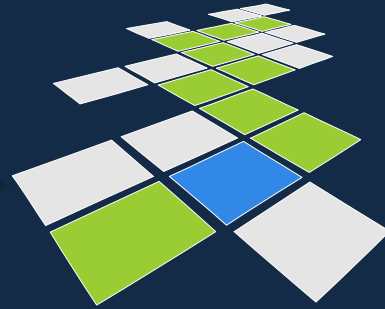
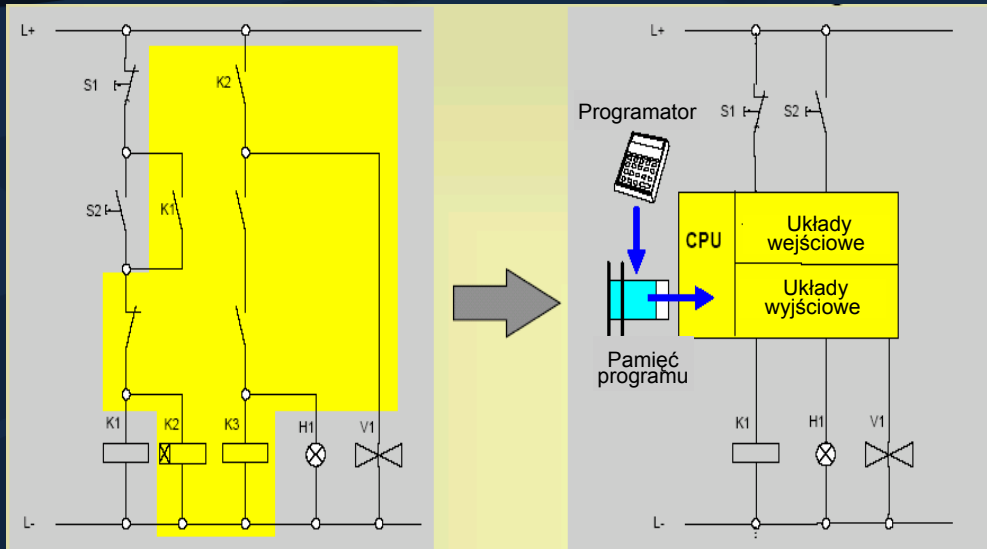




# Sterowniki PLC



**UCZELNIA**  
ONLINE



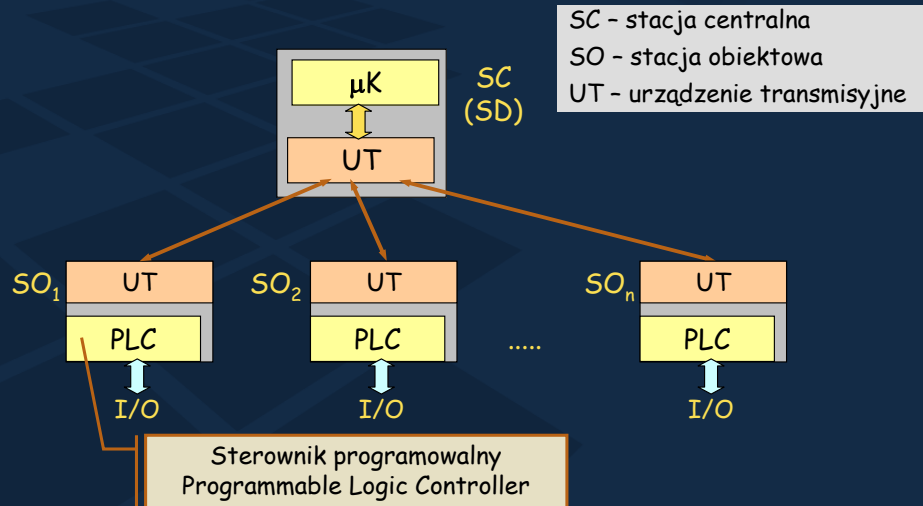
Zygmunt Kubiak

Sterowniki PLC (2)

Pierwsze systemy sterowania budowane były z układów elektromechanicznych, tzn. z różnego typu przekaźników, układów czasowych i liczników. Takie systemy zajmowały dużą objętość. Budowane były duże szafy sterownicze. Systemy charakteryzowały się dużą zawodnością. Kolejne problemy to różne i jednocześnie znaczne czasy propagacji poszczególnych elementów, drgania styków, mała liczba styków przekaźników, problemy nierównomiernego zużywania styków przekaźników itd.

Te problemy spowodowały, że w firmie Modicon postanowiono zastąpić elementy elektromechaniczne ich elektronicznymi analogami. Uzyskano w rezultacie system sterowania znacznie bardziej elastyczny i niezawodny, prostszy w uruchamianiu i serwisowaniu, zajmujący zdecydowanie mniejszą objętość. Ten kierunek zmian doprowadził do powstania tzw. sterowników programowalnych, w skrócie PLC (ang. Programmable Logic Controller).

Przykład pokazuje zastąpienie sterowania przekaźnikowego sterownikiem elektronicznym.

