcie_1	REAL	0.000000e+000	zmierzone napięcie
+8.0	czas_otwarcia	S5TIME	S5T#25S	czas otwarcia zaworu
+10.0	pomiary	ARRAY[1..20]		20 ostatnich pomiarów napięcia
+4.0		REAL		
+90.0	zezwo_praca	BOOL	FALSE	zezwoenie na uruchomienie pompy
+92.0		END_STRUCT		

↓  
długość bloku danych w bajtach (np. 92 bajty)

↓  
adres pola w DB

↓  
nazwa pola

↓  
typ pola

↓  
wartość początkowa

↓  
komentarz

↓  
definicja pola w DB jako tablicy

# Tworzenie i edycja bloku danych w programie

The screenshot displays the SIMATIC Manager interface with three 'Properties - Data Block' dialog boxes open. The top dialog shows 'DB1' as a 'Shared DB'. The middle dialog shows 'DB1' as an 'Instance DB' linked to 'FB12'. The bottom dialog shows 'DB1' as a 'DB of type' 'UDT1'. The background shows a project tree and a list of data blocks.

**Blok danych o dowolnej strukturze**

**Blok danych powiązany z blokiem funkcyjnym**

**Blok danych o strukturze określonej przez typ danych zdefiniowany przez użytkownika**

# Typy zmiennych definiowane przez użytkownika - UDT

Typy zmiennych definiowane przez użytkownika – UDT ( User-Defined Data Types) stanowią strukturę, która później może zostać wykorzystana w programie:

- Jako założony typ zmiennej w programie ( np. W bloku FC,FB,DB), ale także w trakcie definicji bloku danych
- Jako wzorzec przydatny w trakcie tworzenia bloku danych o identycznej strukturze ( w trakcie pierwszego otwierania DB należy określić ten blok jako „ Data Block referencing a User-Defined Data Type” – patrz folia „Tworzenie bloków danych”).

Proces tworzenia UDT jest bardzo podobny do tworzenia DB. UDT to deklaracja, nie jest ładowany do CPU.

## Przykład definicji UDT

Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	data	DATE	D#1990-1-1	
+2.0	godzina	TIME_OF_DAY	TOD#0:0:0.0	
+6.0	rodzaj	BYTE	B#16#0	
+8.0	adres	INT	0	
=10.0		END_STRUCT		

# Typy zmiennych definiowane przez użytkownika - UDT

## Deklaracja

Address	Name	Type	Initial value
*0.0		STRUCT	
+0.0	ALARMY	ARRAY[1..5]	
*10.0		UDT1	
=50.0		END_STRUCT	

## Podgląd

Address	Name	Type	Initial value	Actual value	Comment
0.0	ALARMY[1].data	DATE	D#1990-1-1	D#1990-1-1	
2.0	ALARMY[1].godzina	TIME_OF_DAY	TOD#0:0:0.0	TOD#0:0:0.0	
6.0	ALARMY[1].rodzaj	BYTE	B#16#0	B#16#0	
8.0	ALARMY[1].adres	INT	0	0	
10.0	ALARMY[2].data	DATE	D#1990-1-1	D#1990-1-1	
12.0	ALARMY[2].godzina	TIME_OF_DAY	TOD#0:0:0.0	TOD#0:0:0.0	
16.0	ALARMY[2].rodzaj	BYTE	B#16#0	B#16#0	
18.0	ALARMY[2].adres	INT	0	0	
20.0	ALARMY[3].data	DATE	D#1990-1-1	D#1990-1-1	
22.0	ALARMY[3].godzina	TIME_OF_DAY	TOD#0:0:0.0	TOD#0:0:0.0	
26.0	ALARMY[3].rodzaj	BYTE	B#16#0	B#16#0	
28.0	ALARMY[3].adres	INT	0	0	
30.0	ALARMY[4].data	DATE	D#1990-1-1	D#1990-1-1	
32.0	ALARMY[4].godzina	TIME_OF_DAY	TOD#0:0:0.0	TOD#0:0:0.0	
36.0	ALARMY[4].rodzaj	BYTE	B#16#0	B#16#0	
38.0	ALARMY[4].adres	INT	0	0	
40.0	ALARMY[5].data	DATE	D#1990-1-1	D#1990-1-1	
42.0	ALARMY[5].godzina	TIME_OF_DAY	TOD#0:0:0.0	TOD#0:0:0.0	
46.0	ALARMY[5].rodzaj	BYTE	B#16#0	B#16#0	
48.0	ALARMY[5].adres	INT	0	0	



# Tworzenie i edycja bloku danych w programie

Funkcja SFC 22 pozwala na tworzenie bloków danych w programie. Funkcja ta zawsze tworzy tylko jeden blok DB. Istnieje możliwość określenia numeru tworzonego DB z zakresu ( stworzony blok będzie miał najniższy wolny numer).

## Funkcja SFC 22 „CREAT\_DB”

Zmienna	Dekl.	Typ		Znaczenie
LOW_LIMIT	INPUT	WORD	I, Q,M, D, L, stała	Dolna granica numerów dla tworzonego bloku danych
UP_LIMIT	INPUT	WORD	I, Q,M, D, L, stała	Górna granica numerów dla tworzonego bloku danych
COUNT	INPUT	WORD	I, Q,M, D, L, stała	Rozmiar tworzonego bloku danych (w bajtach)
RET_VAL	OUTPUT	INT	I, Q,M, D, L	Informacja o ewentualnym błędzie jaki mógł powstać w trakcie realizacji funkcji
DB_NUMBER	OUTPUT	WORD	I, Q,M, D, L	Numer stworzonego bloku danych



# Tworzenie i edycja bloku danych w programie

Funkcja SFC 20 pozwala na kopiowanie obszarów pamięci. Warunkiem poprawnego działania funkcji jest to, aby obszar źródłowy i docelowy nie nakładały się.

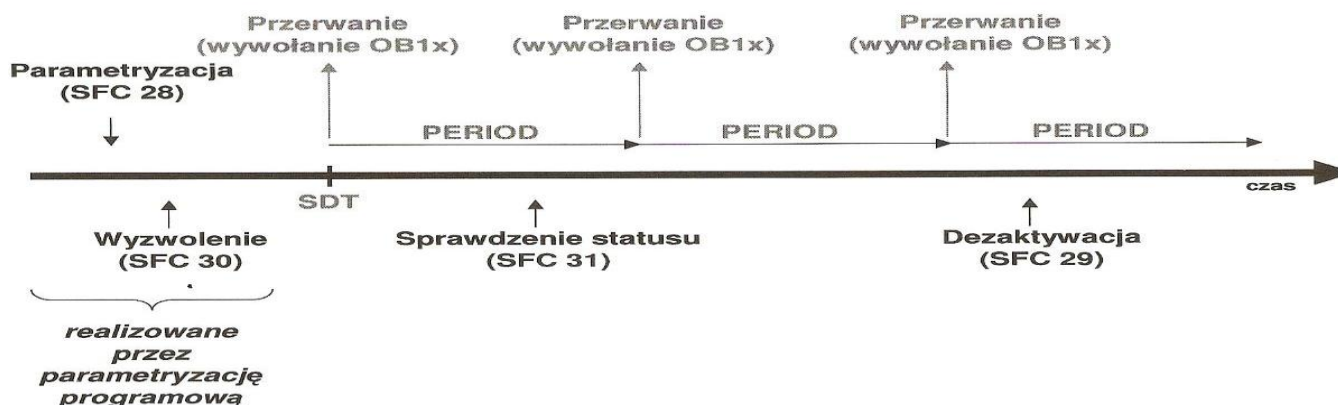
## Funkcja SFC 20 „BLKMOV”

Zmienna	Dekl.	Typ		Znaczenie
SRCBLK	INPUT	ANY	I, Q,M, D, L,	Wskaźnik na obszar, którego zawartość powinna być skopiowana
RET_VAL	OUTPUT	INT	I, Q,M, D, L	Informacja o ewentualnym błędzie jaki mógł powstać w trakcie realizacji funkcji
DSTBLK	OUTPUT	ANY	I, Q,M, D, L	Wskaźnik na obszar, do którego zawartość obszaru wskazywanego przez SRCBLK powinna być skopiowana

# Przerwania czasu rzeczywistego OB10-OB17

Przerwania czasu rzeczywistego są zdarzeniami generowanymi cyklicznie od określonego punktu początkowego. Minimalny cykl generacji przerwania czasu rzeczywistego to 1 minuta. Przerwania te można parametryzować z poziomu aplikacji Hardware Configuration lub z poziomu programu użytkownika. Parametryzacja z poziomu programu wykorzystywana jest przy pomocy funkcji SFC 28 „Set Tint”. Jeżeli przerwania te zostały sparametryzowane z poziomu programu (SFC 28) należy je dodatkowo aktywować przy pomocy funkcji SFC 30 „Act Tint”. Istnieje także możliwość dezaktywacji przerwania przy pomocy SFC 29 „Can Tint”.

