

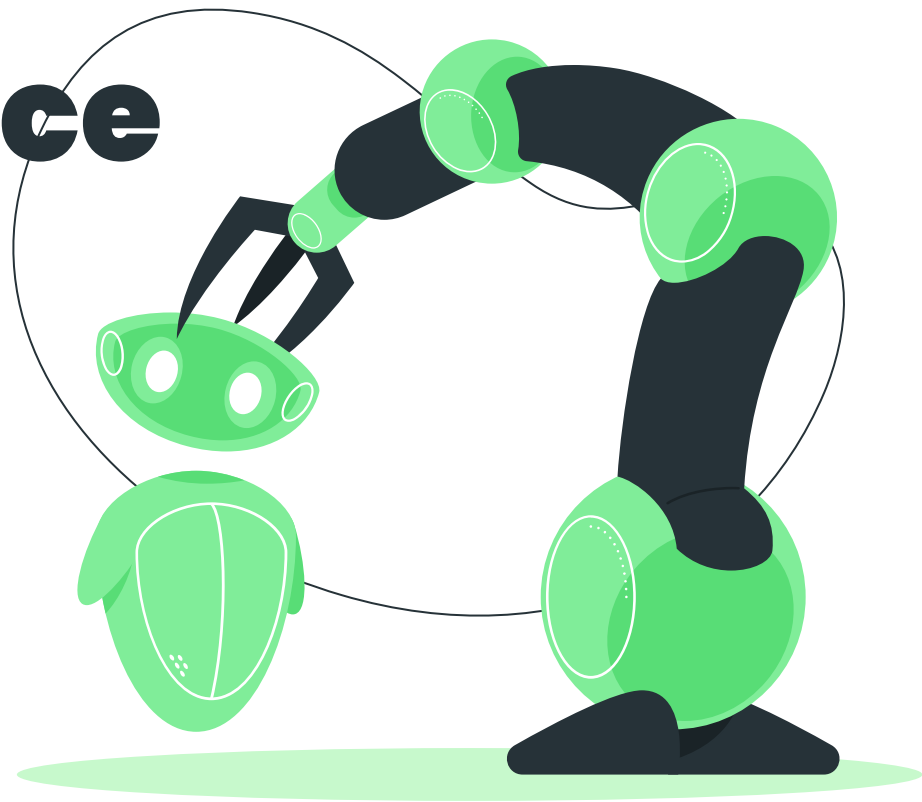
Warsztaty wprowadzające

Do II Ogólnopolskiego Konkursu
Robotyki Przemysłowej

ETAP 2

Sterowniki PLC

mgr inż. Sylwester Paterek
CKPiDN Mielec





1
Czym są
sterowniki PLC
?

Wprowadzenie

Sterowniki PLC, czyli **Programmable Logic Controllers**, to programowalne urządzenia wykorzystywane w automatyce przemysłowej. Ich głównym zadaniem jest sterowanie procesami produkcyjnymi oraz monitorowanie działania urządzeń. PLC pełni rolę „mózgu” systemu automatyki, podejmując decyzje na podstawie informacji zebranych z czujników i realizując zadania, takie jak włączanie lub wyłączanie urządzeń



Definicja i rola sterowników PLC w automatyce przemysłowej

Sterownik PLC to elektroniczne urządzenie cyfrowe, które działa w oparciu o wgrany program logiczny. Program ten steruje urządzeniami wykonawczymi, takimi jak silniki, zawory, czy systemy transportowe, według ustalonej logiki.

PLC umożliwia:

- **Automatyzację** procesów przemysłowych, co zwiększa wydajność i bezpieczeństwo pracy.
- **Elastyczność** – zmiana procesu nie wymaga zmiany fizycznych elementów sterujących, wystarczy edycja programu.
- **Niezawodność** – sterowniki są projektowane tak, aby działały bez przerw w wymagających środowiskach przemysłowych.

Historia

Historia sterowników programowalnych zaczyna się w 1968, gdy w grupie General Motors podjęto prace nad nową generacją sterowników po to, aby zastąpić układy sterowania stycznikowoprzełącznikowe.

Od tego czasu sterowniki PLC przeszły ogromny rozwój:

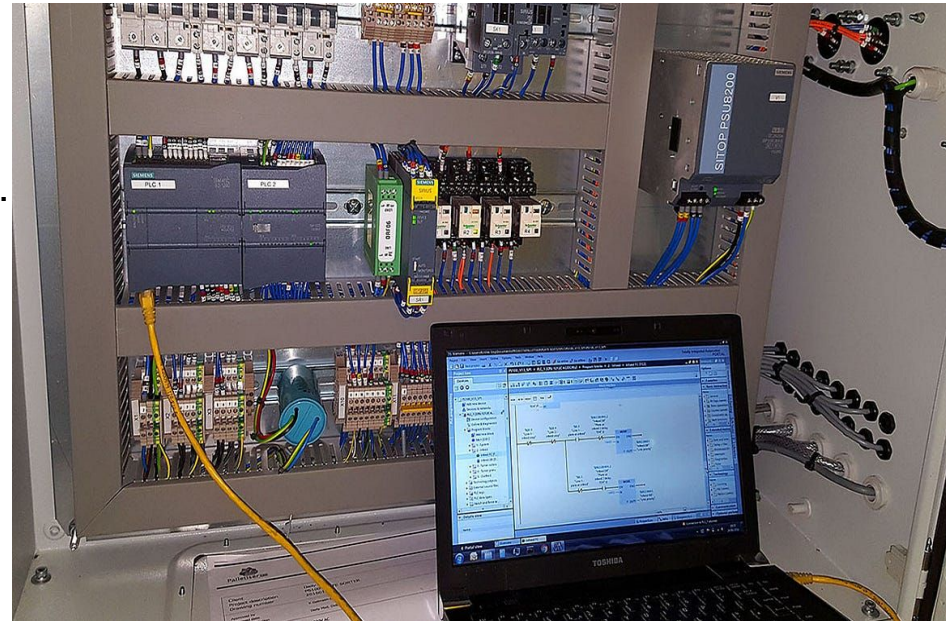
1. **Lata 70.** – wprowadzenie podstawowych języków programowania, takich jak LAD (ladder diagram).
2. **Lata 80. i 90.** – integracja z systemami komputerowymi i rozwój standardu IEC 61131-3.
3. **Współczesność** – sterowniki PLC są zintegrowane z Internetem Rzeczy (IoT) i systemami Przemysłu 4.0, oferując zaawansowane możliwości komunikacyjne oraz obsługę sztucznej inteligencji.



Założenia

Przyjęto następujące założenia dla sterowników PLC:

1. Łatwość programowania i przeprogramowywania stosowne do zmieniających się warunków przemysłowych.
2. Łatwość utrzymania w ruchu produkcyjnym z możliwością napraw przez wymianę instalowanych modułów.
3. Większa niezawodność w warunkach przemysłu wych przy mniejszych gabarytach.
4. Koszty porównywalne ze stosowanymi panelami przekaźnikowymi i szafami sterowniczymi.