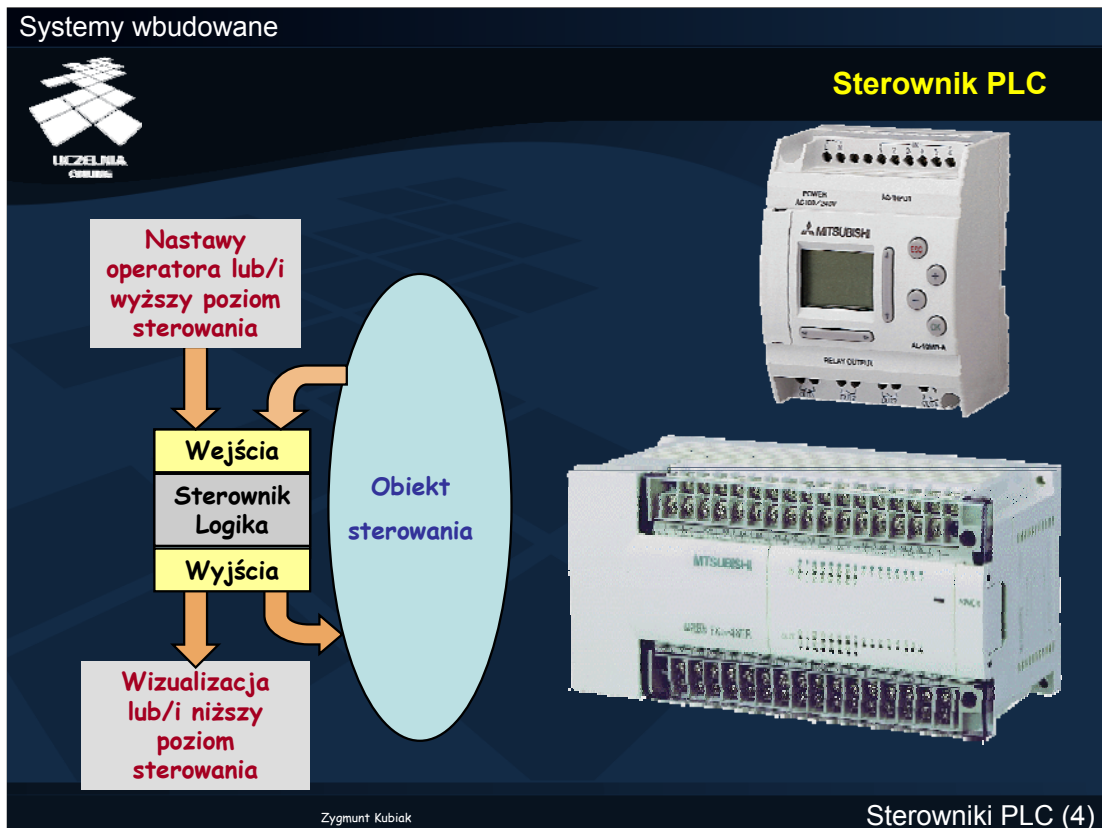


Slajd przedstawia system sterowania uwzględniający w swojej strukturze sterowniki PLC.

Istotny postęp nastąpił w latach 80-tych, w momencie pojawienia się względnie tanich monolitycznych procesorów 8-bitowych. Do najpopularniejszych należały Intel 8080 oraz Zilog Z80.

W stacji centralnej pojawiły się komputery o zdecydowanie większej mocy - mikrokomputery. Rozwiązania sterowników stacji obiektowych również stały się bardziej uniwersalne dzięki wykorzystaniu mikroprocesorów. Znalazły one zastosowanie w pierwszych programowalnych sterownikach – sterownikach PLC. W rezultacie stacje obiektowe mogły realizować różne zadania autonomiczne, np. regulację dwupoziomą wody w zbiorniku, czy regulacja ciśnienia gazu. W tych przypadkach zadania stacji centralnej sprowadzały się do ustalania limitów wartości oraz odczytu wartości bieżącej. Zmniejszyła się ilość danych wymienianych między SC i SO. W rezultacie system mógł obsługiwać więcej punktów we/wy, a stacja centralna mogła realizować dodatkowe zadania w postaci wizualizacji procesu, archiwizacji danych i raportowania.

Sterowniki PLC programowane są aktualnie przy specjalnych języków programowania z poziomu komputera PC. Głównym kryterium doboru sterownika do obiektu jest liczba we/wy. Wprowadzenie sterowników PLC w rezultacie spowodowało obniżkę kosztów projektowania i budowania systemów sterowania. Skrócił się czas wdrażania systemu, uprościło się serwisowanie a zwiększyła się niezawodność systemu.



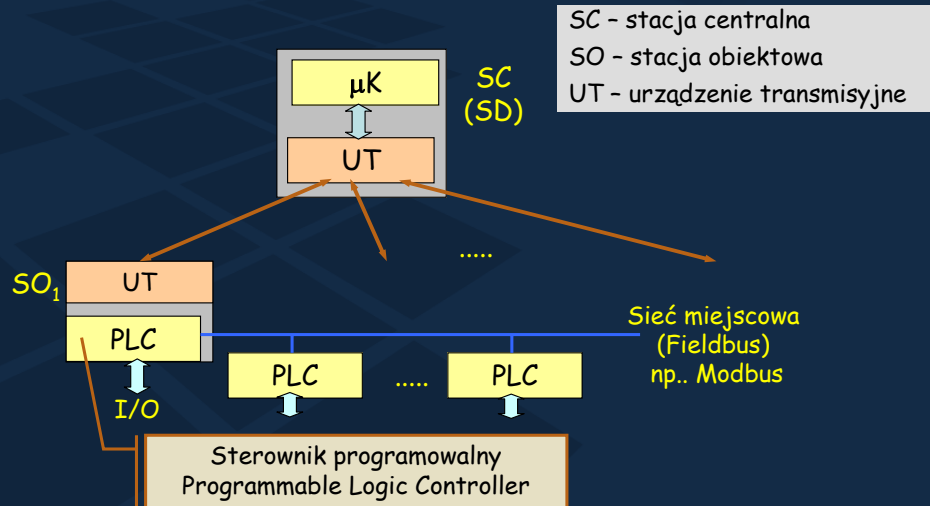
Zygmunt Kubiak

Sterowniki PLC (4)

Slajd ilustruje ogólną zasadę powiązania sterownika PLC z obiektem. Na wejścia sterownika wprowadzana jest informacja o stanie obiektu a ponadto nastawy operatora lub/i informacje z wyższego poziomu sterowania. Wyjścia sterujące połączone są z urządzeniami wykonawczymi obiektu sterowania. Ponadto wyjścia mogą być wykorzystane dla zadań prezentacji stanu obiektu.

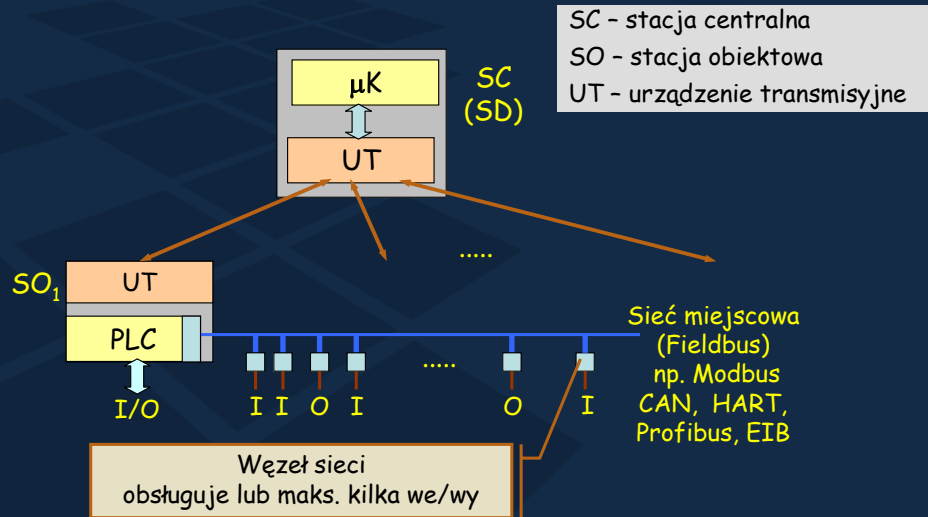
Na slajdzie przedstawiono również przykłady realizacji sterowników.