Za obsługę przerwania czasu rzeczywistego odpowiedzialne są bloki OB grupy 10



# Przerwania czasu rzeczywistego OB10-OB17

Zmienna	Typ	Znaczenie
OB1x_EV_CLASS	BYTE	Identyfikator zdarzenia: B#16#11 – przerwanie jest aktywne
OB1x_STRT_INFO	BYTE	Żądanie obsługi zdarzenia przez OB 10 (B#16#11) Żądanie obsługi zdarzenia przez OB 11 (B#16#12) ... Żądanie obsługi zdarzenia przez OB 17 (B#16#18)
OB1x_PRIORITY	BYTE	Klasa priorytetowa: 2
OB1x_OB_NUMBER	BYTE	Numer bloku OB (10-17)
OB1x_RESERVED_1	BYTE	Rezerwa
OB1x_RESERVED_2	BYTE	Rezerwa
OB1x_PERIOD_EXE	WORD	Częstotliwość wywoływania (patrz parametryzacja przerwania SFC 28)
OB1x_RESERVED_3	INT	Rezerwa
OB1x_RESERVED_4	INT	Rezerwa
OB1x_DATE_TIME	DT	Data i czas wywołania bloku OB

# Programowa parametryzacja przerwań OB 10-17 – SFC 28 „Set Tint”

## Funkcja SFC 28 „SET\_TINT”

Zmienna	Dekl.	Typ		Znaczenie
OB_NR	INPUT	INT	I, Q,M, D, L, stała	Numer OB, który powinien zostać uruchomiony
SDT	INPUT	DT	D, L, stała	Data i czas pierwszego wyzwolenia OB 1x
PERIOD	INPUT	WORD	I, Q,M, D, L, stała	Częstotliwość wywoływania od momentu SDT
RET_VAL	OUTPUT	INT	I, Q,M, D, L	Informacja o ewentualnym błędzie jaki mógł powstać w trakcie realizacji funkcji

# Programowa parametryzacja przerwań OB 10-17 – SFC 29 „Can Tint”

## Funkcja SFC 29 „CAN\_TINT”

Zmienna	Dekl.	Typ		Znaczenie
OB_NR	INPUT	INT	I, Q,M, D, L, stała	Numer OB, który powinien zostać dezaktywowany
RET_VAL	OUTPUT	INT	I, Q,M, D, L	Informacja o ewentualnym błędzie jaki mógł powstać w trakcie realizacji funkcji

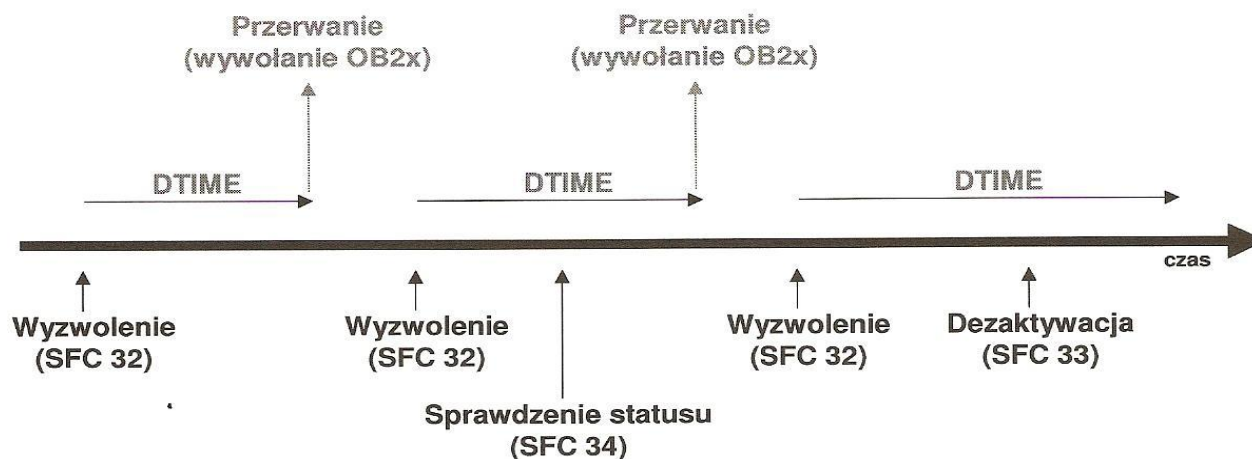
# Programowa parametryzacja przerwań OB 10-17 – SFC 30 „Act Tint”

## Funkcja SFC 30 „ACT\_TINT”

Zmienna	Dekl.	Typ		Znaczenie
OB_NR	INPUT	INT	I, Q,M, D, L, stała	Numer OB, który powinien zostać aktywowany
RET_VAL	OUTPUT	INT	I, Q,M, D, L	Informacja o ewentualnym błędzie jaki mógł powstać w trakcie realizacji funkcji

# Przerwania opóźnione w czasie OB 20-23

Za obsługę przerw opóźnionych w czasie odpowiedzialne są bloki OB z grupy 20. Przerwania te są wyzwalane przy pomocy funkcji SFC 32 „Srt Dint” w programie użytkownika. Przerwanie wystąpi po czasie określonym w parametrze SFC 32. Istnieje możliwość „odwołania” przerwania przy pomocy funkcji SFC 29 „Can Dint”. Jeżeli funkcja ta zostanie wywołana przed upływem czasu po jakim powinno wystąpić przerwanie – przerwanie to nie zostanie wygenerowane. Mechanizm przerw opóźnionych w czasie może zostać wykorzystany do realizacji zadań występujących po sobie w określonej sekwencji czasowej.



# Programowa parametryzacja przerwań OB 20-23 – SFC 32 „Srt Dint”

## Funkcja SFC 32 „SRT\_DINT”

Zmienna	Dekl.	Typ		Znaczenie
OB_NR	INPUT	INT	I, Q,M, D, L, stała	Numer OB, który powinien zostać uruchomiony (OB20-23)
DTIME	INPUT	TIME	I, Q,M, D, L, stała	Czas opóźnienia
SIGN	INPUT	WORD	I, Q,M, D, L, stała	Identyfikator, który zostanie przypisany do parametru OB2x_SIGN w bloku obsługującym to przerwanie
RET_VAL	OUTPUT	INT	I, Q,M, D, L	Informacja o ewentualnym błędzie jaki mógł powstać w trakcie realizacji funkcji



# Programowa parametryzacja przerwań OB 20-23 – SFC 33 „Can Dint”

## Funkcja SFC 33 „CAN\_DINT”

Zmienna	Dekl.	Typ		Znaczenie
OB_NR	INPUT	INT	I, Q,M, D, L, stała	Numer OB, który powinien zostać dezaktywowany
RET_VAL	OUTPUT	INT	I, Q,M, D, L	Informacja o ewentualnym błędzie jaki mógł powstać w trakcie realizacji funkcji

# Programowa parametryzacja przerwań OB 20-23 – SFC 34 „Qry Dint”

## Funkcja SFC 34 „QRY\_DINT”

Zmienna	Dekl.	Typ		Znaczenie
OB_NR	INPUT	INT	I, Q,M, D, L, stała	Numer OB, którego status powinien zostać odczytany
RET_VAL	OUTPUT	INT	I, Q,M, D, L	Informacja o ewentualnym błędzie jaki mógł powstać w trakcie realizacji funkcji
STATUS	OUTPUT	WORD	I, Q,M, D, L	Status wybranego przerwania opóźnionego w czasie