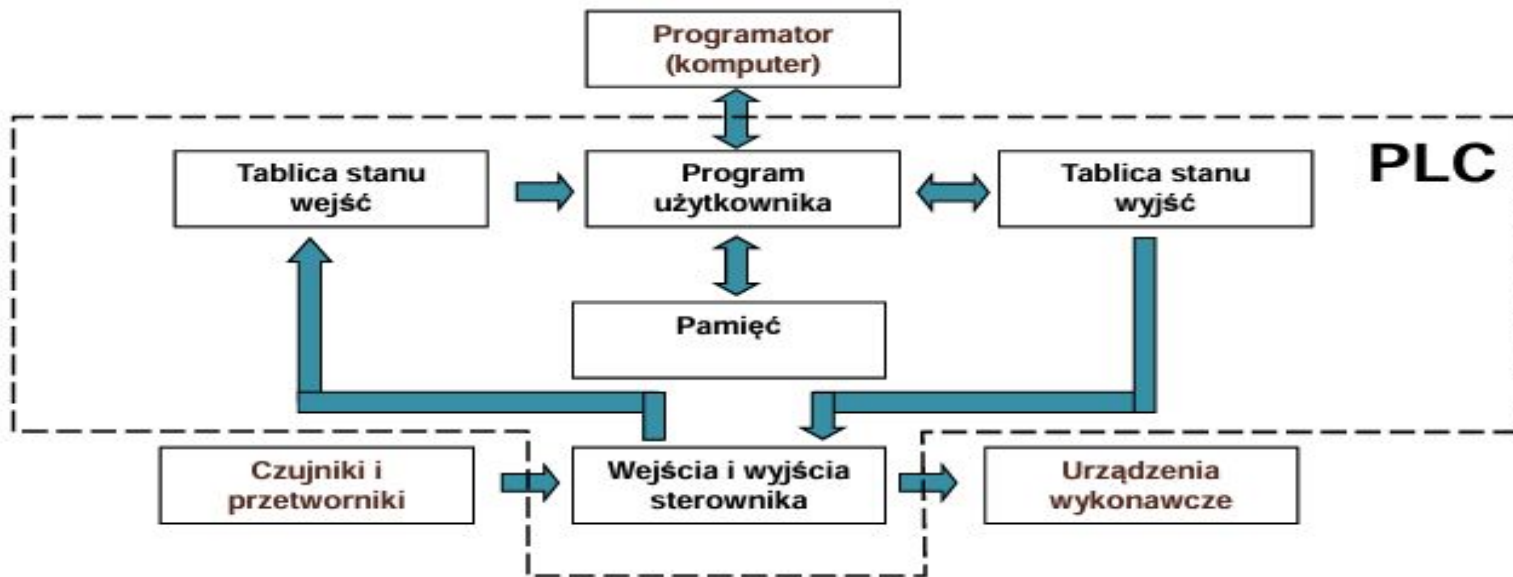


Założenia

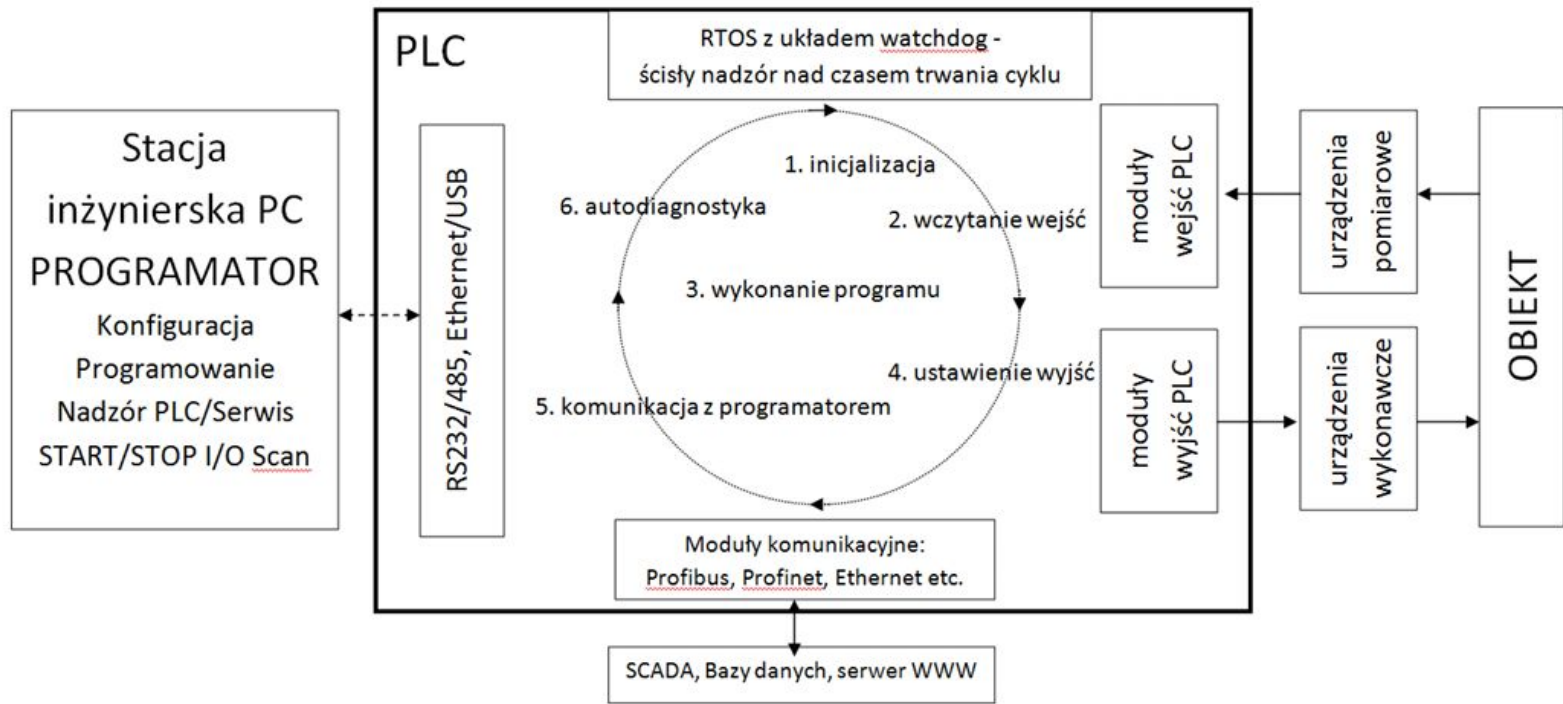
PLC (ang. Programmable Logic Controller) Sterowniki programowalne PLC są komputerami przemysłowymi, które pod kontrolą systemu operacyjnego czasu rzeczywistego:

- zbierają pomiary za pośrednictwem modułów wejściowych z analogowych i dyskretnych czujników oraz urządzeń pomiarowych,
- transmitują dane za pomocą modułów i łącz komunikacyjnych,
- wykonują programy aplikacyjne na podstawie przyjętych parametrów i uzyskanych danych o sterowanym procesie lub maszynie,
- generują sygnały sterujące zgodnie z wynikami obliczeń tych programów i przekazują je poprzez moduły wyjściowe do elementów i urządzeń wykonawczych,
- realizują funkcje diagnostyki programowej i sprzętowej

Schemat działania



Schemat działania



Miejsce pracy i reżim PLC

Typowym środowiskiem pracy sterownika PLC jest **szafa sterownicza**.

Sterowniki PLC są zaprojektowane do działania w trybie ciągłym: **24 godziny na dobę, 365 dni w roku**.

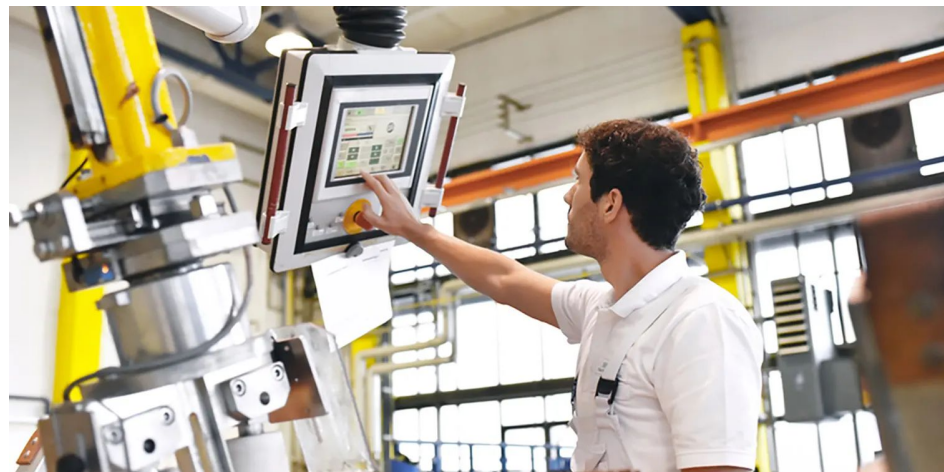
Wiele instalacji potwierdza niezawodność PLC – mogą pracować **nieprzerwanie przez kilkanaście lat**.



Gdzie wykorzystujemy PLC?

Kluczowe zastosowania w różnych sektorach

1. **Przemysł motoryzacyjny:**
 - Sterowanie robotami przemysłowymi w liniach montażowych.
 - Automatyzacja procesów malowania i spawania.
2. **Przemysł spożywczy:**
 - Kontrola systemów pakowania i etykietowania.
 - Monitorowanie procesów takich jak pasteryzacja czy mieszanie składników.
3. **Przemysł chemiczny:**
 - Zarządzanie skomplikowanymi procesami chemicznymi, np. w reaktorach.
 - Monitorowanie parametrów, takich jak temperatura czy ciśnienie.
4. **Budownictwo i infrastruktura:**
 - Sterowanie systemami HVAC w dużych budynkach.
 - Automatyzacja systemów oświetleniowych i bezpieczeństwa.



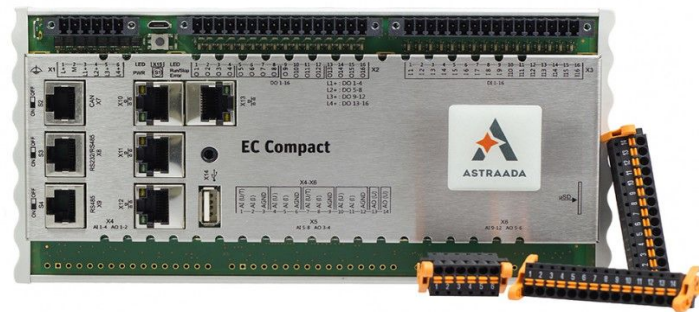


2 Rodzaje i
przykłady
sterowników
PLC

Podział sterowników ze względu na zastosowanie

Sterowniki kompaktowe

- Sterowniki kompaktowe to urządzenia o zintegrowanej budowie, gdzie moduły wejść/wyjść (I/O) są wbudowane w samą jednostkę centralną (CPU).
- **Zalety:**
 - Łatwe w instalacji i konfiguracji.
 - Kompaktowe rozmiary, co sprawia, że są idealne do małych aplikacji.
- **Zastosowanie:**
 - Maszyny jednozadaniowe, linie pakujące, prostsze procesy przemysłowe.



Podział sterowników ze względu na zastosowanie

Sterowniki modułowe

- Sterowniki modułowe składają się z jednostki centralnej (CPU) oraz oddzielnych modułów I/O, które można dobierać w zależności od wymagań aplikacji.
- **Zalety:**
 - Wysoka elastyczność i możliwość rozbudowy.
 - Możliwość dostosowania do skomplikowanych procesów.
- **Zastosowanie:**
 - Linie produkcyjne, złożone systemy sterowania w przemyśle chemicznym i motoryzacyjnym.



Podział sterowników ze względu na zastosowanie

Sterowniki modułowe vs kompaktowe

- **Sterowniki kompaktowe** to zintegrowane urządzenia, idealne do prostych aplikacji automatyki. Są ekonomiczne, łatwe w instalacji i konfiguracji, lecz mają ograniczone możliwości rozbudowy.
- **Sterowniki modułowe** oferują elastyczność i skalowalność dzięki możliwości dodawania modułów IO i komunikacyjnych. Sprawdzają się w zaawansowanych systemach, takich jak linie produkcyjne, robotyka czy sterowanie ruchem.
- **Wybór sterownika** zależy od wymagań aplikacji – kompaktowe dla prostych systemów, modułowe dla skomplikowanych i rozwijających się projektów.

