

Programowanie Sterowników Przemysłowych Przerwania

Paweł Strączyński

p.straczynski@tu.kielce.pl

Katedra Urządzeń Elektrycznych i Automatyki

Politechnika Świętokrzyska



Czy są przerwania w systemach automatyki?

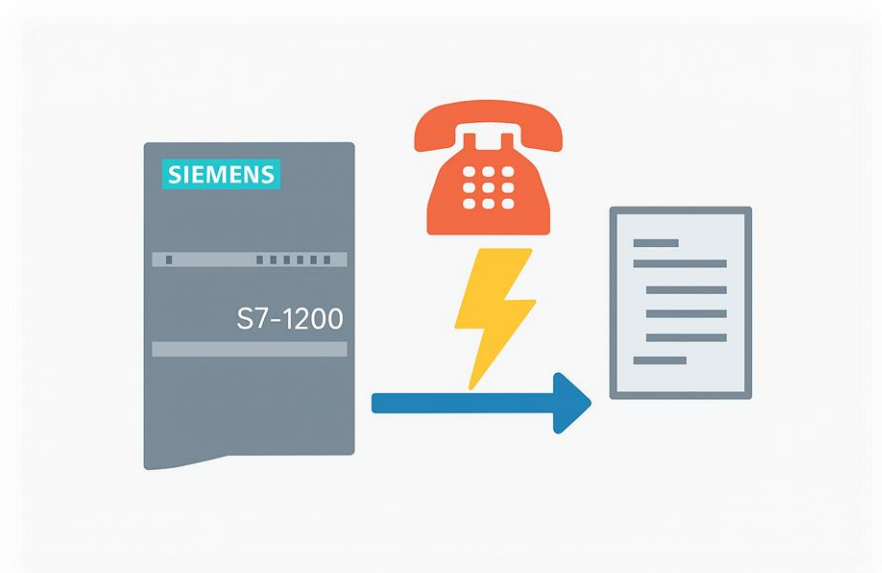
Przerwanie to specjalny mechanizm, który pozwala sterownikowi natychmiast zareagować na określone zdarzenie – niezależnie od tego, co aktualnie wykonuje główny program.

Dlaczego są potrzebne?

- ☐ Umożliwiają reakcję w czasie rzeczywistym (np. awaria, zmiana sygnału).
- ☐ Redukują opóźnienia w systemie sterowania.
- ☐ Ułatwiają organizację zadań cyklicznych i asynchronicznych.

Zatrzymanie maszyny przy wciśnięciu przycisku STOP.

- ☐ Reakcja na sygnał z enkodera lub czujnika położenia.
- ☐ Cykliczne zbieranie danych z czujników co 100 ms.
- ☐ Reakcja na impuls zewnętrzny z drugiego urządzenia.



Zastosowanie przerwań w systemach sterowania – typowe scenariusze.

⚡ Awaryjne zatrzymanie maszyny

- ❑ Reakcja na sygnał z przycisku STOP / czujnika bezpieczeństwa.
- ❑ Natychmiastowe przerwanie działania niezależnie od stanu programu głównego.

🕒 Precyzyjne zadania czasowe

- ❑ Cykliczne zbieranie danych z czujników co np. 50 ms.
- ❑ Niezależne wykonywanie obliczeń w stałym odstępie czasu.

🔄 Obsługa szybkich sygnałów wejściowych

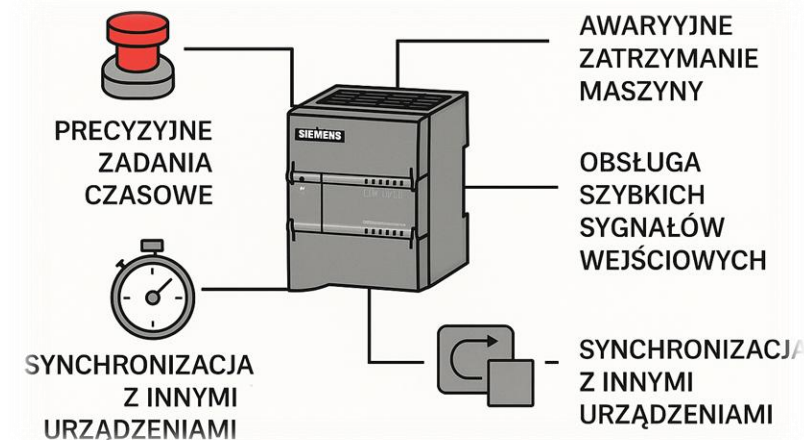
- ❑ Reakcja na impuls z enkodera (np. zliczanie pozycji).
- ❑ Rejestracja bardzo krótkich impulsów, które mogłyby zostać pominięte w programie głównym.