# Wykorzystanie bloków organizacyjnych reakcji na błędy

# Rodzaje błędów i bloki odpowiedzialne za ich obsługę

Klasa błędu	Rodzaj błędu	Nr OB	Priorytet
asynchroniczne	Błąd czasu	<b>OB 80</b>	26 (lub 28 jeżeli błąd powstał w trakcie rozruchu)
	Błąd zasilania	<b>OB 81</b>	
	Przerwanie diagnostyczne	<b>OB 82</b>	
	Wyjęcie/wstawienie modułu	<b>OB 83</b>	
	Błąd CPU	<b>OB 84</b>	
	Błąd sekwencji programu	<b>OB 85</b>	
	Błąd konfiguracji rozszerzającej	<b>OB 86</b>	
	Błąd komunikacyjny	<b>OB 87</b>	
	Błąd przetwarzania bloków obsługi przerw	<b>OB 88</b>	
synchroniczne	Błąd programowania	<b>OB 121</b>	priorytet bloku w którym powstał błąd
	Błąd dostępu do obszaru wejść/wyjść	<b>OB 122</b>	

# Błąd czasu - OB 80

System operacyjny CPU wywołuje blok OB 80 w przypadku wystąpienia w trakcie realizacji programu sterowania jednego z poniższych błędów:

- Przekroczenie czasu obiegu pętli programowej
- Pominięcie czasu wywołania przerwania czasu rzeczywistego w wyniku zmiany czasu systemowego (przesunięcia do przodu)
- Zbyt duże opóźnienie w trakcie realizacji programu dla danej klasy priorytetowej ( wywołanie danego bloku OB w przypadku, kiedy jego poprzednie wywołanie nie zostało zrealizowane)

Brak bloku OB 80 spowoduje przejście CPU w stan STOP

Blok OB 80 może zostać wykorzystany do:

- określenia, które przerwanie czasu rzeczywistego zostało pominięte
- korzystając z SFC 29 „Can Tint” możliwe jest skasowanie pominiętego przerwania tak, aby wywoływane były tylko i wyłącznie przerwania zgodnie z nowym czasem.

# Błąd czasu - OB 80

Zmienna	Typ	Znaczenie
OB80_EV_CLASS	BYTE	Identyfikator zdarzenia: B#16# <b>35</b> .
OB80_FLT_ID	BYTE	Kod błędu – identyfikacja zdarzenia.
OB80_PRIORITY	BYTE	Klasa priorytetowa: <b>26</b> .
OB80_OB_NUMBER	BYTE	Numer bloku OB (80).
OB80_RESERVED_1	BYTE	Zarezerwowany.
OB80_RESERVED_2	BYTE	Zarezerwowany.
OB80_ERROR_INFO	WORD	Informacja o błędzie (znaczenie zależne od kodu błędu).
OB80_ERR_EV_CLASS	BYTE	Klasa zdarzenia odpowiedzialnego za błąd.
OB80_ERR_EV_NUM	BYTE	Numer zdarzenia, które wywołało błąd.
OB80_OB_PRIORITY	BYTE	Kl. priorytetowa OB aktywnego w momencie wyst. zdarzenia.
OB80_OB_NUM	BYTE	Numer OB aktywnego w momencie wyst. zdarzenia.
OB80_DATE_TIME	DT	Data i czas wywołania bloku OB80.

# Błąd zasilania - OB 81

Blok OB 81 jest wywoływany przez system operacyjny CPU w sytuacji, gdy wystąpi jedna z poniższych usterek:

- Bateria wyczerpie się lub zostanie wyjęta
- Uszkodzony zostanie zasilacz 24 V

Blok OB 81 jest wywoływany dwukrotnie: w momencie wystąpienia i usunięcia błędu. Brak tego bloku w sterowniku nie powoduje przejścia w stan STOP jednostki CPU, jednak w buforze diagnostycznym pojawią się odpowiednie wpisy oraz zapalą się odpowiednie LED na panelach CPU oraz PS

Informacje zwracane przez argumenty OB 81 mogą zostać wykorzystane do:

- Określenia, który zasilacz uległ uszkodzeniu
- Zapalania lampki na stacji operatorskiej informującej o konieczności wymiany baterii.

Możliwe kody błędów:

- B#16#21 – przynajmniej jedna bateria podtrzymująca w CR jest wyczerpana bądź usunięta.
- B#16#22 – wystąpił problem z napięciem podtrzymującym w CR (BAF)

# Błąd zasilania - OB 81

Zmienna	Typ	Znaczenie
OB81_EV_CLASS	BYTE	Identyfikator zdarzenia: B#16#38 – przyczyna zanika B#16#39 – powstanie zdarzenia.
OB81_FLT_ID	BYTE	Kod błędu – identyfikacja zdarzenia.
OB81_PRIORITY	BYTE	Klasa priorytetowa: 26 (domyślnie - RUN) lub 28 dla rozruchu.
OB81_OB_NUMBER	BYTE	Numer bloku OB (81).
OB81_RESERVED_1	BYTE	Zarezerwowany.
OB81_RESERVED_2	BYTE	Zarezerwowany.
OB81_MDL_ADDR	WORD	Zarezerwowany.
OB81_RESERVED_3	BYTE	Ważne tylko dla kodów błędu B#16#31, B#16#32 oraz B#16#33.
OB81_RESERVED_4	BYTE	
OB81_RESERVED_5	BYTE	
OB81_RESERVED_6	BYTE	
OB81_DATE_TIME	DT	Data i czas wywołania bloku OB81.



# Przerwania diagnostyczne – OB 82

System operacyjny wywołuje blok OB 82 w sytuacji, kiedy moduł z wbudowanymi funkcjami diagnostycznymi ( po ich uaktywnieniu) zwróci informację o błędzie. Modułem tym może być moduł sygnałowy, ale również może to być stacja DP Slave. Blok ten jest wywoływany dwukrotnie : w momencie powstania i zaniku błędu. Brak OB 82 powoduje przejście w STOP CPU. Parametry OB 82 pozwalają na określenie modułu, w którym powstał błąd. Blok ten może zostać również wykorzystany do odczytania pełnej informacji diagnostycznej z uszkodzonego modułu.

Zmienna	Typ	Znaczenie
OB82_EV_CLASS	BYTE	Identyfikator zdarzenia: B#16#38 – przyczyna zanika B#16#39 – powstanie zdarzenia.
OB82_FLT_ID	BYTE	Kod błędu – identyfikacja zdarzenia (B#16#42).
OB82_PRIORITY	BYTE	Klasa priorytetowa: 26 (domyślnie - RUN) lub 28 dla rozruchu.
OB82_OB_NUMBER	BYTE	Numer bloku OB (82).
OB82_RESERVED_1	BYTE	Zarezerwowany.
OB82_IO_FLAG	BYTE	Moduł wejściowy: B#16#54. Moduł wyjściowy: B#16#55.
OB82_MDL_ADDR	WORD	Adres bazowy modułu zgłaszającego błąd.
OB82_MDL_DEFECT	BOOL	Moduł jest uszkodzony.
OB82_INT_FAULT	BOOL	Błąd wewnętrzny.
OB82_EXT_FAULT	BOOL	Błąd zewnętrzny.
OB82_PNT_INFO	BOOL	Błąd kanału.
...	...	...

# Błąd klasy priorytetowej – OB 85

Blok OB 85 jest wywoływany przez system operacyjny CPU w sytuacji, gdy wystąpi jedno z poniższych zdarzeń:

- Powstaje warunek wywołania OB lecz blok ten nie jest załadowany do CPU
- Wystąpi błąd dostępu do bloku danych skojarzonego z systemowym blokiem funkcyjnym
- Wystąpi błąd w trakcie odświeżania obszaru odwzorowania (moduł sygnałowy jest niedostępny lub uszkodzony)

Brak bloku OB 85 powoduje przejście CPU w stan STOP.