Podział sterowników ze względu na zastosowanie

Mikrosterowniki

- To małe i ekonomiczne sterowniki PLC, które charakteryzują się ograniczonymi możliwościami I/O i podstawową funkcjonalnością.
- **Zalety:**
 - Niski koszt.
 - Prosta obsługa.
- **Zastosowanie:**
 - Automatyka domowa, sterowanie małymi urządzeniami, systemy HVAC.



Przykłady popularnych sterowników

Siemens SIMATIC S7

- Jedna z najbardziej rozpoznawalnych serii PLC na rynku, oferująca modele od podstawowych (S7-1200) po zaawansowane (S7-1500).
- **Zastosowanie:** Przemysł spożywczy, motoryzacyjny, chemiczny.
- **Siemens SIMATIC S7-1200** to modułowy i kompaktowy sterownik, idealny do precyzyjnych zadań automatyzacji. Oferuje różne klasy wydajności CPU (1211C, 1212C, 1214C, 1215C, 1217C, oraz wersje failsafe 1214FC i 1215FC) z możliwością rozbudowy o moduły komunikacyjne i sygnałowe



Przykłady popularnych sterowników

Allen-Bradley ControlLogix

- Zaawansowany sterownik modułowy od Rockwell Automation, ceniony za wysoką wydajność i niezawodność, szczególnie w Ameryce Północ
- **Cechy:** Obsługa sterowania ruchem, zaawansowane funkcje logiki i czasu rzeczywistego, integracja z SCADA i protokoła (EtherNet/IP, ControlNet).
- **Zastosowania:** Linie produkcyjne, systemy SCADA, robotyka, przemysł energetyczny i chemiczny.
- Dzięki elastyczności i wsparciu zaawansowany technologii jest idealny do złożonych systemów automatyzacji.



Przykłady popularnych sterowników

Mitsubishi MELSEC

- Wszechstronna seria sterowników PLC od Mitsubishi Electric, obejmująca zarówno mikrosterowniki, jak i zaawansowane systemy modułowe.
- **Cechy:** Szeroki zakres funkcjonalności, możliwość rozbudowy, obsługa zaawansowanych protokołów komunikacyjnych (EtherNet/IP, CC-Link).
- **Zastosowania:** Automatyka budynkowa, przemysł maszynowy, linie produkcyjne i sterowanie ruchem.
- Dzięki elastyczności i niezawodności MELSEC jest idealny do aplikacji o różnej skali i złożoności.
- **MELSEC Q Series:** Modułowy, zaawansowany do dużych systemów produkcyjnych.
- **MELSEC iQ-R:** Szybki, z funkcjami bezpieczeństwa i integracją z robotami.
- **MELSEC iQ-F:** Rozszerzone mikrosterowniki z zaawansowaną technologią.
- **MELSEC L Series:** Kompaktowy, modułowy z możliwością rozbudowy.



Przykłady popularnych sterowników

Schneider Electric Modicon

Pierwszy na świecie sterownik PLC, stale modernizowany, dostosowany do różnych aplikacji przemysłowych.

Cechy: Modułowa budowa, zaawansowane protokoły komunikacyjne (Ethernet, Modbus), wysoka niezawodność.

Przykłady modeli:

- **M221** – kompaktowy, idealny dla małych maszyn.
- **M241** – z obsługą CANopen, do sterowania urządzeniami.
- **M580** – PAC z redundancją, do procesów krytycznych.

Zastosowania: Zarządzanie energią, systemy wodno-kanalizacyjne, przemysł chemiczny.



Przykłady popularnych sterowników

Omron SYSMAC

Zaawansowana seria sterowników PLC z szybkim przetwarzaniem danych, dedykowana do aplikacji wymagających precyzji i niezawodności.

Cechy: Wysoka wydajność, integracja sterowania ruchem, logiki i wizji maszynowej, obsługa protokołów EtherCAT i Ethernet/IP.

Przykłady modeli:

- **NJ501** – do złożonych systemów sterowania ruchem i robotyki.
- **NX1P2** – kompaktowy, idealny do precyzyjnych aplikacji montażowych.
- **NJ101** – ekonomiczny, do mniejszych systemów automatyzacji.

Zastosowania: Pakowanie, montaż precyzyjny, przemysł farmaceutyczny.



Przykłady popularnych sterowników

Unitronics UniStream

Mikrosterownik PLC z wbudowanym interfejsem HMI, idealny do małych maszyn.

Cechy: Kompaktowa konstrukcja, integracja sterowania i wizualizacji, programowanie w UniLogic, obsługa Ethernet i Modbus.

Przykłady modeli:

- **USP-070-B10** – mikrosterownik z panelem HMI, idealny do sterowania i wizualizacji.
Zastosowania: Małe maszyny z wymaganiami wizualizacji i sterowania, przemysł spożywczy, pakowanie.



Przykłady popularnych sterowników

Horner APG XL7 Prime

Sterownik PLC zintegrowany z panelem HMI o przekątnej 7", oferujący zaawansowane funkcje sterowania i wizualizacji.

Cechy: Kolorowy, dotykowy wyświetlacz TFT, wbudowane I/O, porty komunikacyjne (RS232, RS485, Ethernet, USB).

Zastosowania: Automatykacja maszyn, systemy sterowania z wizualizacją, przemysł spożywczy i pakowanie.





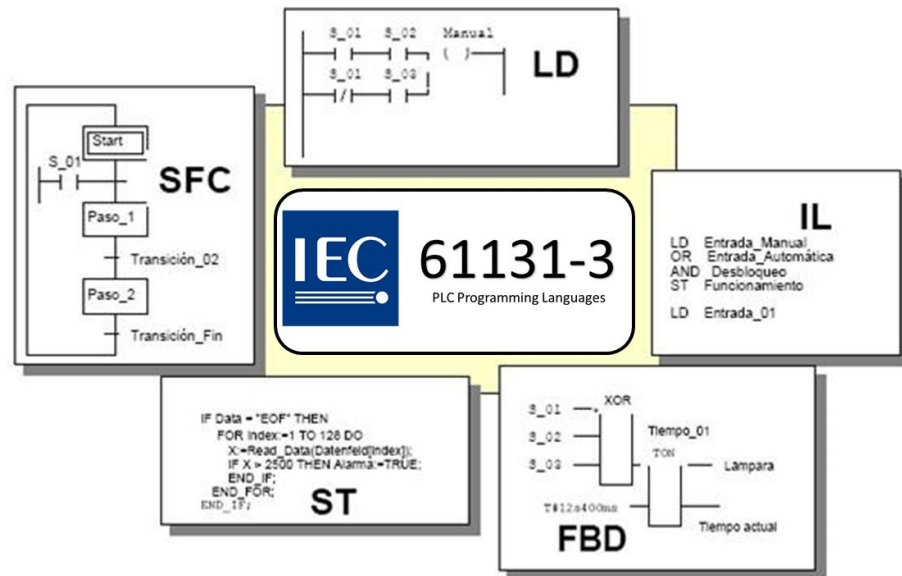
3

**Standardy
programowania
sterowników
PLC**

Norma IEC 61131-3

Opis normy i jej znaczenie:

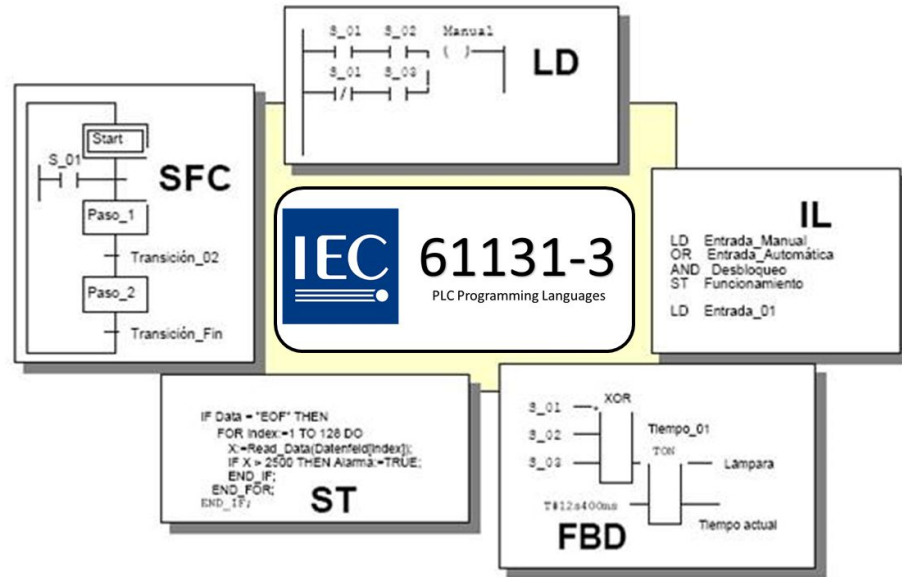
- IEC 61131-3 to międzynarodowy standard, który definiuje języki programowania stosowane w sterownikach PLC. Został stworzony przez **Międzynarodową Komisję Elektrotechniczną (IEC)**, aby ujednolicić sposób programowania sterowników.
- Norma ta umożliwia inżynierom korzystanie z jednolitych standardów, niezależnie od producenta sterownika, co zwiększa przenośność i uniwersalność programów.