📦 Reakcja na zdarzenia zewnętrzne

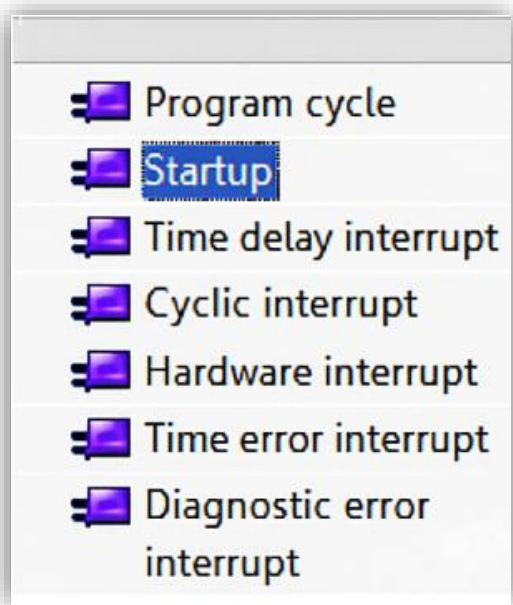
- ❑ Np. pojawienie się sygnału od czujnika położenia, gotowości siłownika, etc.
- ❑ Umożliwia reakcję dokładnie w momencie wystąpienia zdarzenia.

🔄 Synchronizacja z innymi urządzeniami

- ❑ Odpowiedź na sygnał z urządzenia nadrzędnego lub innego sterownika.
- ❑ Zmniejszenie opóźnień w komunikacji między systemami.



Typy przerwań w S7-1200 (TIA Portal)



- ◆ **Program cycle (OB1)**

Główny cykliczny program użytkownika – nie jest przerwaniem, ale odniesieniem do działania pozostałych bloków.

- ◆ **Startup (OB100)**

Wykonywany raz po włączeniu CPU lub po wgraniu programu – np. do inicjalizacji zmiennych.

- ◆ **Time delay interrupt (OB20...29)**

Przerwanie wywoływane po określonym opóźnieniu – jednozdaniowe opóźnione akcje.

- ◆ **Cyclic interrupt (OB30...38)**

Najczęściej używane przerwania czasowe – uruchamiane cyklicznie co określony interwał czasu.

- ◆ **Hardware interrupt (OB40...49)**

Reakcja na sygnały wejściowe z fizycznych wejść (np. zbocze narastające na I0.0) – szybkie reagowanie na zdarzenia.

- ◆ **Time error interrupt (OB80)**

Przerwanie uruchamiane, gdy CPU nie zdąży wykonać cyklu w zadanym czasie – do diagnostyki.

- ◆ **Diagnostic error interrupt (OB82)**

Wywoływane w przypadku błędów sprzętowych (np. brak modułu, zanik komunikacji itp.).

Przerwanie startowe (Startup)

Czym jest Startup OB100?

Blok organizacyjny uruchamiany jednorazowo:

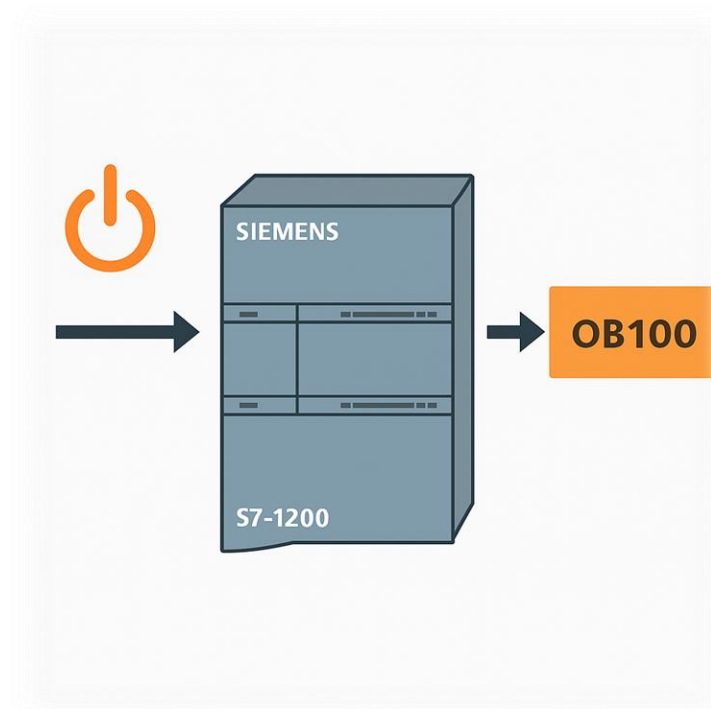
- ☐ po załączeniu zasilania CPU,
- ☐ po przejściu CPU z trybu STOP do RUN,
- ☐ po wgraniu programu do CPU.

Do czego służy?

- ☐ Inicjalizacja danych i zmiennych (np. ustawienie wartości początkowych).
- ☐ Wykonanie konfiguracji startowej (np. ustawienie stanów wyjść, reset liczników.)
- ☐ Wywołanie komunikatów diagnostycznych lub alarmów po starcie.

Ważne uwagi:

- ☐ OB100 wykonuje się tylko raz – nie powtarza się cyklicznie.
- ☐ Nie działa, jeśli CPU jest resetowane programowo (w niektórych przypadkach).
- ☐ Warto go używać zamiast kodu startowego w OB1 – jest przejrzystszy i bezpieczniejszy.



Time Delay Interrupt

Time Delay Interrupt (OB20...29)?

- ☐ Przerwanie wywoływane raz, po upływie określonego czasu od momentu aktywacji.
- ☐ Działa jak opóźniony timer – CPU wykonuje blok OB dokładnie po zadanym czasie.

Do czego służy?

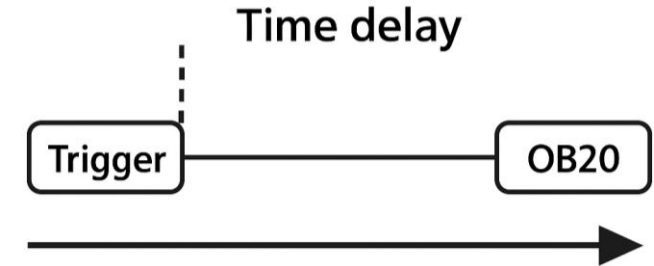
- ☐ Opóźnione działania po wystąpieniu określonego zdarzenia (np. uruchomienie napędu po 5 sekundach).
- ☐ Sekwencyjne sterowanie – np. kolejne etapy uruchamiania maszyny.
- ☐ Możliwość „odroczenia” kodu bez użycia timerów w głównym programie.

Cechy charakterystyczne:

- ☐ Czas opóźnienia można ustawić z dokładnością do milisekund.
- ☐ OB20–OB29 umożliwiają jednoczesne definiowanie wielu różnych opóźnień.
- ☐ Przerwanie jest jednorazowe – po wywołaniu nie powtarza się automatycznie.

Uwaga:

Przerwania tego typu nie są tak często używane jak przerwy cykliczne, ale bywają bardzo przydatne w aplikacjach sekwencyjnych i procesowych.



Przerwania cykliczne (Cyclic Interrupts)

Czym są?

- ☐ Przerwania wywoływane automatycznie przez CPU w określonych odstępach czasu.
- ☐ Wykorzystywane do cyklicznego wykonywania zadań niezależnych od głównego programu.

Jak działają?