Zmienna	Typ	Znaczenie
OB85_EV_CLASS	BYTE	Identyfikator zdarzenia.
OB85_FLT_ID	BYTE	Kod błędu – identyfikacja zdarzenia.
OB85_PRIORITY	BYTE	Klasa priorytetowa: <b>26</b> (domyślnie - RUN) lub <b>28</b> dla rozruchu.
OB85_OB_NUMBER	BYTE	Numer bloku OB (85).
OB85_RESERVED_1	BYTE	Zarezerwowany.
OB85_RESERVED_2	BYTE	Zarezerwowany.
OB85_RESERVED_3	INT	Zarezerwowany.
OB85_ERR_EV_CLASS	BYTE	Klasa zdarzenia odpowiedzialnego za błąd.
OB85_ERR_EV_NUM	BYTE	Numer zdarzenia, które wywołało błąd.
OB85_OB_PRIORITY	BYTE	Kl. priorytetowa OB aktywnego w momencie wyst. Zdarzenia.
OB85_OB_NUM	BYTE	Numer OB aktywnego w momencie wyst. Zdarzenia.
OB85_DATE_TIME	DT	Data i czas wywołania bloku OB85.

# Błąd konfiguracji rozszerzającej – OB 86

System operacyjny CPU wywołuje OB 86 w przypadku wystąpienia następujących błędów:

- Problem z szyną rozszerzającą ( brak lub uszkodzenie modułu IM ewentualnie uszkodzenie kabla łączącego)
- Problem z dystrybucją zasilania pomiędzy szynami rozszerzającymi
- Brak dostępu/uszkodzenie stacji DP Slave

Brak bloku OB 86 powoduje przejście CPU w stan STOP

Zmienna	Typ	Znaczenie
OB86_EV_CLASS	BYTE	Identyfikator zdarzenia: B#16# <b>38</b> – przyczyna zanika B#16# <b>39</b> – powstanie zdarzenia.
OB86_FLT_ID	BYTE	Kod błędu – identyfikacja zdarzenia.
OB86_PRIORITY	BYTE	Klasa priorytetowa: <b>26</b> (domyślnie - RUN) lub <b>28</b> dla rozruchu.
OB86_OB_NUMBER	BYTE	Numer bloku OB (81).
OB86_RESERVED_1	BYTE	Zarezerwowany.
OB86_RESERVED_2	BYTE	Zarezerwowany.
OB86_MDL_ADDR	WORD	Znaczenie zależne od kodu błędu.
OB86_RACKS_FLTD	DWORD	Znaczenie zależne od kodu błędu.
OB86_DATE_TIME	DT	Data i czas wywołania bloku OB86.

# Błąd komunikacyjny – OB 87

Blok OB 87 jest wywoływany przez CPU w przypadku wystąpienia błędu komunikacyjnego w trakcie przekazywania danych z wykorzystaniem funkcji komunikacyjnych lub zmiennych globalnych. Przykładowymi błędami mogą być:

- Niewłaściwy identyfikator ramki odebranych w trakcie przekazywania zmiennych globalnych
- Brak możliwości przestania pakietów synchronizujących

Brak OB 87 powoduje przejście w stan STOP modułu CPU serii 300. Informacje udostępnione przez parametry bloku OB mogą być wykorzystane do:

- Określenia dokładnej przyczyny błędu
- Stworzenia bloku danych, w którym mają być przechowywane informacje statusowe dla pakietu zmiennych globalnych, jeżeli nie istnieje lub jego rozmiar jest nieodpowiedni

# Błąd komunikacyjny – OB 87

Zmienna	Typ	Znaczenie
OB87_EV_CLASS	BYTE	Identyfikator zdarzenia.
OB87_FLT_ID	BYTE	Kod błędu – identyfikacja zdarzenia.
OB87_PRIORITY	BYTE	Klasa priorytetowa: <b>26</b> (domyślnie - RUN) lub <b>28</b> dla rozruchu.
OB87_OB_NUMBER	BYTE	Numer bloku OB (87).
OB87_RESERVED_1	BYTE	Zarezerwowany.
OB87_RESERVED_2	BYTE	Zarezerwowany.
OB87_RESERVED_3	WORD	Znaczenie zależne od kodu błędu.
OB87_RESERVED_4	DWORD	Znaczenie zależne od kodu błędu.
OB87_DATE_TIME	DT	Data i czas wywołania bloku OB87.

# Błąd przetwarzania bloków obsługi przerwań – OB 88

Blok OB 88 jest wywoływany przez CPU w przypadku zakłócenia realizacji programu.

Przykładowymi przyczynami mogą być:

- Ilość zagnieźdżeń bloków obsługi błędów synchronicznych jest zbyt duża
- Ilość zagnieźdżeń wywołań bloków jest zbyt duża
- Błąd alokacji zmiennych lokalnych

Brak OB 88 powoduje przejście CPU w stan STOP

Zmienna	Typ	Znaczenie
OB88_EV_CLASS	BYTE	Identyfikator zdarzenia.
OB88_SW_FLT	BYTE	Kod błędu – identyfikacja zdarzenia.
OB88_PRIORITY	BYTE	Klasa priorytetowa: <b>28</b> .
OB88_OB_NUMBER	BYTE	Numer bloku OB (88).
OB88_BLK_TYPE	BYTE	Typ bloku w którym wystąpił błąd.
OB88_RESERVED_1	BYTE	Zarezerwowany.
OB88_FLT_PRIORITY	BYTE	Klasa priorytetowa bloku OB powodującego błąd.
OB88_FLT_OB_NUMBR	BYTE	Numer bloku OB generującego błąd.
OB88_BLK_NUM	WORD	Numer bloku z instrukcją, która spowodowała błąd
OB88_PRG_ADDR	WORD	Adres względny instrukcji powodującej błąd.
OB88_DATE_TIME	DT	Data i czas wywołania bloku OB88.

# Błąd programowania – OB 121

Blok OB 121 jest wywoływany w przypadku wystąpienia błędu programowania:

- Zaadresowany element ( np. komórka pamięci, licznik, ....) nie jest dostępny
- Wywoływany blok nie został załadowany do CPU

Brak bloku OB 121 powoduje przejście CPU w stan STOP

Zmienna	Typ	Znaczenie
OB121_EV_CLASS	BYTE	Identyfikator zdarzenia.
OB121_FLT_ID	BYTE	Kod błędu – identyfikacja zdarzenia.
OB121_PRIORITY	BYTE	Klasa priorytetowa = klasa OB w którym powstał błąd.
OB121_OB_NUMBER	BYTE	Numer bloku OB (121).
OB121_BLK_TYPE	BYTE	Typ bloku w którym powstał błąd (tylko CPU serii 400).
OB121_RESERVED_1	BYTE	Zarezerwowany.
OB121_FLT_REG	WORD	Źródło błędu - znaczenie zależne od kodu błędu.
OB121_BLK_NUM	WORD	Numer bloku z komendą MC7 w którym powstał błąd (tylko CPU serii 400).
OB121_PRG_ADDR	WORD	Adres względny rozkazu powodującego błąd (tylko CPU serii 400).
OB121_DATE_TIME	DT	Data i czas wywołania bloku OB121.