Norma IEC 61131-3

Korzyści wynikające ze standardu:

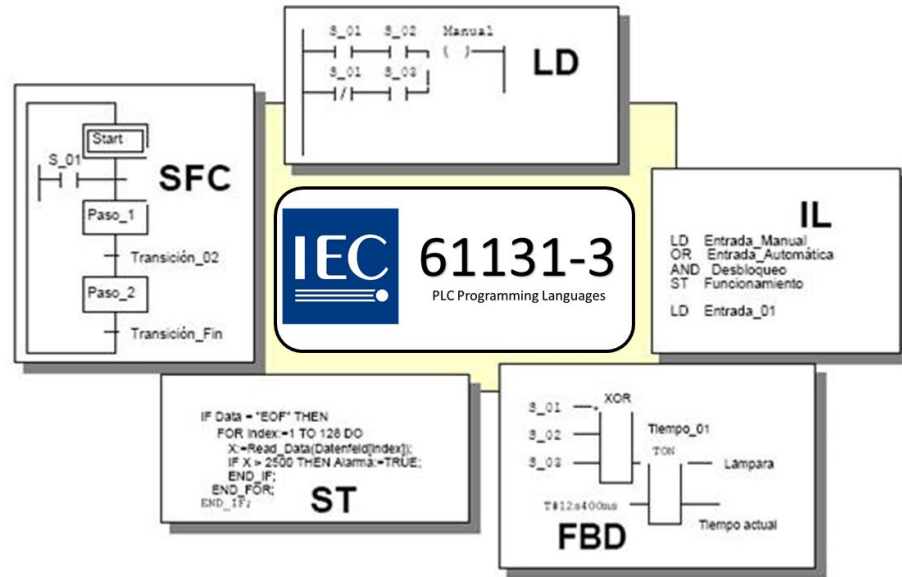
- **Standaryzacja:** Ujednolica języki programowania, co ułatwia naukę i pracę na różnych platformach PLC.
- **Uniwersalność:** Programy mogą być przenoszone między sterownikami różnych producentów, minimalizując czas i koszty integracji.
- **Zwiększenie efektywności:** Dzięki uporządkowanej strukturze programowania, inżynierowie mogą szybciej wdrażać rozwiązania.



Norma IEC 61131-3

Korzyści wynikające ze standardu:

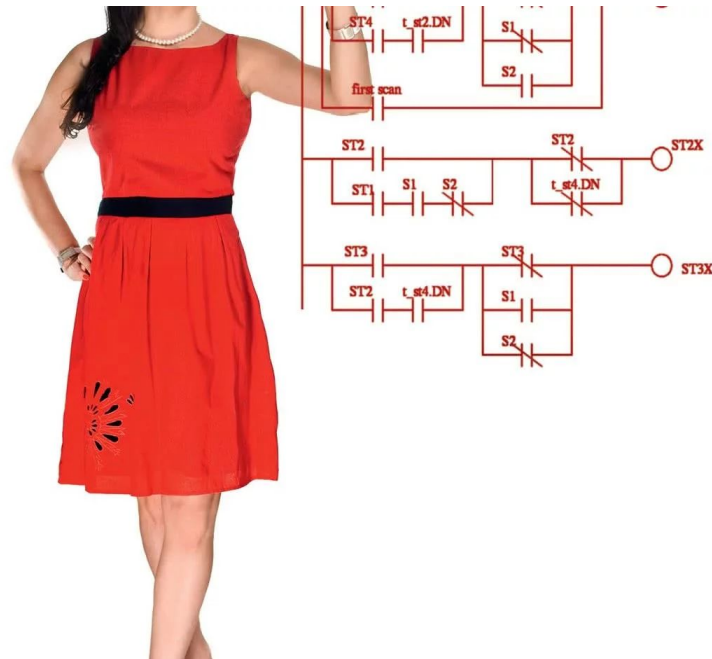
- **Standaryzacja:** Ujednolica języki programowania, co ułatwia naukę i pracę na różnych platformach PLC.
- **Uniwersalność:** Programy mogą być przenoszone między sterownikami różnych producentów, minimalizując czas i koszty integracji.
- **Zwiększenie efektywności:** Dzięki uporządkowanej strukturze programowania, inżynierowie mogą szybciej wdrażać rozwiązania.



Norma IEC 61131-3

Znaczenie normy w przemyśle:

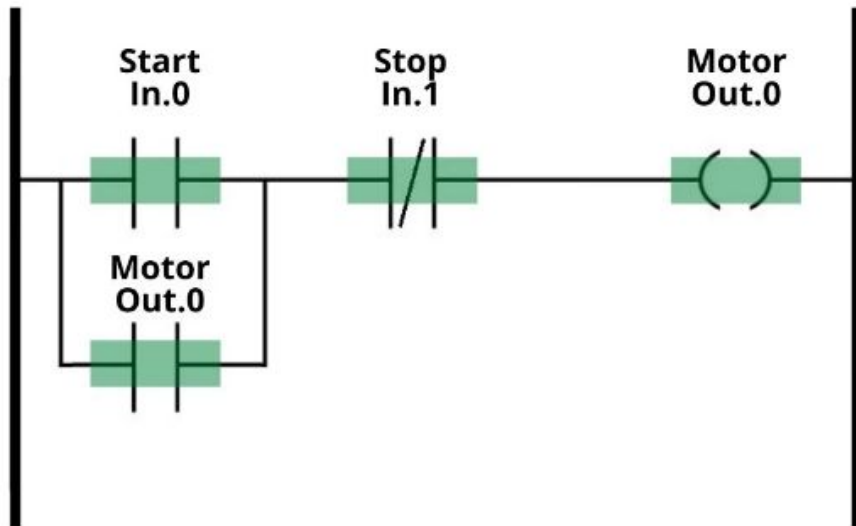
- Wspiera **integrację systemów automatyki** w różnych gałęziach przemysłu.
- Umożliwia współpracę między zespołami używającymi różnych platform i środowisk programistycznych.



Języki programowania określone w normie

LAD (Ladder Diagram):

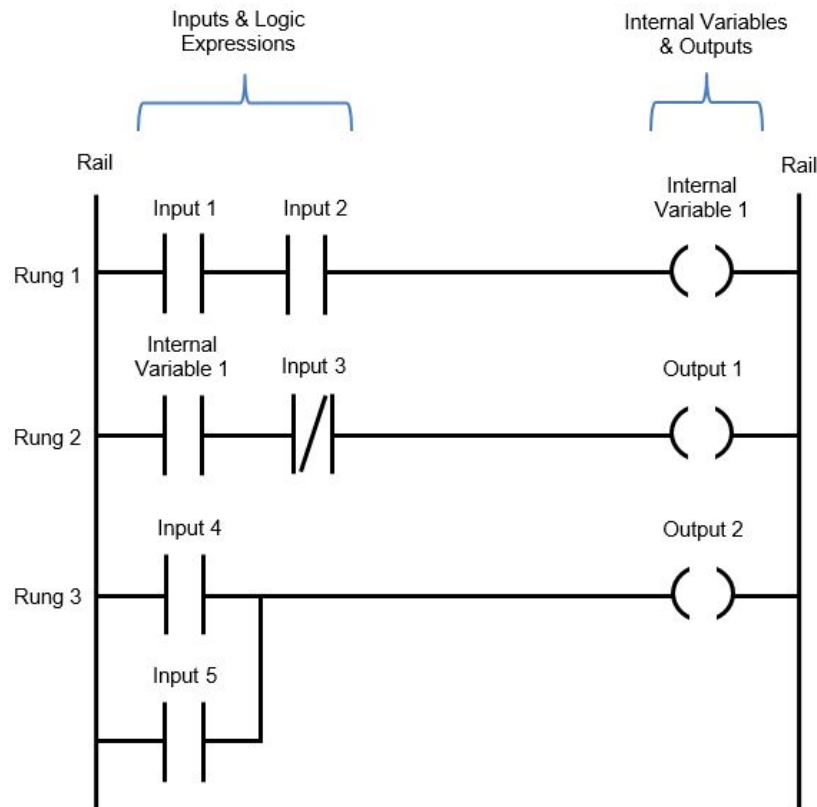
- **Opis:**
LAD, czyli schemat drabinkowy, jest językiem graficznym opartym na schematach elektrycznych obwodów przekaźnikowych.
- **Charakterystyka:**
 - Przejrzystość i łatwość nauki, szczególnie dla osób z doświadczeniem w elektryce.
 - Logika programu przedstawiana w formie "drabinki", z wejściami po lewej stronie i wyjściami po prawej.
- **Zastosowanie:**
Proste procesy sterowania, takie jak włączanie/wyłączanie urządzeń czy sekwencyjne sterowanie maszynami.