Pojedyncze sterowniki znajdują zastosowanie do obsługi maszyn, obrabiarek, gniazd produkcyjnych, tzn. w przypadkach gdy odległości między skrajnymi punktami we/wy nie przekraczają kilkunastu metrów. W innych rozwiązaniach, sterowniki łączone są w sieć. Ponadto PLC stosowane są jako urządzenia zarządzające siecią miejscową, bezprzewodową siecią sensorową lub jako urządzenie bramy dla tych sieci.

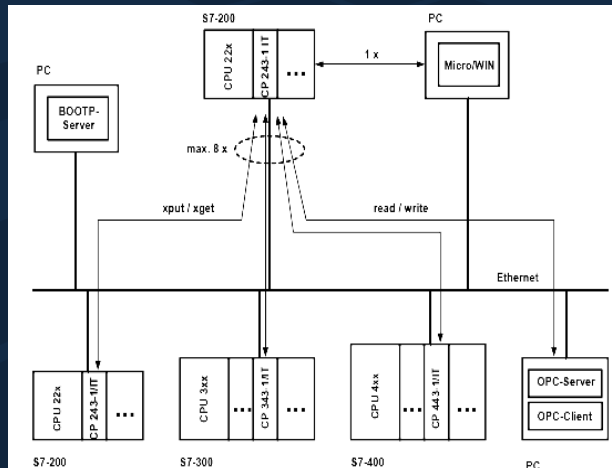


Początkowo zwiększanie liczby we/wy sterownika uzyskiwane było poprzez dołączanie dodatkowych modułów do sterownika matki. Skrajne rozwiązania tego typu osiągnęły rozmiar do kilku tysięcy we/wy. Można sobie wyobrazić problemy związane z rozprowadzeniem takiej liczby przewodów po obiekcie, problemy z uruchamianiem i diagnozowaniem takiego systemu.

Firma Modicon, która była również pomysłodawcą sterowników PLC, jako pierwsza wprowadziła sieci sterowników – podstawą jest sterownik o kilkunastu we/wy; jeżeli zapotrzebowanie jest większe to tworzona jest sieć sterowników. Dla powyższego celu firma Modicon opracowała i udostępniła protokół Modbus.



Protokół Modbus uzyskał dużą popularność. Stał się podstawą realizacji sieci przemysłowych nie tylko do łączenia sterowników PLC firmy Modicon. Zaliczany jest do tzw. sieci miejscowych (ang. Fieldbus). Są to sieci o znaczeniu ponad firmowym, ponad narodowym, udokumentowane standardami. Sieci te pracują na najniższym poziomie sterowania, tzn. na styku z nadzorowanym obiektem. Dzięki znacznej obniżce kosztów mikrokontrolerów węzeł sieci miejscowej może obsługiwać nawet pojedyncze wejście lub wyjście binarne, np. przełącznik światła. Sieci miejscowe były intensywnie rozwijane w latach 90-tych. Znanymi przykładami sieci miejscowych mogą być np. wspomniany już Modbus, HART, CAN, Profibus, ASi, EIB itd.. Producenci sterowników PLC oferują moduły do obsługi sieci różnych standardów.



### SIMATIC NET

**CP 243-1 IT**  
**Communications Processor**  
 for Industrial Ethernet and  
 Information Technology

Sterowniki PLC mogą być również wyposażone w moduły komunikacji poprzez sieć Internet/Intranet. Przykładowe urządzenia przystosowane do pracy w sieci Internet/Intranet firmy Siemens.

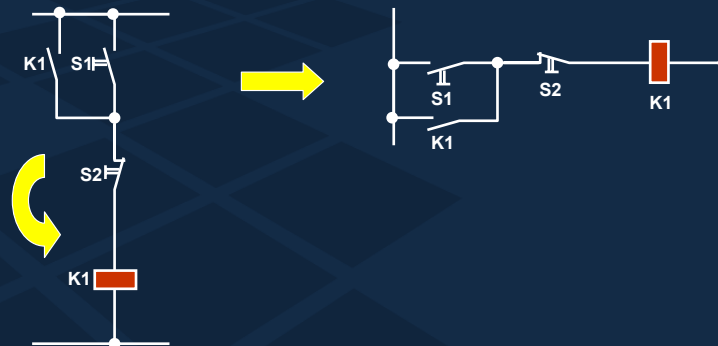


Pierwsze systemy sterowania realizowane były przy pomocy elementów przełącznikowych. Rysowane były często na arkuszach różnych formatów, często przypadkowych. Dlatego też już na tym etapie systemów sterowania, szukano pewnych uproszczonych, unormowanych rozwiązań.

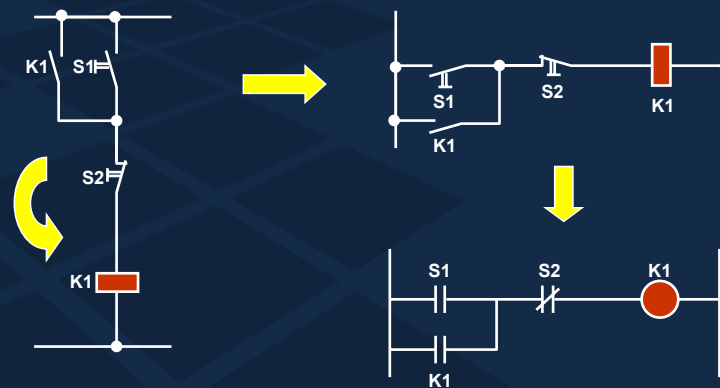




## Języki programowania Przełącznikowe rozwiązania systemów sterowania

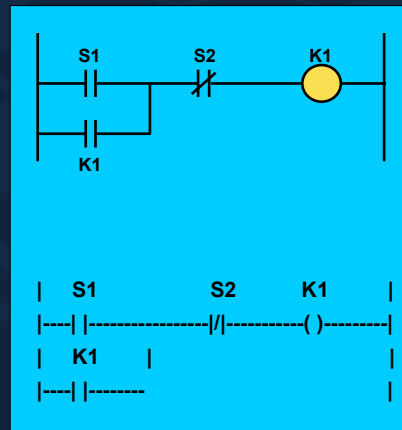


Pierwszy etap polega na obróceniu schematu o 90 stopni w kierunku przeciwnym do ruchu wskazówek zegara.