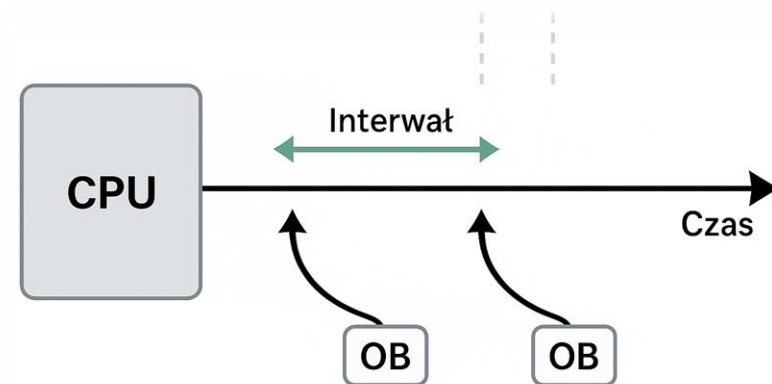
- ☐ CPU uruchamia specjalny blok OB (np. OB30, OB31...) w zaprogramowanym interwale (np. co 100 ms).
- ☐ Po zakończeniu obsługi, sterownik wraca do programu głównego.

Typowe zastosowania:

- ☐ Cykliczny odczyt wartości analogowych.
- ☐ Monitorowanie stanu czujników.
- ☐ Aktualizacja liczników, liczników czasu, zbieranie danych.

Ważne informacje:

- ☐ Można zdefiniować kilka przerw czasowych w jednym programie.
- ☐ Można ustawić czas rozpoczęcia, częstotliwość, priorytet.
- ☐ Nadmierna liczba zbyt częstych przerw może przeciążyć CPU.



Przerwania cykliczne (Cyclic Interrupts)

Przykład: Konfiguracja przerwania cyklicznego w TIA Portal



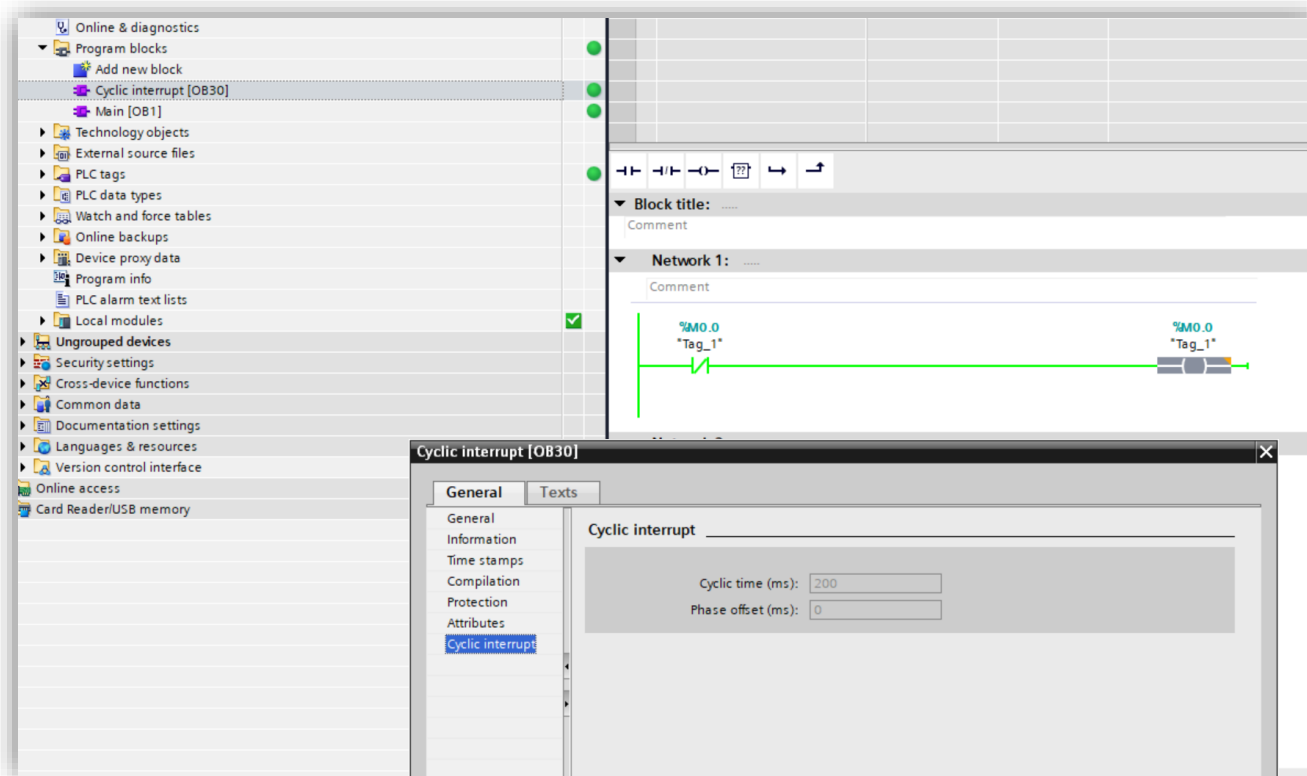
Cel:

Wykonanie przerwania co 200 ms, które przełącza bit.