# Błąd dostępu do obszaru wejść/wyjść – OB 122

Blok OB 122 jest wywoływany w sytuacji, kiedy instrukcja zapisana w STEP 7 odwołuje się do komórki peryferii (obszar wejść lub wyjść), do której nie został przypisany żaden moduł sygnałowy w trakcie wykonywania ostatniego pełnego rozruchu,

Brak bloku OB 122 powoduje przejście CPU w stan STOP. Program zapisany w tym bloku może zostać wykorzystany do:

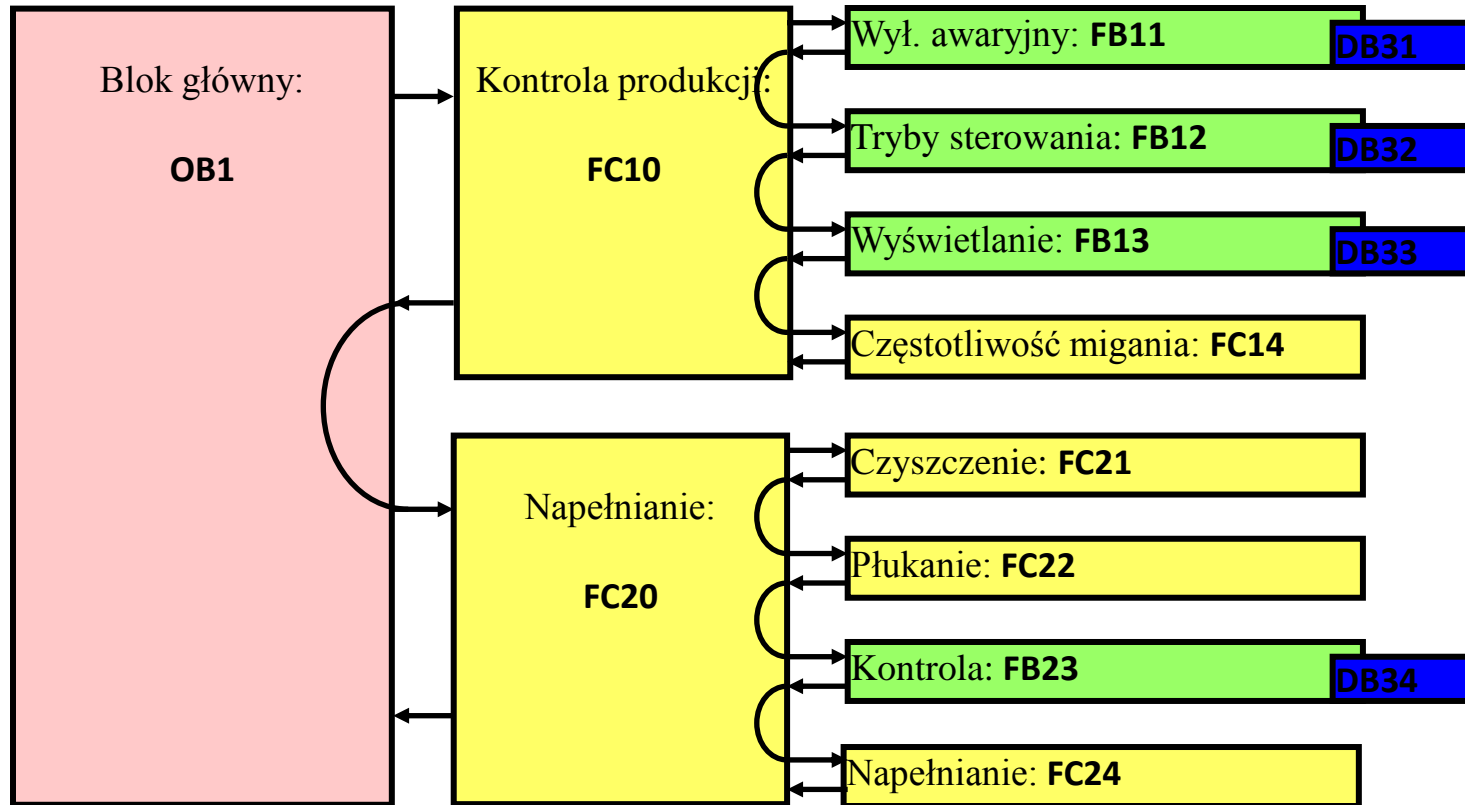
- Określenia komórki, do której nie został przypisany moduł,

Zmienna	Typ	Znaczenie
OB122_EV_CLASS	BYTE	Identyfikator zdarzenia.
OB122_SW_FLT	BYTE	Kod błędu – identyfikacja zdarzenia.
OB122_OB_NUMBER	BYTE	Numer bloku OB (122).
OB122_BLK_TYPE	BYTE	Typ bloku w którym powstał błąd (tylko CPU serii 400).
OB122_MEM_AREA	BYTE	Kod obszaru pamięci oraz rodzaju odwołania.
OB122_MEM_ADDR	WORD	Adres komórki do której nie można się odwołać.
OB122_BLK_NUM	WORD	Numer bloku z komendą MC7 w którym powstał błąd (tylko CPU serii 400).
OB122_PRG_ADDR	WORD	Adres względny rozkazu powodującego błąd (tylko CPU serii 400).
OB122_DATE_TIME	DT	Data i czas wywołania bloku OB122.

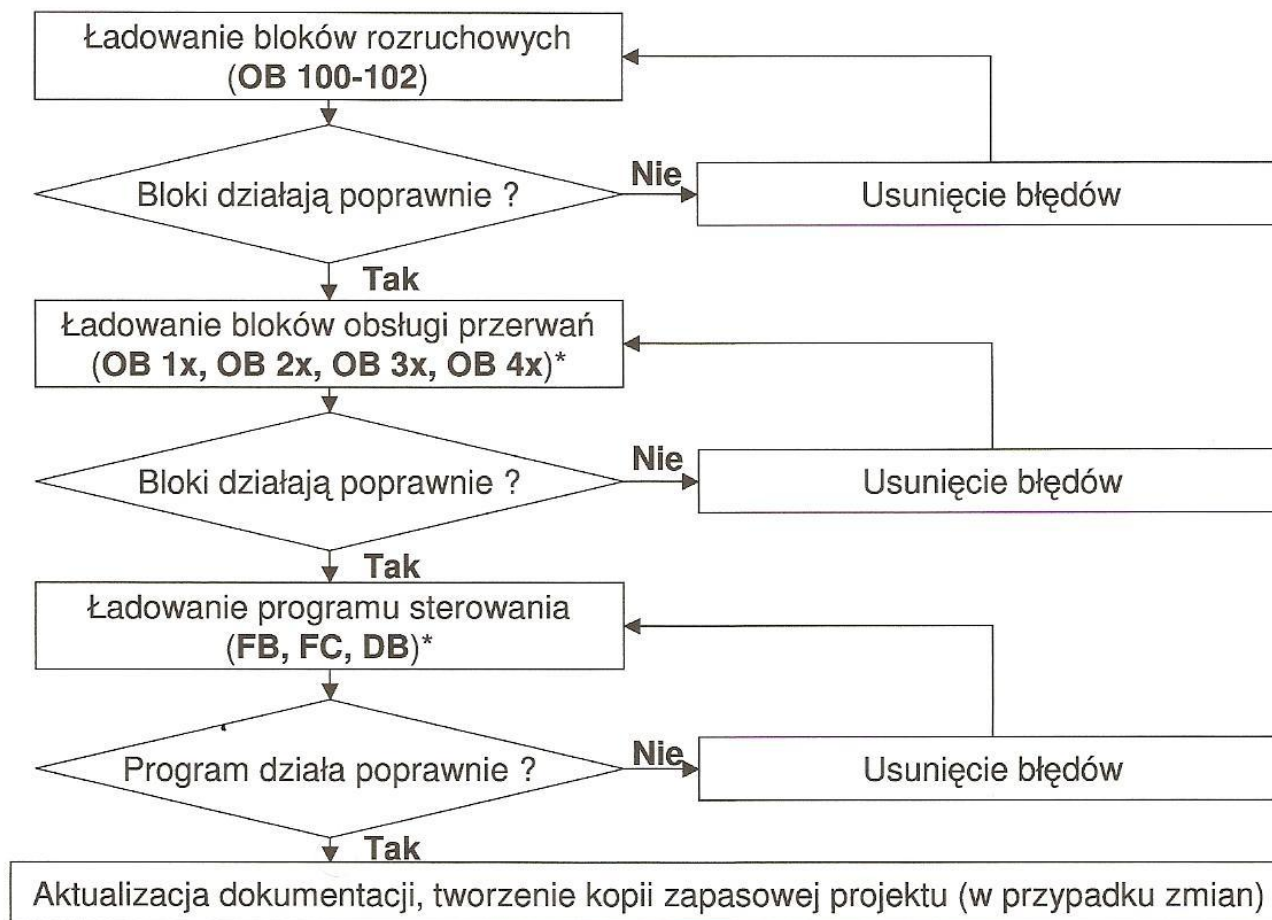


# Przykładowa struktura programu PLC

Program sterowników Simatic składa się z bloków o następującej przykładowej strukturze:



# Kolejność uruchamiania złożonego programu



# Rodzaje języków programowania

Firmy produkujące sterowniki programowalne dostarczają zwykle wraz z nimi środowisko programistyczne, pozwalające pisać aplikacje w jednym lub kilku językach programowania. Języki te są zwykle mniej lub bardziej dokładną implementacją zaleceń normy IEC 61131-3, która to norma definiuje następujące warianty:

- **LAD (Ladder Diagram)** - schemat stykowy (drabinkowy) – sposób zapisu programu w postaci podobnej do schematu elektrycznego, w którym styki, cewki i inne elementy zastąpione zostały odpowiednimi symbolami. Ten sposób reprezentacji, ze względu na duże podobieństwo do schematu elektrycznego, powinien być najbliższy elektrykom.
- **FBD (Function Block Diagram)** – schemat funkcyjny – pozwala na zapisanie programu w postaci podobnej do schematu układu elektronicznego, składającego się z szeregu bloków reprezentujących funkcje logiczne. Ta forma reprezentacji programu powinna ułatwić programowanie sterowników osobom, które znają i potrafią analizować schematy układów logicznych.
- **STL (Statement List)** - lista instrukcji – pozwala na zapisanie programu jako ciągu instrukcji ( rozkazów ). Jest to forma tekstowa podobna do języka assemblera wykorzystywanego do programowania procesorów jednoukładowych czy też komputerów PC. Forma ta jest najbliższa informatykom oraz elektronikom wykorzystującym układy programowalne

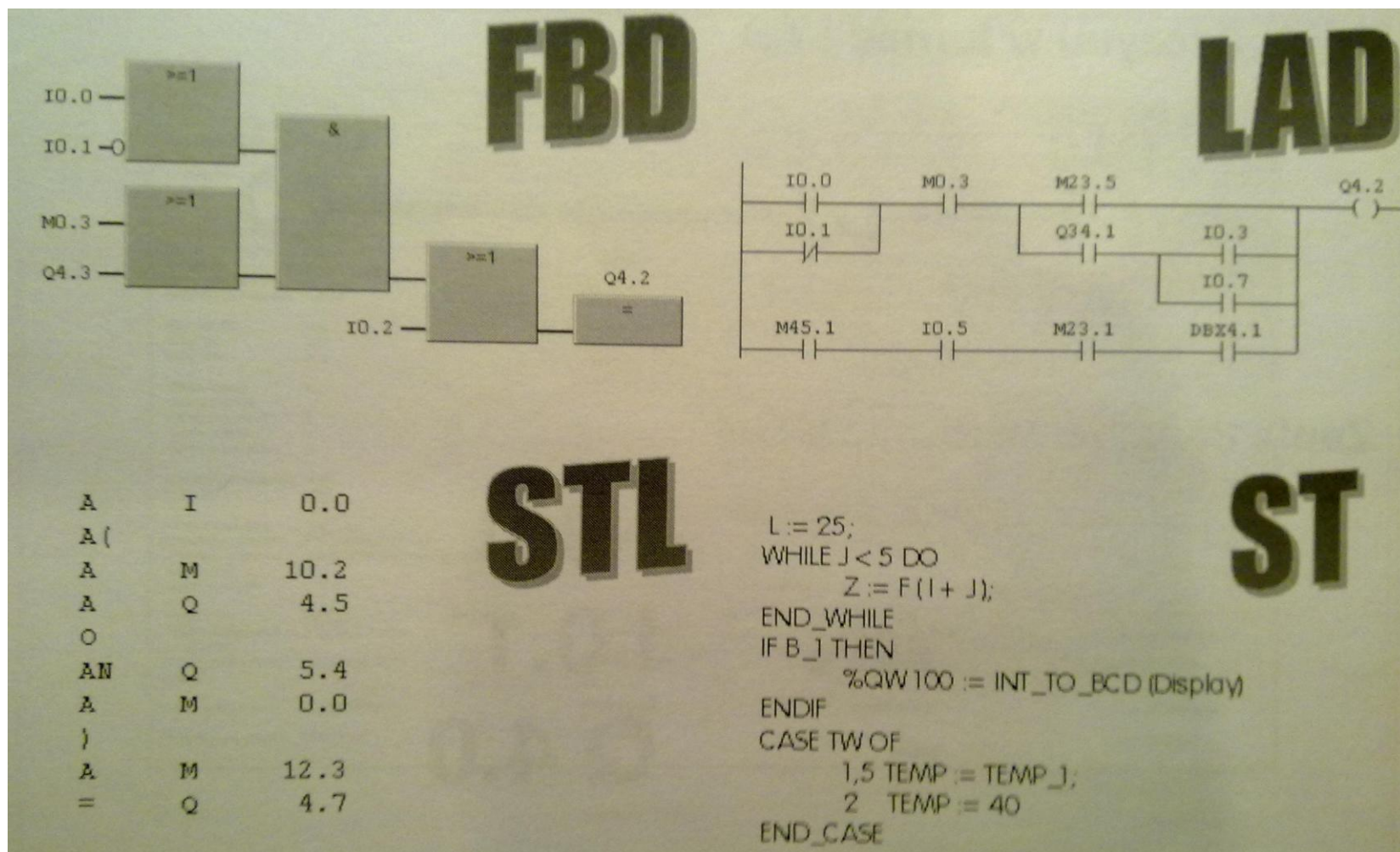
# Rodzaje języków programowania

- **ST ( Structured Text )** - tekst strukturyzowany – tekstowa forma opisu zadania zbliżona do języka Pascal wykorzystywanego do programowania komputerów PC. Daje do dyspozycji programiście możliwość zagnieżdżania instrukcji zapisywania iteracji, funkcji warunkowych itp.
- **SFC ( Sequential Function Chart)** - ułatwia opis zadań sekwencyjnych, gdzie podstawowymi elementami są stany ( kroki) oraz warunki przejścia pomiędzy stanami.

## Dane języki pojawią się w normie IEC 61131-1 od 1.12.2010

- **S7-HiGraph** - będzie opierać się na podziale zadania technologicznego na poszczególne jednostki funkcyjne. Charakterystyka wydajności obiektów technologicznych oraz jednostek funkcyjnych opisywane będą za pomocą wykresów stanów.
- **S7-Graph** – programowanie będzie polegać na krokowym wykonywaniu rozkazów, czekaniu na wykonanie rozkazów oraz spełnieniu warunków procesowych niezbędnych do przejścia do następnych kroków.

# Struktura języków programowania

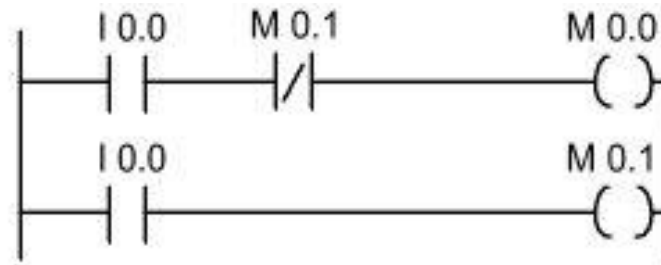


Rys.6 Przykładowa struktura języków kilku reprezentacji według normy IEC 61131-3

# Wykrywanie zbocza

Funkcje wykrywania zbocza mają za zadanie podanie sygnału 1 tylko na moment w którym sygnał na wejściu sterownika zmienia swój stan z 0 na 1 dla zbocza narastającego i z 1 na 0 dla zbocza opadającego a następnie powrót do wartości 0. Tutaj omówimy budowę takiej funkcji za pomocą podstawowych symboli języka drabinkowego .