Schemat drabinkowy  
(ang. Ladder Diagram)

Drugi etap przekształcenia polega na wprowadzeniu uproszczonych symboli styków i cewek.

Uzyskano w efekcie tzw. schemat drabinkowy (ang. ladder diagram). W tej postaci schemat sterowania można przedstawić na arkuszu o stałej szerokości, niezależnie od jego złożoności.

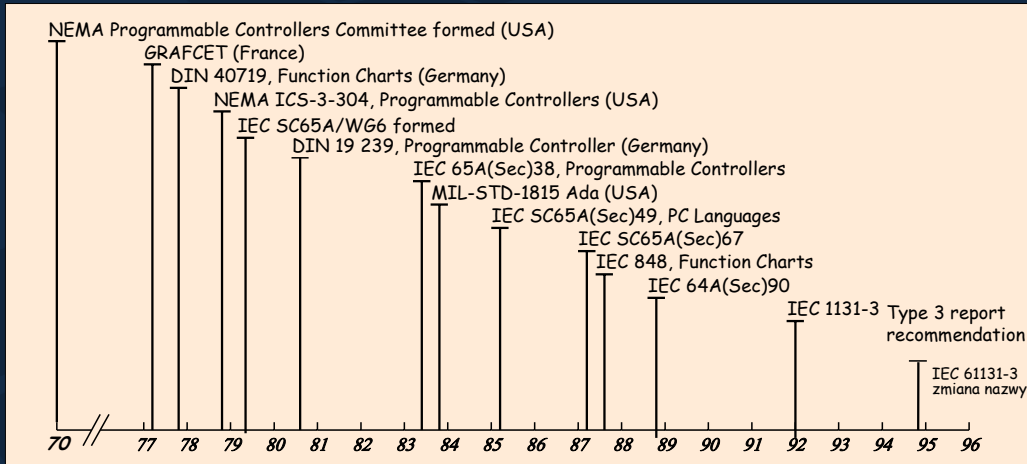


Schemat drabinkowy utworzony przy pomocy znaków tekstowych

Z drugiej strony można tego typu schemat można „narysować” stosując wyłącznie znaki tekstowe. Takie rozwiązanie było korzystne w czasach gdy upowszechnienie monitorów graficznych było niewielkie. Taki schemat można było wygenerować również na maszynie do pisania.



## Standaryzacja języków systemów przemysłowych

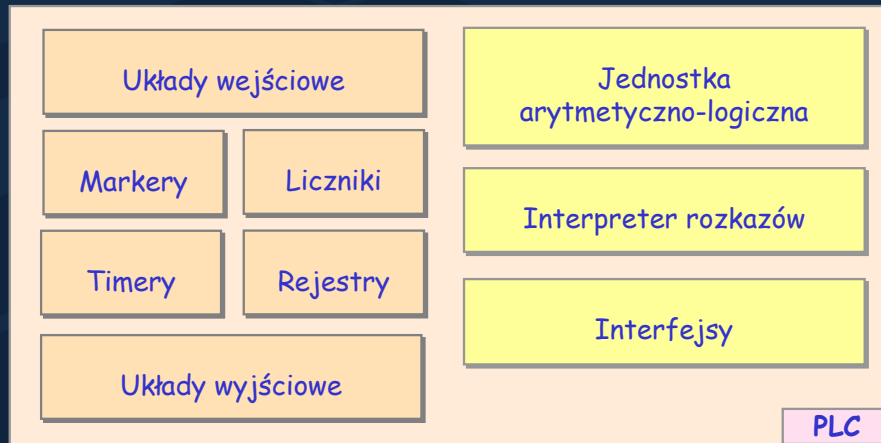


Source: Dr. J. Christensen

Na początkowym etapie rozwoju sterowników PLC każdy producent stosował swoje rozwiązania programistyczne. Prace normalizacyjne rozpoczęły się dosyć wcześnie bo w roku 1970. Trwały jednak bardzo długo ponieważ poszczególne języki programowania stosowane w sterownikach różnych firm znacznie się różniły. Końcowa postać specyfikacji dotyczącej języków programowania (IEC 1131-3) pojawiła się w latach 90-tych.



## Schemat logiczny sterownika PLC



Sterownik PLC od strony rozwiązań układowych niewiele różni się od innych sterowników. Do charakterystycznych różnic natomiast należą:

- architektura z punktu widzenia programisty,
- działanie sterownika,
- języki programowania,

**Pierwsza cecha** tkwi w oprogramowaniu, które powoduje, że programista użytkownik ma do swojej dyspozycji wyłącznie pewne układy funkcjonalne. Należą do nich **Układy wejściowe**, **Układy wyjściowe**, **Markery**, **Liczniki**, **Timery** i **Rejestry**. Natomiast zupełnie zbędna jest znajomość rzeczywistej architektury sterownika. Z punktu widzenia użytkownika nie ma znaczenia np. typ procesora, jego lista rozkazów, adresy pamięci, układów we/wy itp.

Program użytkownika na bieżąco tłumaczony jest przy pomocy Interpretera rozkazów.



Cykl pracy sterownika PLC

**Drugą cechą** współczesnych sterowników programowalnych jest ogólnie przyjęty model ich działania.

Po starcie i inicjalizacji sterownik przechodzi do podstawowego cyklu pracy. Cykl (skan) rozpoczyna się od równoległego odczytu wejść. Obraz stanu wejść zapamiętany zostaje w pamięci wewnętrznej sterownika. Instrukcje programu wykonywane są w naturalnej kolejności aż do instrukcji kończącej program – **END**. Jeżeli w trakcie wykonywania programu potrzebna jest informacja o stanie wejść to pobierana jest z pamięci obrazu. Podobnie aktualny stan wyjść zapisywany jest do pamięci wewnętrznej.

Po instrukcji **END** sterownik przechodzi do kolejnej fazy cyklu – stan wyjść z pamięci wewnętrznej zostaje przepisany do bufora wyjść fizycznych i jest podtrzymany przez kolejny cykl pracy sterownika.

Kolejna faza zamykająca pojedynczy cykl sterownika przeznaczona jest dla wewnętrznych operacji diagnostycznych oraz dla potrzeb komunikacji z urządzeniami zewnętrznymi (w tym z programatorem).



Zalecenia IEC1131-3	
Języki graficzne	Języki tekstowe
Schemat drabinkowy LD (ang. Ladder Diagram)  	Lista instrukcji IL (ang. Instruction List)  <pre> LD  A OR  B ORN C AND D ST  E           </pre>

**Trzecia cechą** wyróżniającą sterowniki PLC są języki programowania zgodne z zaleceniami normy IEC 1131 rozdz. 3 (IEC 61131-3). Ogólnie można je podzielić na graficzne i tekstowe.