Języki programowania określone w normie

LAD (Ladder Diagram):

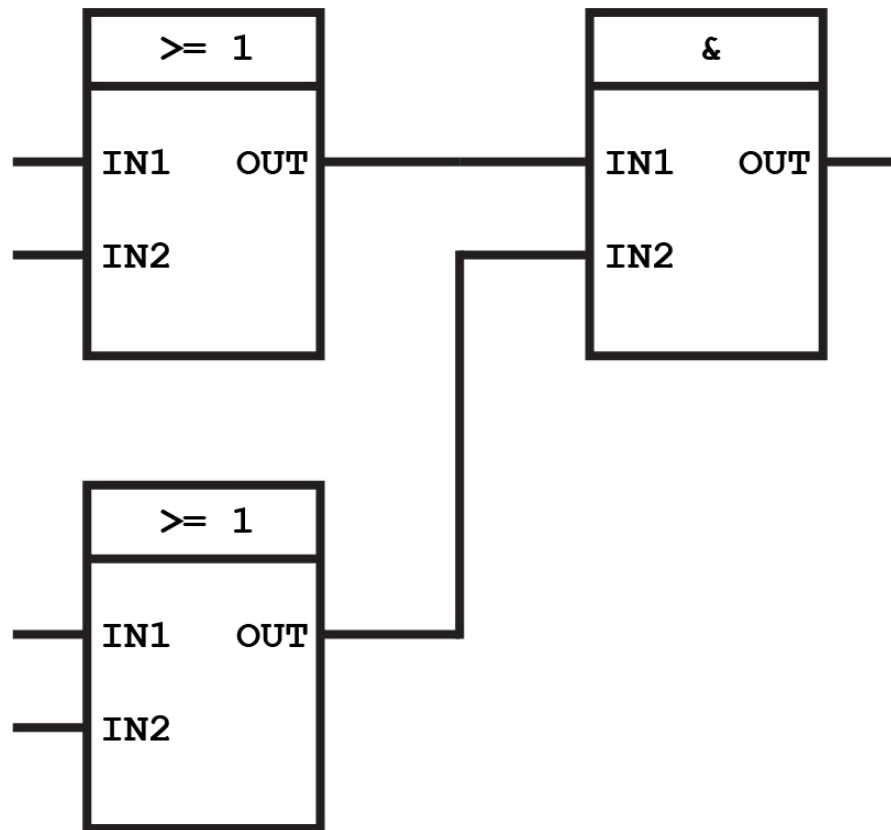
- **Zalety:**
 - **Łatwość nauki:** Idealny dla elektryków, którzy mają doświadczenie z tradycyjnymi schematami elektrycznymi.
 - **Intuicyjny interfejs graficzny:** Logika programu jest przedstawiana w formie „drabinki”, co ułatwia odczytywanie i diagnozowanie problemów.
 - **Szybkie wdrożenie:** Doskonale nadaje się do prostych aplikacji sterowania.
- **Wady:**
 - **Ograniczona złożoność:** Trudny do zastosowania w bardziej zaawansowanych systemach sterowania.
 - **Mniej efektywny dla dużych projektów:** Złożone logiki mogą być trudne do zarządzania i nieczytelne.



Języki programowania określone w normie

FBD (Function Block Diagram):

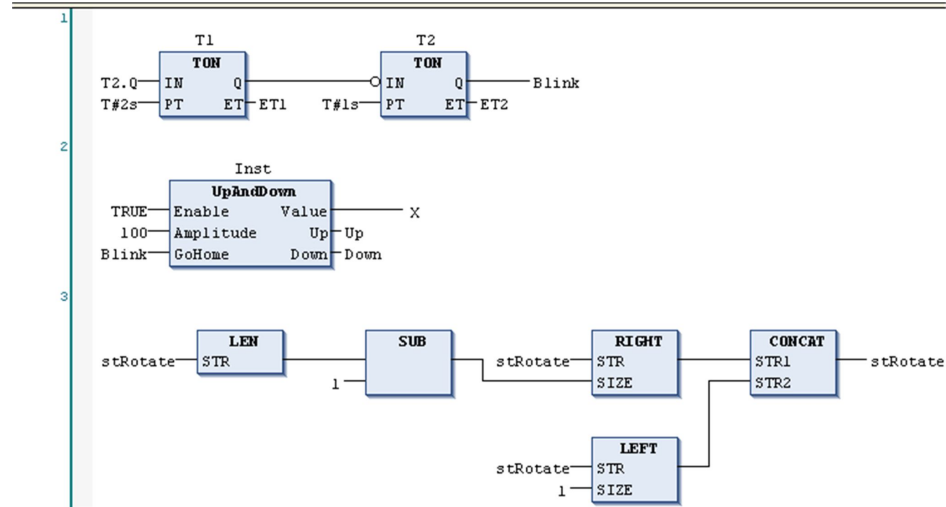
- **Opis:**
FBD to język graficzny, w którym logika jest przedstawiana jako połączenie bloków funkcjonalnych.
- **Charakterystyka:**
 - Każdy blok funkcjonalny odpowiada konkretnej operacji, np. AND, OR, liczniki czy timery.
 - Łatwość wizualizacji i debugowania skomplikowanych procesów.
- **Zastosowanie:**
Sterowanie ruchem, procesy technologiczne wymagające zaawansowanej logiki.



Języki programowania określone w normie

FBD (Function Block Diagram):

- **Zalety:**
 - **Intuicyjność:** Graficzne przedstawienie logiki w formie połączeń między blokami funkcjonalnymi.
 - **Elastyczność:** Łatwo integruje elementy takie jak liczniki, timery, operacje logiczne.
 - **Czytelność dla procesów sterowania:** Doskonale nadaje się do wizualizacji skomplikowanych relacji między funkcjami.
- **Wady:**
 - **Trudna obsługa dużych projektów:** Bardzo złożone schematy mogą stać się nieczytelne.
 - **Wymagania sprzętowe:** Wymaga dużej mocy obliczeniowej w przypadku bardzo szczegółowych bloków.



Języki programowania określone w normie

ST (Structured Text):

- **Opis:**
ST to tekstowy język programowania podobny do języków wysokopoziomowych, takich jak Pascal.
- **Charakterystyka:**
 - Elastyczność w pisaniu złożonych algorytmów i operacji matematycznych.
 - Struktura programu oparta na instrukcjach warunkowych, pętlach i zmiennych.
- **Zastosowanie:**
Zaawansowane aplikacje, obliczenia matematyczne, złożone systemy sterowania.

```
1 PROGRAM POU
2 VAR
3 MotorOverload : ARRAY[1..3] OF BOOL;
4 MotorOverCurrent : ARRAY[1..3] OF BOOL;
5 MotorPBStart : ARRAY[1..3] OF BOOL;
6 MotorPBStop : ARRAY[1..3] OF BOOL;
7 MotorCoil : ARRAY[1..3] OF BOOL;
8 i : INT;
9
10 END_VAR
11
```

```
1
2 FOR i:=1 TO 3 BY 1 DO
3
4 IF MotorPBStop[i] OR MotorOverload[i] OR MotorOverCurrent[i] THEN
5 MotorCoil[i]:=0;
6 ELSIF MotorPBStart[i] THEN
7 MotorCoil[i]:=1;
8 END_IF
9 END_FOR
```

Języki programowania określone w normie

ST (Structured Text):

- **Zalety:**

- **Wysoka elastyczność:** Umożliwia pisanie złożonych algorytmów i funkcji matematycznych.
- **Kompaktowy kod:** W porównaniu z LAD czy FBD, pozwala na bardziej zwarte przedstawienie logiki.
- **Bliskość języków programowania wysokiego poziomu:** Idealny dla inżynierów programistów z doświadczeniem w Pascalu, C++ czy Pythonie.

- **Wady:**

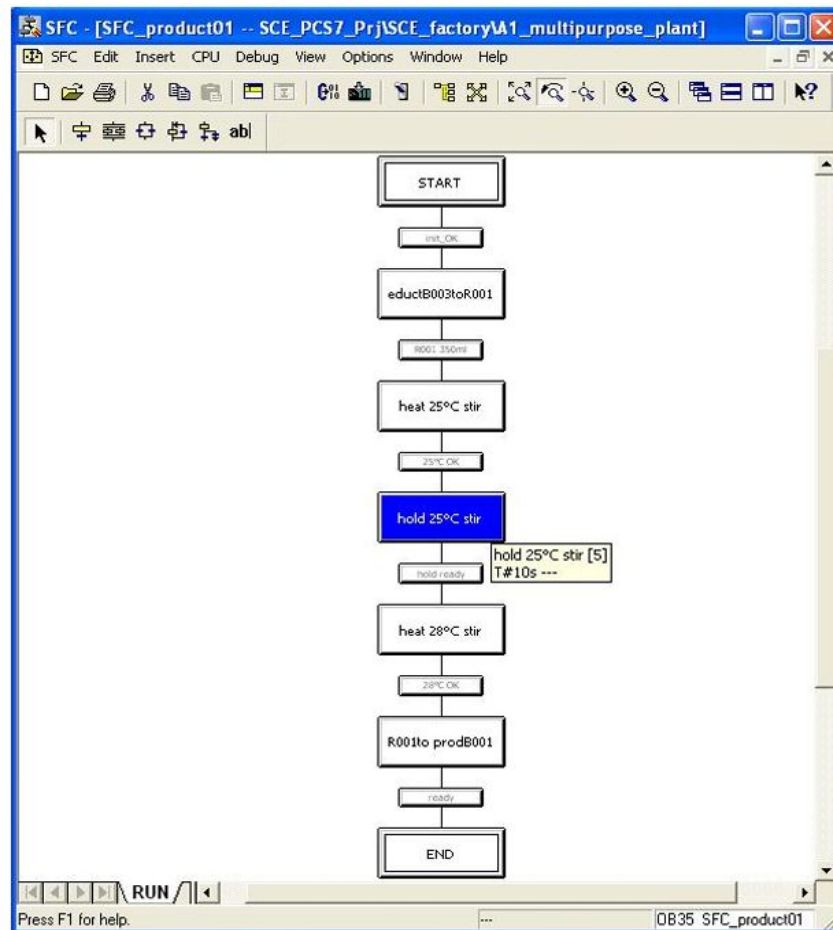
- **Wymaga znajomości programowania tekstowego:** Nie jest intuicyjny dla osób przyzwyczajonych do graficznych języków.
- **Trudniejsze debugowanie:** Dla osób bez doświadczenia, identyfikacja błędów może być bardziej czasochłonna.

```
1 IF #start = 1 THEN
2     //comment
3     "Max_nr" := #Array[0];
4 FOR #i := 1 TO 10 DO
5     // Statement section FOR
6     IF #Array[#i] > "Max_nr" THEN
7         "Max_nr" := #Array[#i];
8     END_IF;
9 END_FOR;
10 END_IF;
11
```