Kroki konfiguracji:

1. Dodanie OB (np. OB30):
 - Prawym przyciskiem myszy → "Add new block" → "Organization block" → wybierz typ „Cyclic interrupt”.
2. Ustawienie parametrów przerwania:
 - W „Cyclic time” przypisz czas: 200 ms.
3. Programowanie w OB30:
 - Wstaw blok przełączający bit ($M0.0 = \text{NOT } M0.0$)



Hardware Interrupt

Czym są?

- ☐ Przerwanie wywoływane po wystąpieniu zdarzenia sprzętowego – np. zmiany stanu na wejściu cyfrowym (zbocze narastające lub opadające).
- ☐ Wykorzystywane do błyskawicznej reakcji na fizyczne sygnały z czujników, przycisków, enkoderów itp.

Zastosowania:

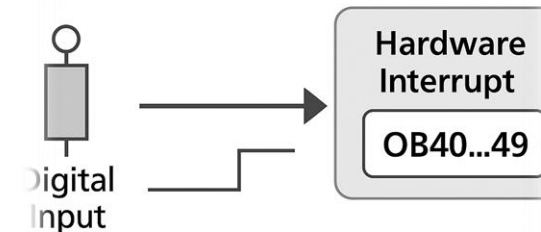
- ☐ Reakcja na przycisk STOP lub awaryjny.
- ☐ Zliczanie impulsów z enkodera lub czujnika zbliżeniowego.
- ☐ Wykrycie dokładnego momentu zadziałania detektora.

Cechy charakterystyczne:

- ☐ Wymaga przypisania sprzętowego wejścia do przerwania w konfiguracji CPU.
- ☐ Można ustawić, czy przerwanie ma reagować na zbocze narastające, opadające czy oba.
- ☐ Obsługiwane w blokach OB40–OB49 – dla różnych kanałów.

Uwaga:

- ☐ Hardware interrupts mają wysoki priorytet, dlatego nie powinny zawierać złożonej logiki – tylko szybkie reakcje (ustawienie flagi, liczniki itd.)..