Zbocze narastające ( *rising edge* )



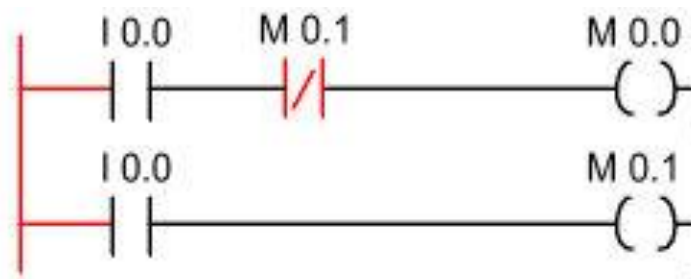
**I 0.0** - wejście sterownika na którym zbocze jest wykrywane

**M 0.0** - marker , który przyjmie **wartość "1"** na czas **jednego cyklu sterownika** po wykryciu zbocza

**M 0.1** - marker pomocniczy, „pamiętający” stan wejścia z poprzedniego cyklu.

# Działanie programu z boczna narastającego (P) cykl po cyklu

1. Stan markerów przed załączeniem wejścia I 0.0 Stan wejścia i markerów = 0



2. Załączono wejście I 0.0 i sygnał "1" został przeniesiony z lewej przez nie załączony jeszcze M 0.0 (=0) na prawo – ustawiając M0.0 na = 1

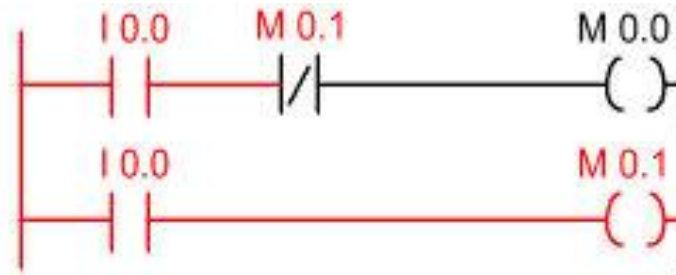


# Działanie programu z boczna narastającego (P) cykl po cyklu

3. Marker M 0.1 zostaje aktywowany ( = 1 ) . Warto zauważyć że sterownik nie zmienia wartości we wcześniejszych ( wyższych liniach ) , na koniec cyklu oba markery mają wartość = 1



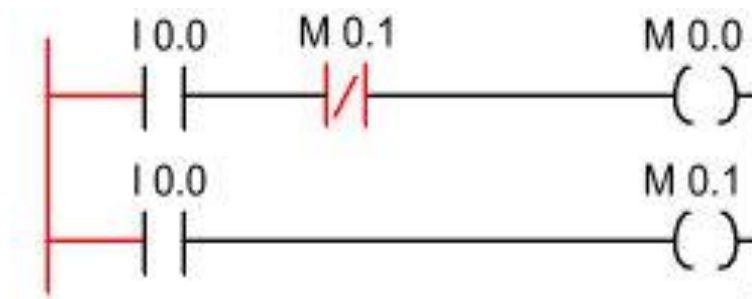
4. Marker M 0.1 = 1 powoduje przerwę i wyłączenie markera M 0.0 ( =0 )





# Działanie programu zbocza narastającego (P) cykl po cyklu

Wszystko pozostaje bez zmian do momentu utrzymywania się stanu "1" na wejściu I 0.0 . Po zmianie stanu wejścia I 0.0 na "0" funkcja wraca do stanu początkowego.

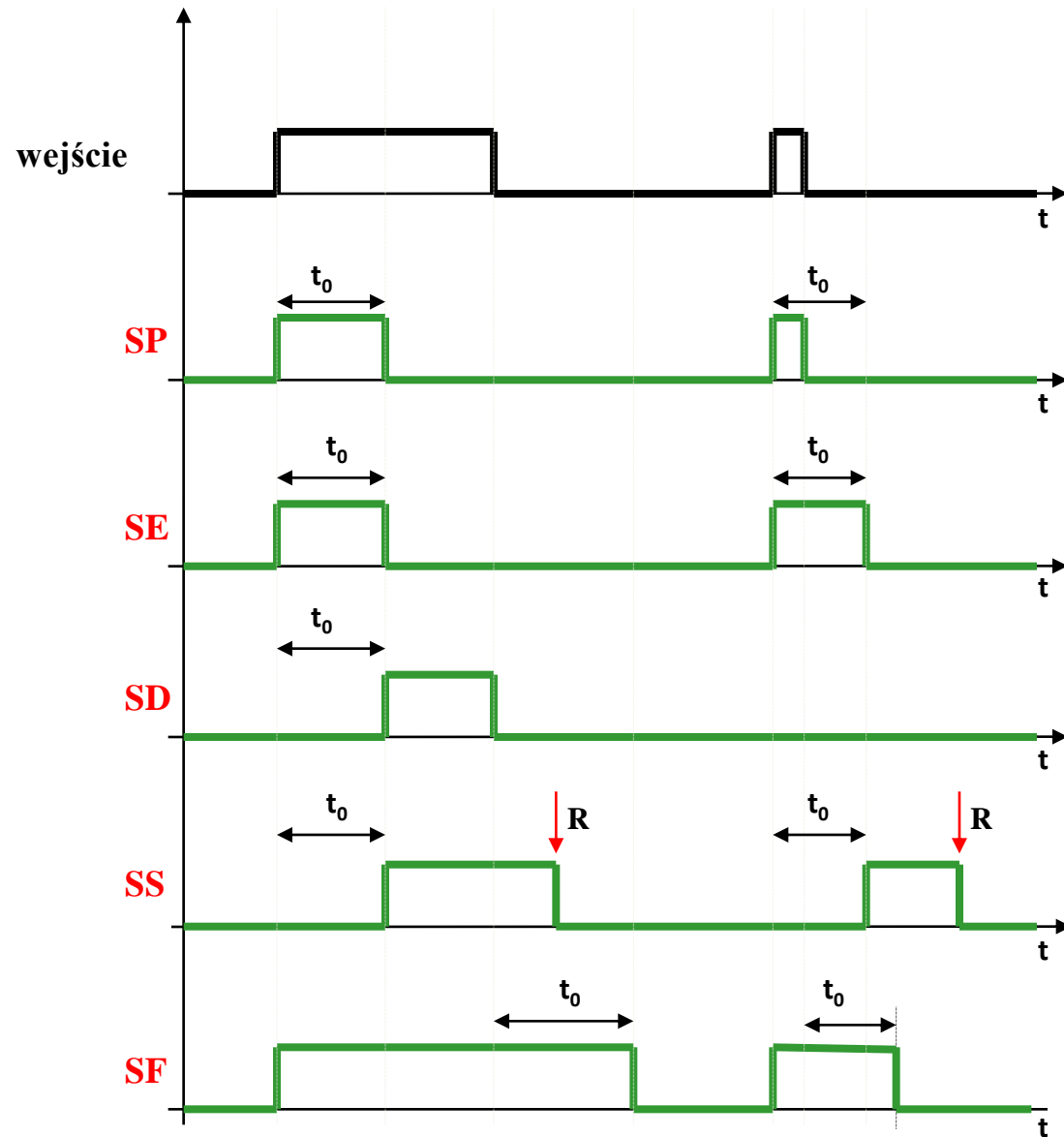


# Moduły czasowe (timer'y)

Działanie modułu czasowego odpowiada sposobowi działania przekaźnika czasowego z opóźnionym załączeniem lub wyłączeniem. Maksymalnie można zaprogramować 128 modułów czasowych oznaczonych instrukcją T0 do T127. W sterownikach Simatic możemy korzystać s 5-ciu różnie działających układów czasowych:

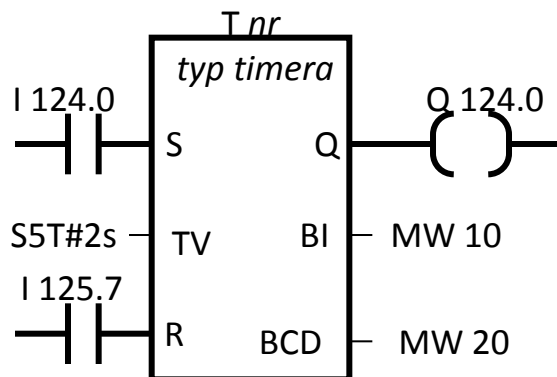
- SP** – **Pulse Timer**: Daje na wyjściu sygnał o określonej długości tylko przy aktywnym sygnale START.
- SE** – **Extended Pulse Timer**: Daje na wyjściu sygnał o określonej długości przy krótkiej aktywacji sygnału START.
- SD** – **On-Delay Timer**: Ustawienie timera jako timer z opóźnionym załączeniem.
- SS** – **Retentiv On-Delay Timer**: Uaktywnia się przez krótką aktywację sygnału START. Kasowanie jest możliwe tylko wejściem kasującym.
- SF** – **Off-Delay Timer**: Ustawienie timera jako timer z opóźnionym wyłączeniem.