Do najwcześniej stosowanych języków należą **Schemat drabinkowy LD** (ang. Ladder Diagram) oraz **Lista instrukcji IL** (ang. Instruction List).



Zalecenia IEC1131-3	
Języki graficzne	Języki tekstowe
Schemat bloków funkcyjnych FBD (ang. Function Block Diagram)	Tekst strukturalny ST (ang. Structured Text)
<pre>graph LR; A --- OR[OR]; B --- OR; C --- OR; OR --- AND[AND]; D --- AND; AND --- E;</pre>	$E = (A \text{ OR } B \text{ OR } C) \text{ AND } D$

Kolejnym językiem graficznym jest **Schemat bloków funkcyjnych FBD** (ang. Function Block Diagram). Jest on szczególnie chętnie stosowany w grupie najprostszycy sterownikóy PLC, dla których jest to z reguły jedyny dostępny język programowania. Język tekstowy **Tekst strukturalny ST** (ang. Structured Text) stosowany jest w grupie najmocniejszych sterownikóy. Jest on podobny do języków wyższego poziomu typu Pascal.

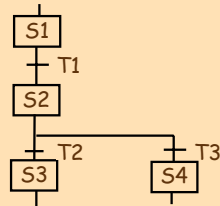




## Zalecenia IEC1131-3

Graf sekwencji  
SFC (ang. Sequential Function Chart)

## Postać graficzna

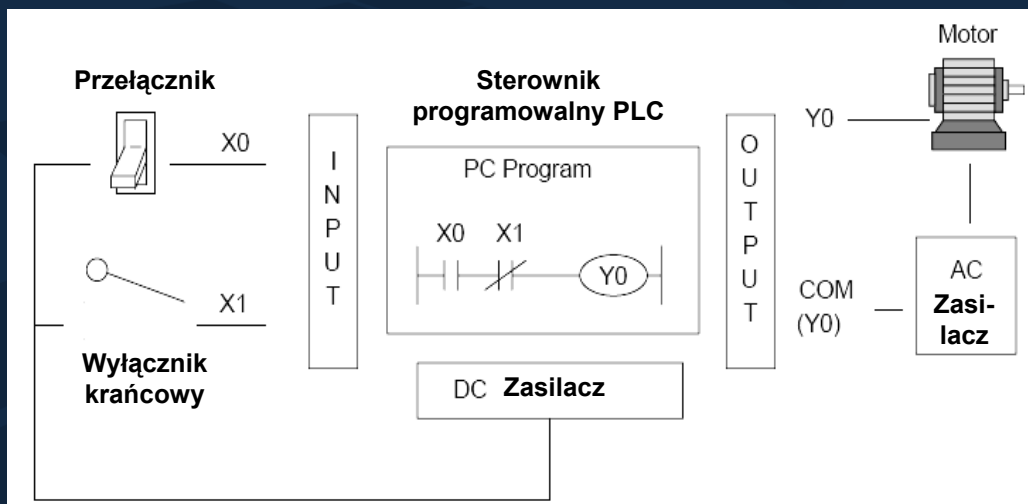


## Postać tekstowa

```
STEP S1
:
END STEP
TRANSITION
FROM S1 TO S2
:
```

W normie IEC 61131-3 zdefiniowano jeszcze metodę programowania sekwencyjnego – **Graf sekwencji SFC** (ang. Sequential Function Chart). Schemat graficzny SFC wymaga dodatkowego opisu w jednym z poprzednio wymienionych języków. Jak pokazano w dalszej części wykładu, jest to bardzo wygodny sposób programowania.

Ze względu na przyjęte ramy czasowe wykładu, dalsze rozważania ograniczone zostały do najwcześniej wprowadzonego a jednocześnie najbardziej popularnego języka - schematu drabinkowego (LD).



Zygmunt Kubiak

Sterowniki PLC (18)

Na rysunku przedstawiono prosty system sterowania silnika.

Do wejścia X0 sterownika dołączony jest przełącznik, który służy do załączania i wyłączania silnika. Wyłącznik krańcowy został dołączony do wejścia X1. Jego zadanie nie zostało dokładnie zdefiniowane. Może to być np. ogranicznik maksymalnych obrotów, może to być wyłącznik krańcowy urządzenia napędzanego przez silnik. Zadziałanie wyłącznika krańcowego ma wyłączyć silnik. Wejścia sterownika sterowane są prądem. Przepływ prądu załączenia wejścia, np. X0 = 1. Brak przepływu prądu oznacza wejście wyłączone, np. X0 = 0. Obwody przełączników, czujników zewnętrznych, dołączanych do wejść z reguły zasilane są z wewnętrznego zasilacza sterownika PLC. W sterownikach Mitsubishi wejścia oznaczane są literą X a wyjścia literą Y. Wejścia i wyjścia numerowane są ósemkowo, tzn. sterownik może być wyposażony w grupy wejść X0 – X7, X10 – X17, itd. Silnik dołączony jest do wyjścia Y0. Rozwiązania wyjść sterowników są różne. Mogą to być przekaźniki, tranzystory, triaki. Oznaczeniem wejścia sterownika jest styk. Może to być styk normalnie otwarty (na rys. np. X0) lub normalnie zamknięty (na rys. np. X1). Liczba styków obu typów praktycznie nie jest dla użytkownika ograniczana. Symbolem wyjścia jest przekaźnik, który składa się z cewki (na rys. np. Y0) dołączanej traktowanej jako argument instrukcji wyjściowej. Każde wyjście dowolną liczbą styków obu typów. Styki oznaczone są tak samo jak cewka.

Program w języku LD składają się z pewnych fragmentów, ograniczonych pionowymi liniami (tzw. linie zasilania), zwane szczeblami drabinki. Każdy szczebel drabinki rozpoczyna się stykiem normalnie otwartym lub zamkniętym, a kończy się elementem wyjściowym, np. cewka przekaźnika.

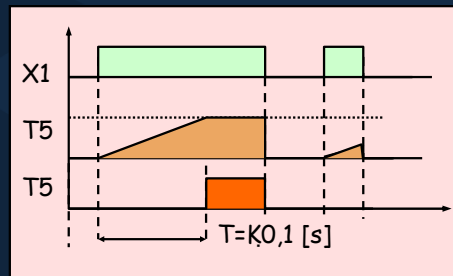
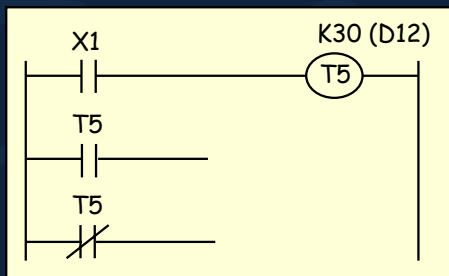
Połączenia styków tworzą warunek zadziałania przekaźnika. W rozważanym przypadku jest to iloczyn logiczny (szeregowe połączenie styków). **Y0 = X0 AND NOT X1**

Spełniony warunek możemy interpretować jako zamknięty obwód zasilający cewkę przekaźnika Y0.