Języki programowania określone w normie

SFC (Sequential Function Chart):

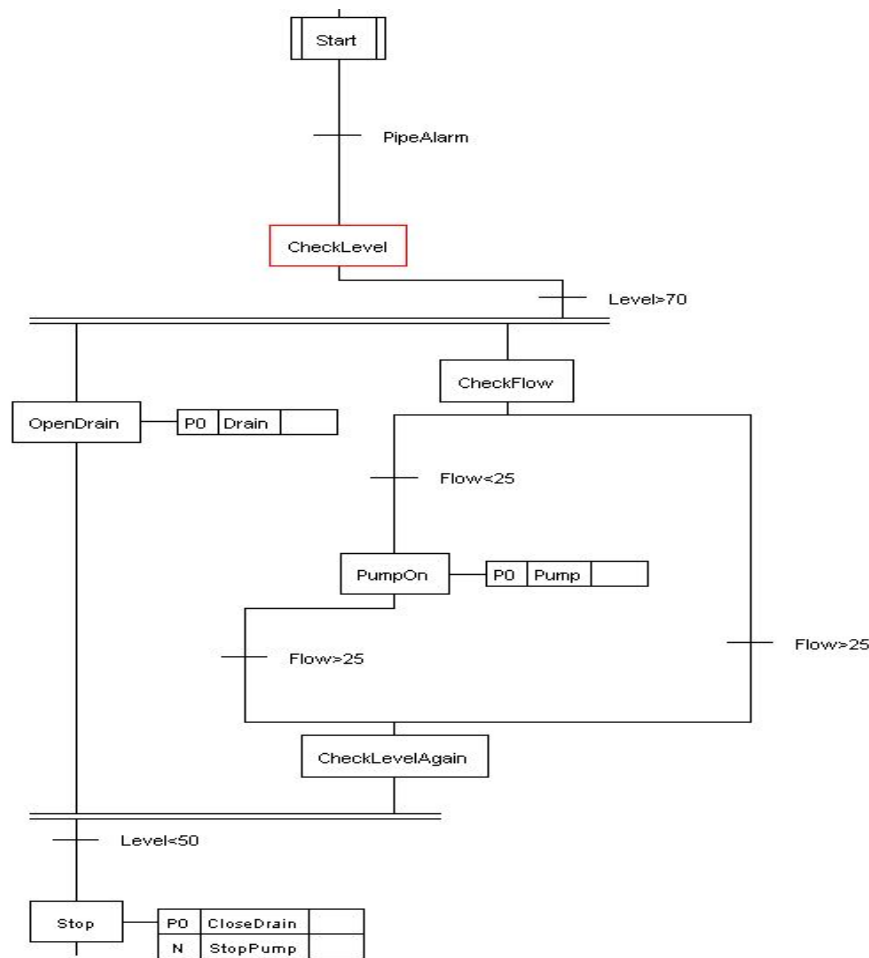
- **Opis:**
SFC to język graficzny wykorzystywany do programowania sekwencji działań w procesach przemysłowych.
- **Charakterystyka:**
 - Opiera się na etapach (steps) i przejściach (transitions), które definiują kolejność wykonywania operacji.
 - Idealny do tworzenia przejrzystych i logicznych sekwencji działań.
- **Zastosowanie:**
Procesy etapowe, takie jak linie montażowe, maszyny pakujące.



Języki programowania określone w normie

SFC (Sequential Function Chart):

- **Zalety:**
 - **Przejrzystość sekwencji:** Graficzne przedstawienie procesów w etapach (steps) i przejściach (transitions) ułatwia projektowanie sekwencji działań.
 - **Doskonale nadaje się do procesów krok po kroku:** Idealny do sterowania etapowego w maszynach pakujących, liniach montażowych itp.
- **Wady:**
 - **Złożoność projektowania zaawansowanych systemów:** Implementacja bardzo skomplikowanych procesów wymaga doświadczenia i dobrego planowania.
 - **Ograniczona elastyczność:** Nie nadaje się do zaawansowanych obliczeń czy nieliniowych procesów.



Porównanie języków programowania

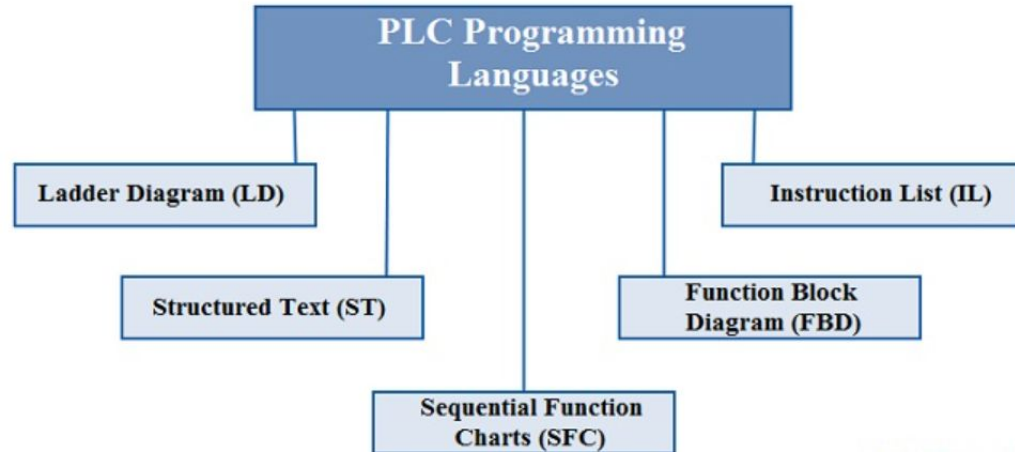
Kiedy wybrać dany język?

1. Proste aplikacje:

- **LAD:** Najlepszy wybór dla projektów opartych na podstawowej logice sterowania (np. włączanie/wyłączanie urządzeń, kontrola prostych sekwencji).
- **FBD:** Sprawdza się w aplikacjach wymagających wizualizacji relacji między funkcjami, takich jak proste systemy automatyki w liniach produkcyjnych.

2. Zaawansowane systemy:

- **ST:** Idealny do aplikacji wymagających skomplikowanych algorytmów matematycznych, precyzyjnej analizy danych i obliczeń.
- **SFC:** Najlepszy wybór do procesów etapowych i sekwencyjnych, takich jak linie montażowe, maszyny pakujące czy kontrola procesów technologicznych.



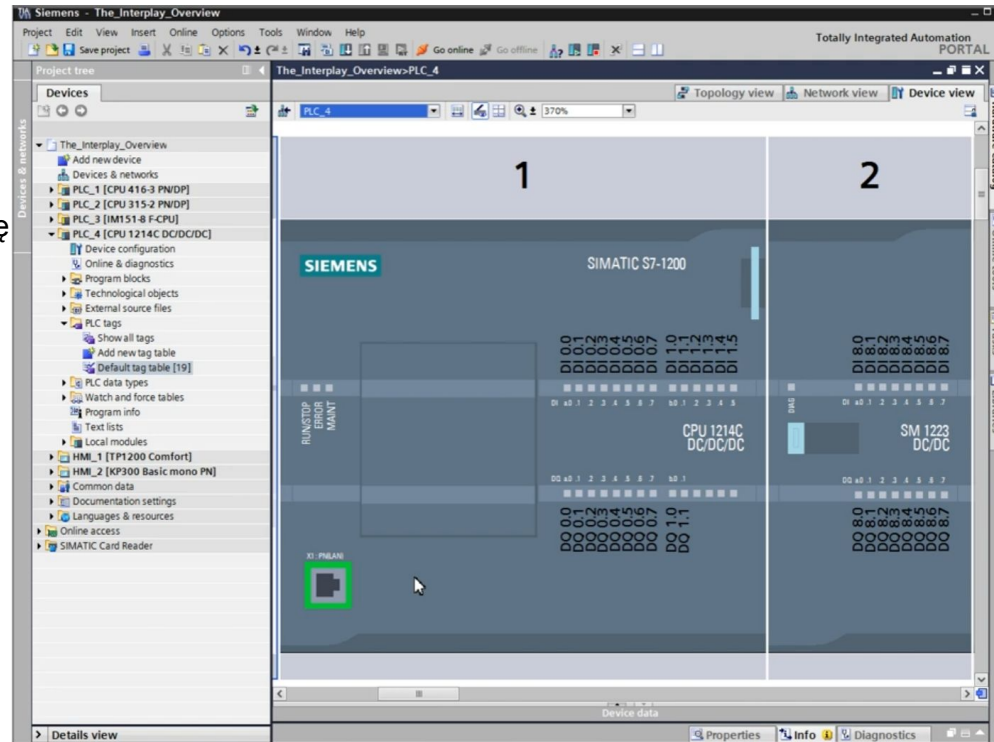


4
Narzędzia i
Środowiska
programistyczn
e

Popularne środowiska do programowania PLC

Siemens TIA Portal:

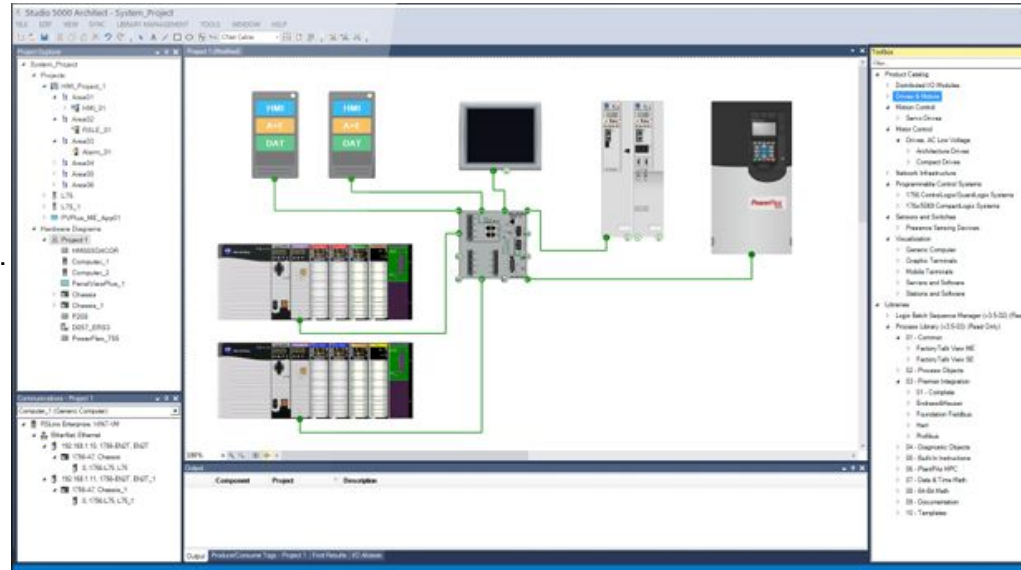
- **Opis:**
Totally Integrated Automation Portal (TIA Portal) to kompleksowe środowisko programistyczne firmy Siemens, które umożliwia programowanie, symulację i diagnostykę sterowników z serii SIMATIC (np. S7-1200, S7-1500).
- **Zalety:**
 - Intuicyjny interfejs użytkownika.
 - Integracja z symulatorem S7-PLCSIM.
 - Rozbudowane narzędzia diagnostyczne i możliwość obsługi różnych języków programowania.
- **Zastosowanie:**
Kompleksowe aplikacje przemysłowe, integracja systemów automatyki.



Popularne środowiska do programowania PLC

Rockwell Studio 5000:

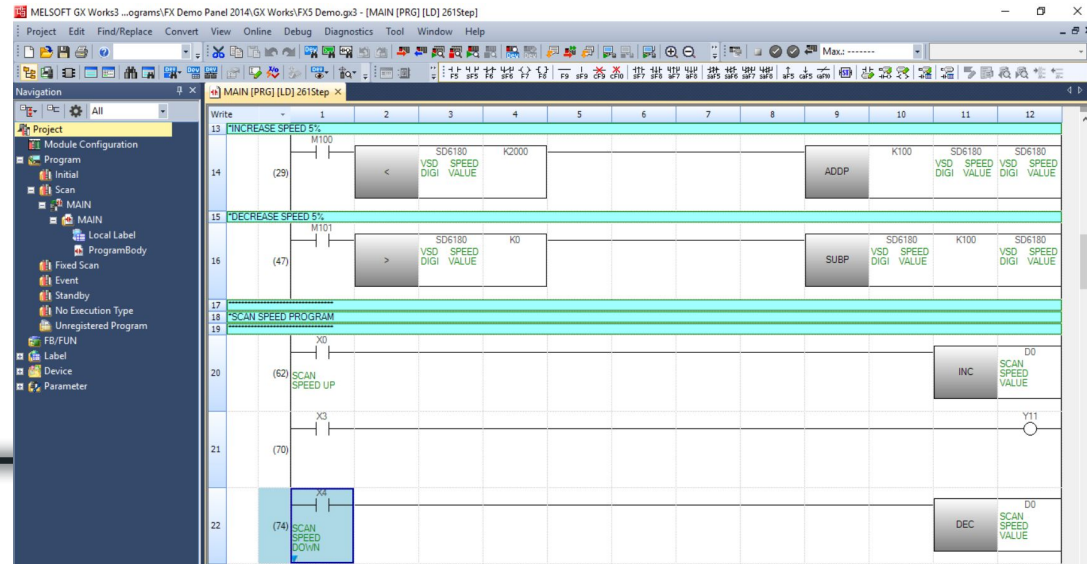
- **Opis:**
Środowisko programistyczne firmy Rockwell Automation, przeznaczone do programowania sterowników Allen-Bradley (ControlLogix, CompactLogix).
- **Zalety:**
 - Silne wsparcie dla aplikacji Przemysłu 4.0.
 - Wbudowane narzędzia wizualizacji i symulacji.
- **Zastosowanie:**
Duże projekty w przemyśle motoryzacyjnym, chemicznym i farmaceutycznym.



Popularne środowiska do programowania PLC

Mitsubishi GX Works:

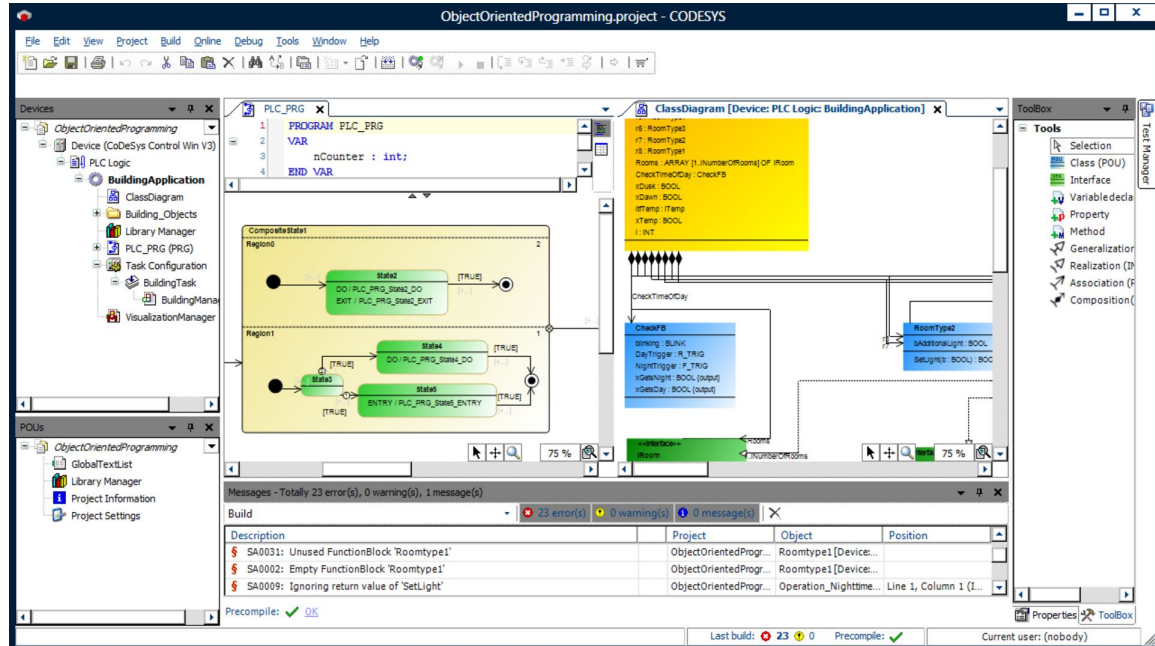
- **Opis:**
Oprogramowanie dedykowane dla sterowników z serii MELSEC firmy Mitsubishi Electric.
- **Zalety:**
 - Rozbudowane opcje konfiguracji i wizualizacji.
 - Integracja z systemami SCADA i HMI.
- **Zastosowanie:**
Przemysł maszynowy, automatyka budynkowa



Popularne środowiska do programowania PLC

CODESYS:

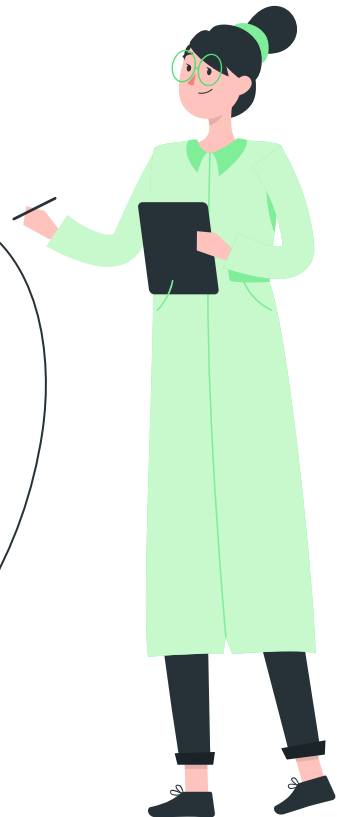
- **Opis:**
Uniwersalne środowisko programistyczne zgodne z normą IEC 61131-3, wspierające szeroką gamę sterowników od różnych producentów.
- **Zalety:**
 - Darmowa wersja edukacyjna.
 - Możliwość tworzenia aplikacji w dowolnym języku normy IEC 61131-3.
- **Zastosowanie:**
Projekty edukacyjne, małe i średnie aplikacje przemysłowe.





5

Przyszłość sterowników PLC



Sterowniki z otwartą architekturą

Czym jest otwarta architektura w sterownikach PLC?

- Tradycyjne sterowniki PLC korzystają z zamkniętego środowiska i ograniczonych możliwości programowania.
- Nowoczesne sterowniki z otwartą architekturą pozwalają na:
 - Programowanie w językach wysokopoziomowych (Python, C++).
 - Integrację z aplikacjami open-source i narzędziami deweloperskimi.
 - Współpracę z szeroką gamą protokołów komunikacyjnych (OPC UA, MQTT).