# Wykresy czasowe



# Wykorzystanie timerów w LAD i STL

## LAD



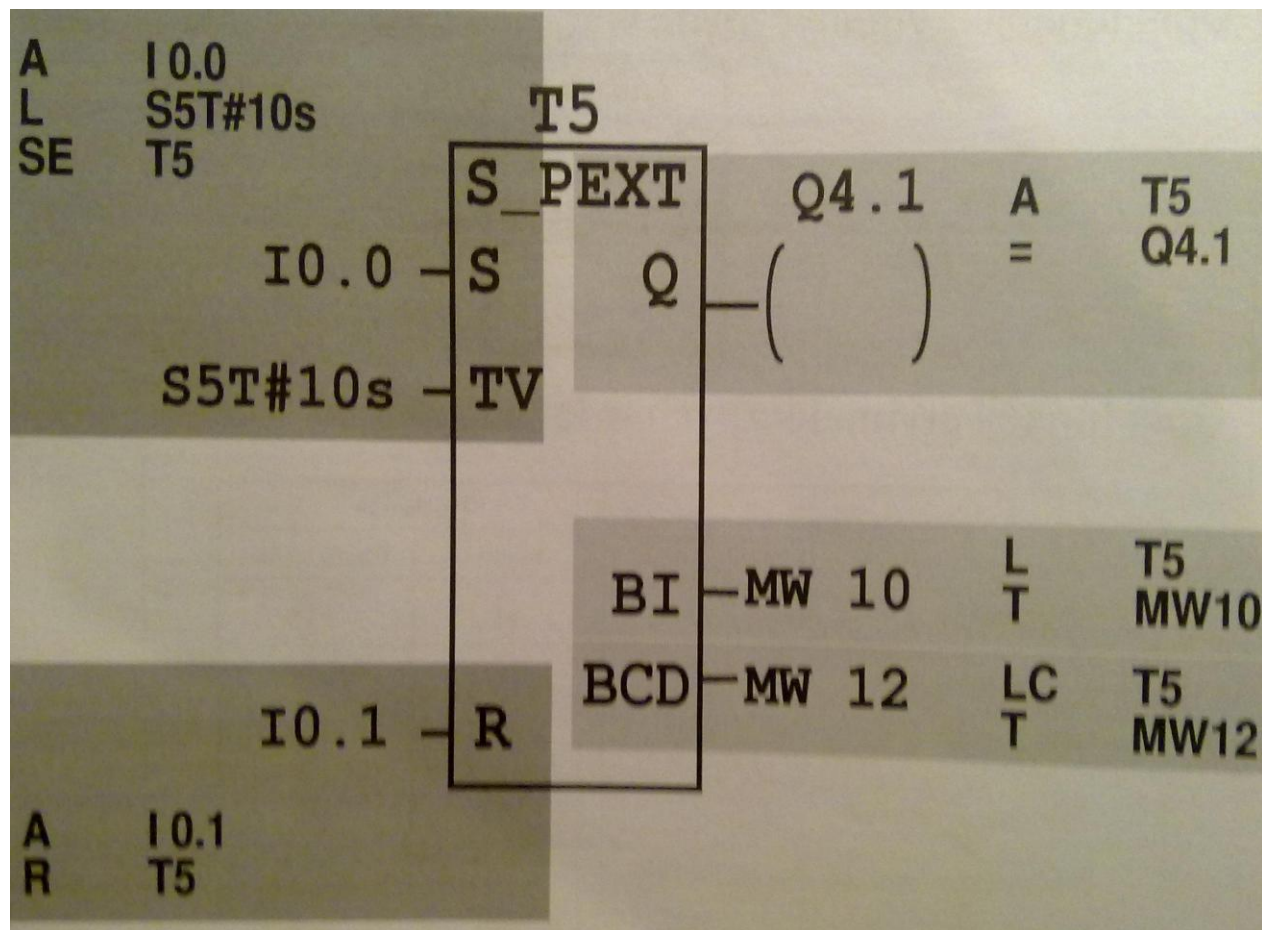
## STL

```

A I 124.0
L S5T#2s
SP T1
A I 125.7
R T1
L T1
T MW 10
LC T1
T MW 20
A T1
= Q124.0
    
```

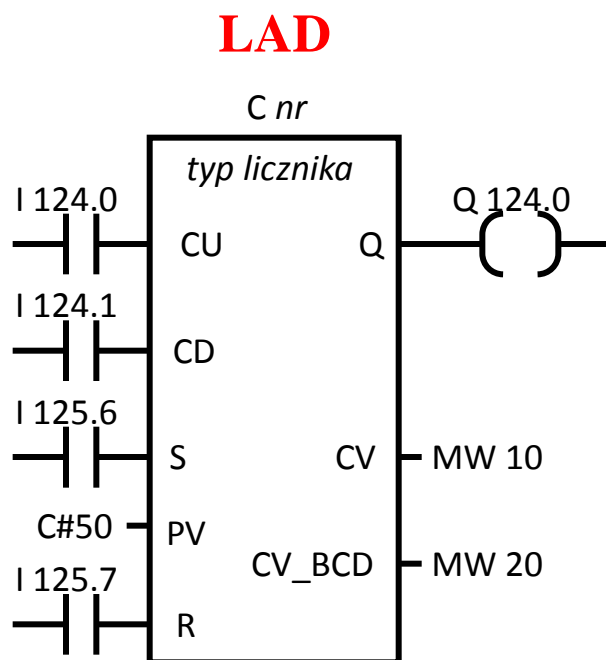
Typ timera	STEP 7
SP	S_PULSE
SE	S_PEXT
SD	S_ODT
SS	S_ODTS
SF	S_OFFDT

# Wykorzystanie timerów w LAD i STL



# Wykorzystanie liczników w LAD i STL

Licznik może zliczać sygnały zarówno w przód jak i do tyłu. Zakres liczenia zawiera się w przedziale od 0 do 999. Maksymalnie można zaprogramować 128 liczników dla sterowników serii S7-300



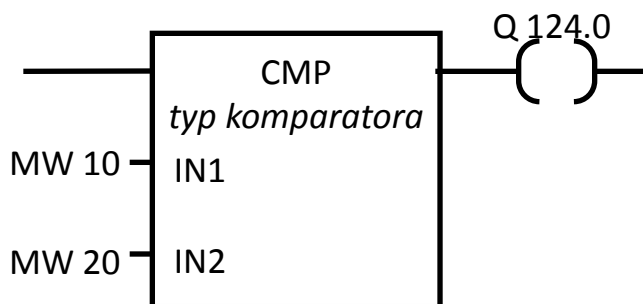
## STL

```
A I 124.0
CU C1
A I 124.1
CD C1
A I 125.6
L C# 50
S C1
A I 125.7
R C1
L C1
T MW10
LC C1
T MW20
A C1
= Q124.0
```

# Komparatory

Komparator służy do porównywania ze sobą dwóch wartości 16-bitowych lub 32-bitowych.

**LAD**



**STL**

```
L MW10
L MW20
==I
= Q124.0
```

typ komparatora	STEP 7		
	16-bit	32-bit	rzeczywiste
równy	==I	==D	==R
różny	<>I	<>D	<>R
większy	>I	>D	>R
mniejszy	<I	<D	<R
większy lub równy	>=I	>=D	>=R
mniejszy lub równy	<=I	<=D	<=R

# Bibliografia

- Kostro J., Elementy, urządzenia i układy automatyki, Wydawnictwa Szkolne i Pedagogiczne, Warszawa 1994
- Mikulczyński T., Samsonowicz Z., Automatykacja dyskretnych procesów produkcyjnych. Metody modelowania procesów dyskretnych i programowania PLC, WNT Warszawa 1997
- Legierski T., Kasprzak J., Hajda J., Wyrwał J., Programowanie sterowników PLC, Wyd. Prac. Komp. J. Skalmierskiego, Gliwice 1998
- Pakiet oprogramowania First ISaGRAF ver. 3.23f (c) 1990-1998 CJ International, udostępniony przez PEP Modular Computers Poland
- [www.intex.com.pl](http://www.intex.com.pl)