Oprócz układów wyjściowych (przekaźników) sterownik wyposażony jest również w tzw. **Markery** (przekaźniki pomocnicze). Markery oznaczone są literą **M**. Mogą być dowolnie używane przez programistę, np. do zapamiętania binarnych wyników pośrednich.



## Czasomierze (Timery)



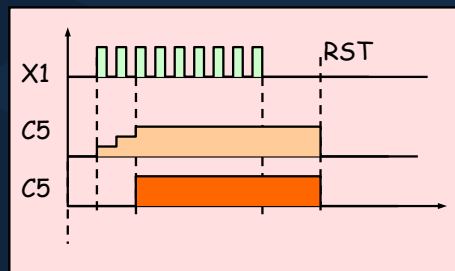
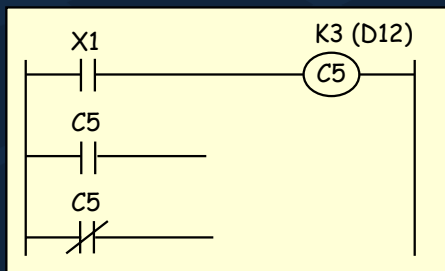
## Działanie timera.

Zamknięcie styku X1 ( $X1=1$ ) uruchamia licznik timera T5. Licznik ten zlicza impulsy z generatora wewnętrznego (tu o okresie 0,1s). Po osiągnięciu wartości określonej w rejestrze stałej (tu 30), tzn. w tym przypadku po czasie 3s następuje zamknięcie styku timera T5. Każdy timer dysponuje dowolną liczbą styków normalnie otwartych (styk bez przekreślenia) i dowolną liczbą styków normalnie zamkniętych (styk z przekreśleniem). Stan ten trwa tak długo, aż nie nastąpi rozwarzenie styku X1 ( $X1=0$ ) lub wyzerowanie licznika timera za pomocą instrukcji RST. Jeżeli X1 zostanie rozwarzony przed upływem 3s, styk timera (normalnie otwarty) nie zostanie zamknięty. Stan styków na rysunku zaznaczono kolorem czerwonym. Stała licznika może być podana w postaci wartości dziesiętnej – liczba dziesiętna poprzedzona dużą literą K (np. K30) lub jako zawartość rejestru (np. rej. D12).



## Schemat drabinkowy (LD) i Lista instrukcji (IL)

### Liczniki



Zygmunt Kubiak

Sterowniki PLC (20)

### Działanie licznika

Liczniki mają podobną budowę do timerów. Tak samo zawierają dwa 16-bitowe rejestry – rejestr stałej i rejestr licznika. Zasadnicza różnica polega na tym, że licznik zlicza impulsy zewnętrzne a nie z wewnętrznego generatora, jak to się dzieje w przypadku timera.

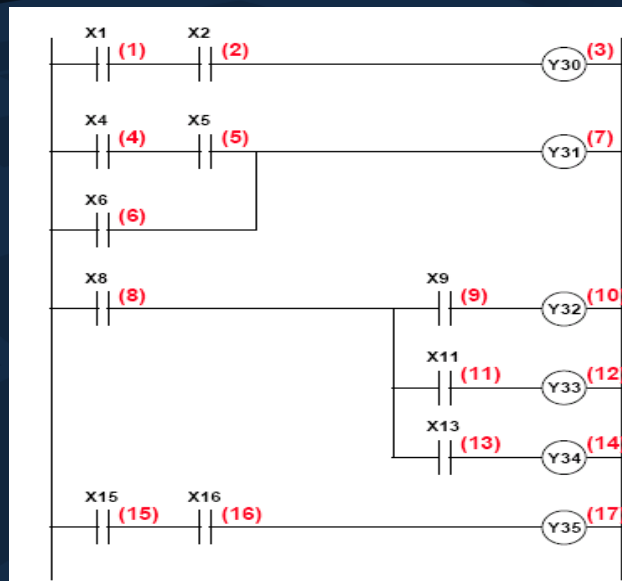
Opis przykładu użycia licznika.

Każde zbocze dodatnie (narastające), związane z załączeniem wejścia X1, powoduje zwiększenie zawartości licznika o 1. Gdy zawartość licznika osiągnie wartość stałej, następuje zablokowanie dalszego zliczania (patrz połączenie na schemacie) oraz zamknięcie styków normalnie otwartych i otwarcie normalnie zamkniętych. Stan taki trwa tak długo, aż licznik nie zostanie wyzerowany przy pomocy instrukcji RST. Dla danego licznika nie ma ograniczenia na liczbę styków, używanych w programie.

Liczniki obsługiwane są w ramach cyklu programowego. Ma to swoje konsekwencje w małej szybkości działania tych liczników. Drugim ograniczeniem jest filtr ustawiony na wejściach (domyślnie 10ms). To powoduje, że maksymalna częstotliwość zliczania impulsów nie przekracza kilkunastu Hz. Niektóre zastosowania przemysłowe wymagają wyższych częstotliwości. Wtedy stosowane są tzw. szybkie liczniki.



## Kolejność przetwarzania programu



Zygmunt Kubiak

Sterowniki PLC (21)

Program jest przetwarzany przez interpreter rozkazów w naturalnej kolejności:

- w ramach pojedynczego cyklu programu (jednego skanu), analizowane są kolejne szczeble drabinki (od góry do dołu), aż do instrukcji END (kończącej program);
- w ramach danego szczebla analizowany jest od lewej linii zasilania do prawej, warunek (zbudowany ze styków) i jeżeli jest prawdziwy to następuje wykonanie operacji wyjściowej;
- w przeciwnym przypadku, jak również po wykonaniu operacji wyjściowej, następuje przejście do analizy następnego szczebla drabinki.

W powyższym przykładzie, kolejność zaznaczona została numerami.