Sterowniki z otwartą architekturą

Przykłady sterowników z otwartą architekturą:

- **Bosch Rexroth ctrlX CORE:**
 - Oparty na systemie operacyjnym Linux.
 - Umożliwia tworzenie aplikacji w środowisku open-source.
 - Zapewnia dostęp do cyfrowego sklepu z aplikacjami (ctrlX Store).
- **Phoenix Contact PLCnext:**
 - Platforma umożliwiająca jednoczesne wykorzystanie IEC 61131-3 i języków wysokopoziomowych.
 - Idealny do aplikacji IoT i integracji z systemami sztucznej inteligencji.

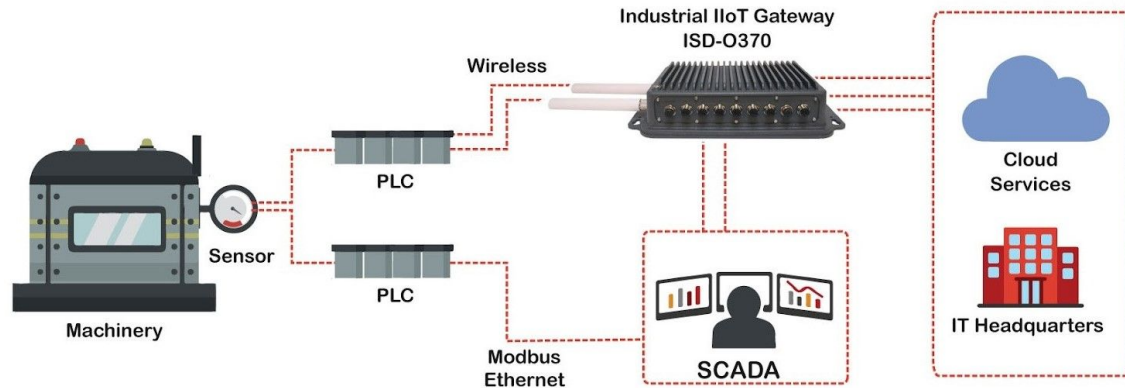


Sterowniki z otwartą architekturą

Korzyści otwartej architektury:

- **Elastyczność:** Możliwość dostosowania sterowników do specyficznych potrzeb aplikacji.
- **Integracja:** Łatwiejsze połączenie z nowoczesnymi systemami IT oraz IoT.
- **Skalowalność:** Rozbudowa systemów w miarę wzrostu wymagań bez konieczności zmiany sprzętu.

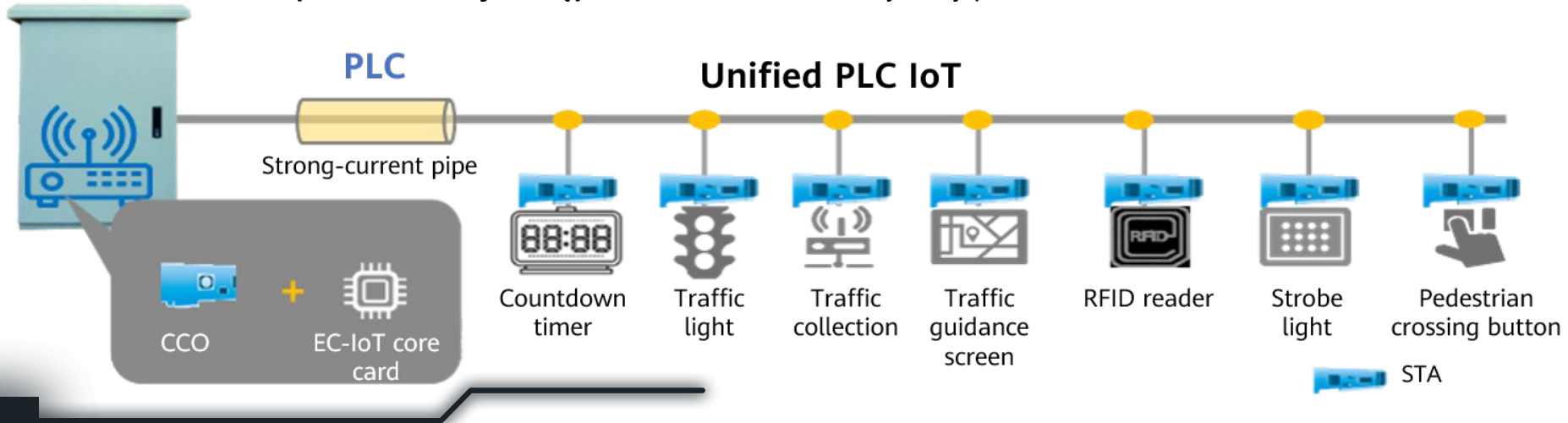
Integration of PLC with Industrial IoT



Integracja z IoT i Przemysłem 4.0

Nowoczesne sterowniki jako część systemów IIoT (Industrial Internet of Things):

- Sterowniki PLC stają się centralnymi elementami w sieci urządzeń połączonych, które umożliwiają zbieranie danych w czasie rzeczywistym z czujników i urządzeń wykonawczych.
- Dzięki komunikacji z chmurą, sterowniki PLC mogą:
 - **Analizować dane** w czasie rzeczywistym (np. monitorowanie zużycia energii).
 - **Przewidywać awarie** poprzez analizę danych historycznych (konserwacja predykcyjna).
 - **Zapewniać zdalny dostęp** do monitorowania i modyfikacji procesów.



Integracja z IoT i Przemysłem 4.0

Przemysł 4.0 – Sterowniki PLC w zautomatyzowanych fabrykach:

- W Przemysle 4.0 sterowniki PLC współpracują z innymi technologiami, takimi jak robotyka, systemy MES (Manufacturing Execution System) oraz ERP (Enterprise Resource Planning).
- Umożliwiają wdrażanie takich koncepcji jak:
 - **Cyfrowe bliźniaki** (Digital Twins), które pozwalają na symulację procesów w czasie rzeczywistym.
 - **Elastyczna produkcja**, gdzie sterowniki PLC są zdolne do adaptacji procesów w zależności od zmieniających się wymagań produkcyjnych.



Integracja z IoT i Przemysłem 4.0

Przykłady zastosowań:

- [Monitorowanie zużycia energii w czasie rzeczywistym w dużych fabrykach.](#)
- [Automatyczna optymalizacja procesów produkcyjnych dzięki algorytmom uczenia maszynowego.](#)
- https://elektrotechnikautomatyk.pl/artykuly/od-wykrywania-anomalii-do-zwiekszania-efektywnosci-produkcji?utm_source=chatgpt.com



Dziękuję

za

uwagę!



GRUPA NA FACEBOOKU

o mnie

ŹRÓDŁA

- <https://new.siemens.com/global/en/products/automation/systems/industrial/plc/simatic-s7.html>
- <https://www.rockwellautomation.com/en-us/products/hardware/allen-bradley/programmable-controllers/large-controllers.html>
- <https://www.mitsubishielectric.com/fa/products/cnt/plc/index.html>
- <https://www.se.com/ww/en/product-range/1465-modicon-m221-logic-controller-for-simple-machines/>
- <https://industrial.omron.eu/en/products/sysmac-platform>
- <https://www.unitronicsplc.com/>
- <https://www.hornerautomation.com/xl7/>
- <https://portalautomatyki.pl/sterowniki-plc>
- <https://automatykab2b.pl/raporty/48302-sterowniki-programowalne-plc-pac-raport-rynek>
- <https://controlengineering.pl/polski-rynek-sterownikow-programowalnych-plc/>
- <https://telmatik.pl/>
- <https://www.astor.com.pl/oferta/automatyzacja/sterowanie/sterowniki-plc.html>
- <https://www.stercontrol.pl/sterowniki-plc>
- https://www.tme.eu/pl/katalog/sterowniki-plc_112601/
- <https://www.unitronicsplc.pl/>
- <https://przemysl-40.pl/kategoria/automatyka/>
- <https://iautomatyka.pl/>
- <http://etacar.put.poznan.pl/marcin.kielczewski/ZSP05.pdf>
- <https://eia.pg.edu.pl/documents/1113028/0/SP2016%20W2%20-%20PLC,%20PAC,%20DCS%20-%20%20definicje,%20podzia%C5%82,%20historia,%20rynek,%20miejsce%20w%20systemie%20automatyki.pdf>
- https://dbc.wroc.pl/Content/7791/PDF/Pawlak_Sterowniki_programowalne.pdf
- <https://www.codesys.com/the-system/why-codesys.html>
- https://isd-soft.com/tech_blog/plc-programming-languages-short-overview/
- <https://infosys.beckhoff.com/english.php?content=../content/1033/tcplccontrol/925244555.html&id=>
- <https://www.automation.siemens.com/sce-static/learning-training-documents/pcs7/v7-0/p01-07-sequential-function-chart-rc1012-en.pdf>
- <https://medium.com/@mouhammad.hamshoo/structured-text-st-in-plc-and-why-you-must-ditch-ladder-logic-595e38d4d9ad>
- <https://control.com/textbook/programmable-logic-controllers/ladder-diagram-ld-programming/>
- http://automatyka.kia.prz.edu.pl/attachments/article/13/Sterowniki_IEC61131-3.pdf
- <https://pl.rs-online.com/web/p/plc-programowalne-sterowniki-logiczne/2741296>
- <https://www.automationreadypanels.com/plc-systems/iec-61131-standard-for-industrial-automation-programming/>
- <https://www.plcacademy.com/function-block-diagram-programming/>
- <https://ladderlogicworld.com/ladder-logic-programming/>
- <https://www.se.com/pl/pl/product/CEXADSCZZTPAZZ/licencja-ecostruxure-control-expert-dodatek-bezpiecze%C5%84stwa-m580-dla-l-lub-xl-10-u%C5%BCytkownik%C3%B3w-zablokowan-y-w%C4%99ze%C5%82-cyfrowy/>