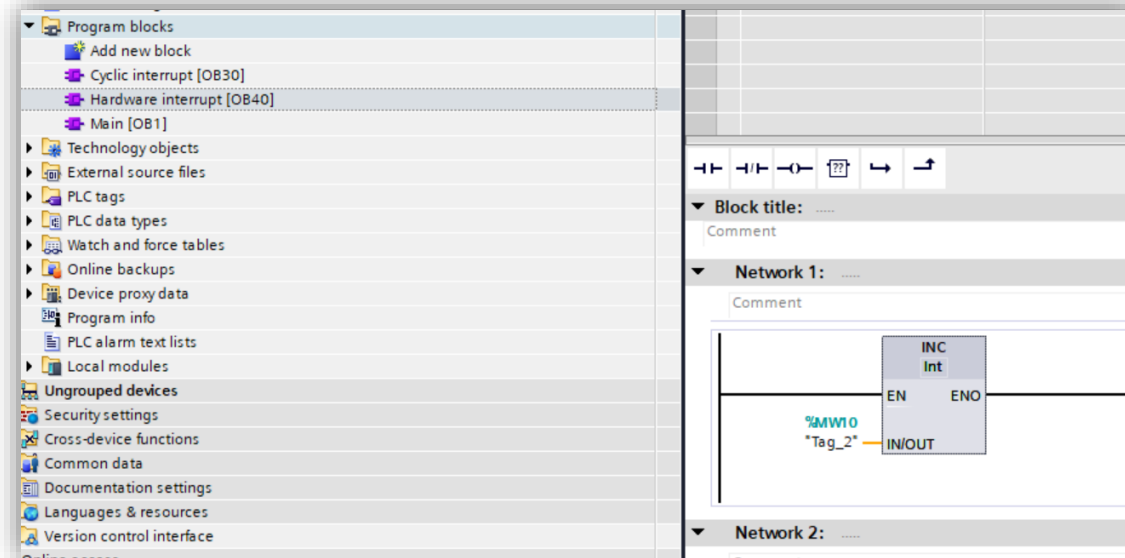
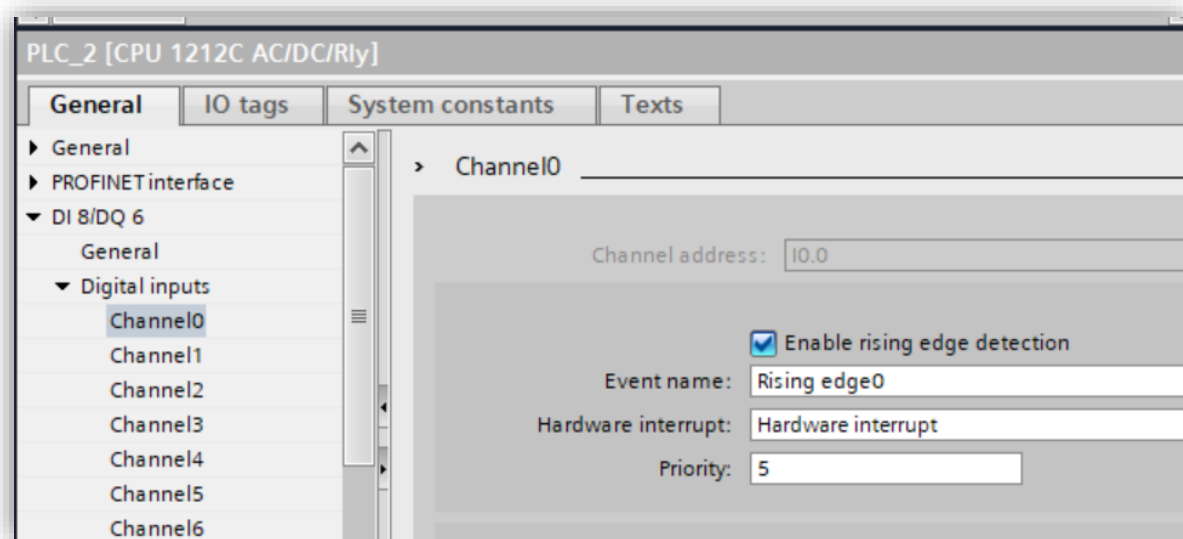
Hardware Interrupt

Przykład: Konfiguracja przerwania sprzętowego w TIA Portal



Cel:

Zliczanie impulsów z czujnika (np. fotokomórki) podłączonego do wejścia I0.0 – każde zbocze narastające wyzwała przerwanie, które zwiększa licznik.



Przerwania krytyczne i diagnostyczne

⚠ Time Error Interrupt (OB80)

Wywoływane automatycznie, gdy CPU nie zdąży wykonać cyklu programu w przewidzianym czasie (przekroczenie "watchdog time").

- ❑ Może sygnalizować nadmierne obciążenie CPU lub źle zaprojektowaną logikę.
- ❑ OB80 pozwala zapisać informacje diagnostyczne, np. do rejestru lub wysłać alarm.

🧠 Zastosowania:

- ❑ Uruchamiane przy wykryciu błędu sprzętowego:
 - ❑ brak modułu,
 - ❑ uszkodzenie wejścia/wyjścia,
 - ❑ zanik komunikacji z rozszerzeniem.
- ❑ Można wykorzystać do rejestracji błędów, zatrzymania maszyny lub wysyłania alarmu HMI.

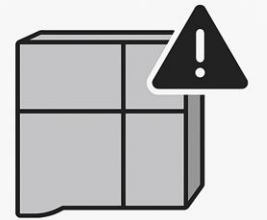
💡 Dobre praktyki:

- ❑ OB80 i OB82 powinny zawierać tylko lekką logikę (np. ustawienie flagi).
- ❑ Warto je włączać w projekcie nawet prewencyjnie – dla lepszej diagnostyki i niezawodności.



OB80

**Time Error
Interrupt**



OB82

**Diagnostic
Error Interrupt**