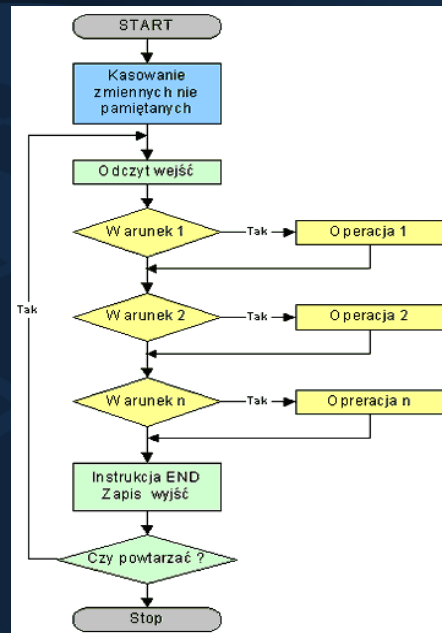
Każdy szczebel drabinki (poza wyjątkami) zawiera warunek zbudowany ze styków. Warunek ten może być bardzo skomplikowany – użytkownik praktycznie nie odczuwa ograniczeń wynikających z edytora programów.

Jeżeli warunek jest prawdziwy (np. dla powyższego przykładu, w drugim szczeblu drabinki –  $(X4 \text{ AND } X5) \text{ OR } X6 = 1$ ) to wykonana zostanie operacja wyjściowa (tu  $Y31=1$ ).



## Schemat drabinkowy (LD)

Schemat przetwarzania programu drabinkowego



Na rysunku przedstawiono szczegółowy algorytm prezentujący schemat przetwarzania programu drabinkowego. Warunek plus operacja (operacje) tworzą pojedynczy szczebel schematu drabinkowego. Można zauważyć, że przy takiej organizacji algorytmu nie ma znaczenia dla poprawności wykonywania programu, zamiana szczebli drabinki. Np. na rozważanym schemacie przestawienie warunków 1 i 2 nie ma wpływu na działanie programu.



- Dane wyjście użyte jest raz w programie (wyjątek – użycie instrukcji SET, co wymaga również instr. RST)
- Podstawowe przypadki
  - Określamy warunek aktywności danego wyjścia
  - Określamy warunki załączenia oraz wyłączenia wyjścia
  - Realizacja sekwencji czasowych
  - Przypadek mieszany
- Kolejność szczebli drabinki z reguły nie ma znaczenia na działanie programu
- Program kończymy instrukcją END

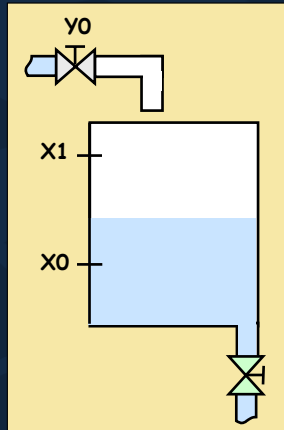
Poprawne tworzenie programów wymaga stosowania reguł przewidzianych w normie IEC 61131-3, ale również uwzględnienia ograniczeń i specyfiki poszczególnych edytorów. Tym drugim problemem nie będziemy się zajmować.

Przedstawione spojrzenie na programowanie sterowników PLC należy traktować jako uproszczone, ale wystarczające dla potrzeb i zadań tego kursu.

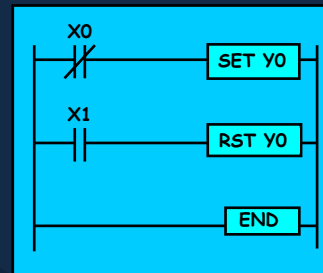
W dalszej części przedstawione zostaną dwa przykłady ilustrujące wspomniane zasady. W pierwszym przykładzie użyta zostanie zasada druga a w kolejnym przykładzie – pierwsza i trzecia.



## Przykład programu z określeniem warunków załączenia i wyłączenia wyjścia



Napełnianie zbiornika  
sterowanie dwustanowe



### Przykład sterowania napełnianiem zbiornika

Przyjęty algorytm pracy jest następujący – zawór napełniania **Y0** zostanie otwarty w momencie gdy poziom spadnie poniżej czujnika **X0**, natomiast zostanie zamknięty po osiągnięciu poziomu **X1**. Opróżnianie zbiornika nie jest kontrolowane przez system sterowania. Czujnik generuje sygnał logiczny „1” gdy jest zanurzony. Wykorzystano rozwiązanie polegające na **określeniu warunków załączenia i wyłączenia zaworu Y0**. Korzystamy z instrukcji **SET** do załączenia wyjścia **Y0** i **RST** do wyłączenia **Y0**. Należy pamiętać, że

•w programie instrukcja SET z takim samym argumentem może wystąpić tylko raz

•każde użycie instrukcji SET musi mieć swoje dopełnienie w instrukcji RST.

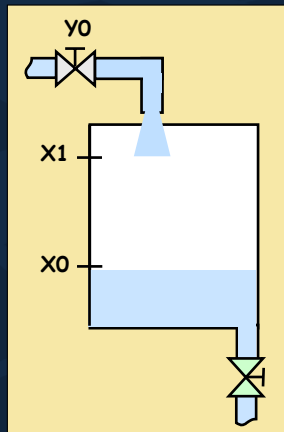
Uwaga

Instrukcja SET powoduje trwałe załączenie układu funkcjonalnego występującego jako argument instrukcji, (np. wyjścia Y). Wyłączenie wymaga użycia instrukcji RST.

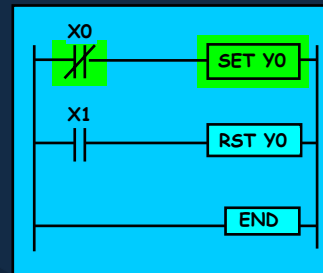




## Przykład programu z określeniem warunków załączenia i wyłączenia wyjścia



Napełnianie zbiornika  
sterowanie dwustanowe



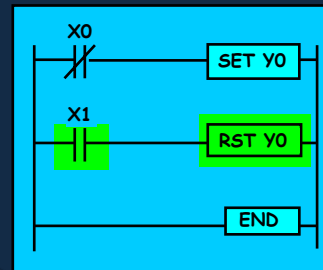
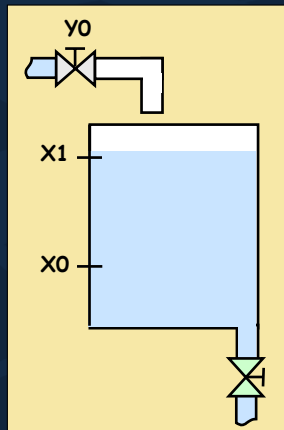
### Przykład sterowania napełnianiem zbiornika

Na rysunku sytuacja obniżenia poziomu medium w zbiorniku poniżej poziomu **X0**.



## Przykład programu z określeniem warunków załączenia i wyłączenia wyjścia

Napełnianie zbiornika  
sterowanie dwustanowe

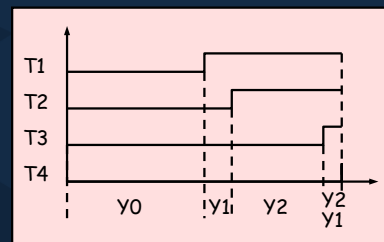
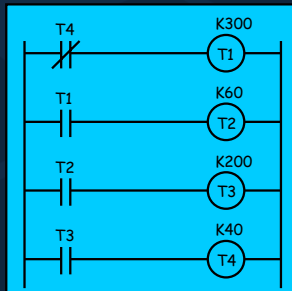
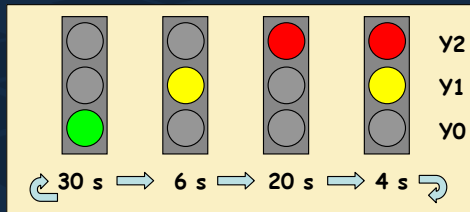


### Przykład sterowania napełnianiem zbiornika

Na rysunku sytuacja przekroczenia poziomu **X1**. Następuje wyłączenie wyjścia **Y0**.



## Przykład programu z realizacją sekwencji czasowych



Zygmunt Kubiak

Sterowniki PLC (27)

### Przykład sterowania sygnalizatorem – realizacja sekwencji czasowych

W przedstawionym przykładzie sterowanie sygnalizatorem polega na cyklicznej zmianie świateł,

wg. sekwencji **Zielone 30s** ⇒ **Żółte 6s** ⇒ **Czerwone 20s** ⇒ **Czerwone/Żółte 4s** ⇒ Zielone 30s .....

Realizację programu rozpoczynamy od zaprogramowania sekwencji czasowej

**30s** ⇒ **6s** ⇒ **20s** ⇒ **4s** ⇒ 30s ⇒ ...

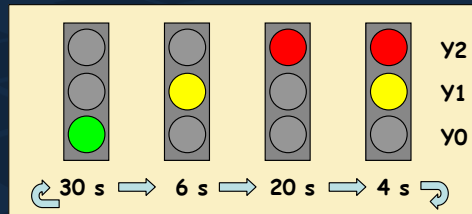
Do tego celu wykorzystano cztery timery **T1**, **T2**, **T3**, **T4** – stałe odpowiadają żądanym czasom sekwencji.

Timery uruchamiają się kolejno, tzn. **T1** uaktywnia **T2**, **T2** uaktywnia **T3**, **T3** uaktywnia **T4**. Zamknięcie pętli następuje przy pomocy styku **T4** (normalnie zamkniętego). Po odliczeniu 4s, zamyka się styk **T4** (styk normalnie zamknięty otwiera się) co powoduje przerwanie obwodu zasilania **T1** (zerowanie **T1**) a dalej zerowanie **T2**, **T3**, **T4** i cykl rozpoczyna się od początku. Pokazano to na przebiegach czasowych.

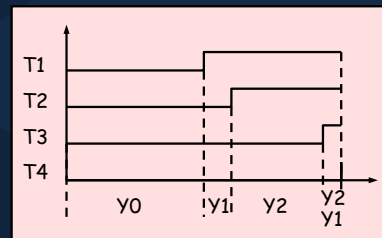
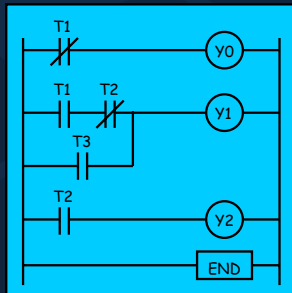
W ogólnym przypadku cykl można sterować również momentem rozpoczęcia cyklu, liczbą pętli cyklu itp.



## Przykład programu z realizacją sekwencji czasowych



		Y1				
		T3T4	00	01	11	10
T1T2	00	0				
	01					
	11	0			1	
	10	1				



Zygmunt Kubiak

Sterowniki PLC (28)

### Przykład sterowania sygnalizatorem – realizacja sekwencji czasowych

W drugim etapie tworzenia programu określamy warunki (funkcje) jakie muszą być spełnione aby dane wyjście było aktywne (światło zapalone). Sposób postępowania pokazano dla wyjścia Y1 (światło żółte).

$$Y1 = (T1 \text{ AND NOT } T2) \text{ OR } T3$$

Analogicznie postępujemy w celu określenia funkcji dla pozostałych wyjść.

Zauważmy – w programie każde wyjście występuje tylko raz.



**Wbudowane funkcje HMI**  
Przyjoma obsługa oraz bardzo przejrzyste wyświetlanie są głównymi punktami sterownika ALP11A XL. Ten duży ekran z podwójnym trybem obsługi się kilkoma nowymi możliwościami, jak wyświetlenie słupkowe czy też przesuwanie tekstu. Tryb pomiarowy zabezpiecza hasłem ogranicza dostęp oraz możliwość wyłączenia pomiarów programu, na co po-

zwala 8 wbudowanych przycisków funkcyjnych.

**Proste programowanie**  
Kilka sterowanych tych zostało dodanych do bardzo popularnego pakietu oprogramowania graficznego ALP-PCS-WIN. Na przykład nowy Display Manager, który przyspiesza i upraszcza konfigurację menu systemu wyświetlania, czy też nowa, bardzo użyteczna biblioteka funkcji użytkownika.

8 wbudowanych wejść analogowych oraz możliwość rozszerzenia sterownika zwiększa elastyczność systemu.

Każde z otoczonych jednostek centralnych może zostać rozszerzona o jeden z czterech dostępnych adresów.

Możliwość pracy w niskich temperaturach do -25 C

Duży, podświetlany wyświetlacz LCD, dodatkowe funkcje HMI

Praca 16 jednostek w sieci

Dodatkowe możliwości komunikacyjne, w tym e-mail i SMS

**Rozwiązania PCB (specjalne systemy sterowników)**

**Mini sterowniki z jednostkami sterującymi**

**Typowe systemy sterujące (przełączniki, liczniki czasu, liczniki zdarzeń)**

**Moduły logiczne**

**ALP11A XL**

Pamięć programu zwiększona do 300%

Dostępne funkcje matematyczne

**BCM**

Zygmunt Kubiak

Sterowniki PLC (29)

Rozwój sterowników PLC trwa od połowy lat sześćdziesiątych ubiegłego wieku. W tym czasie, nastąpił ogromny postęp technologiczny, sterowniki PLC przeszły drogę od prostych układów logicznych do złożonych, bardzo szybkich systemów mikrokomputerowych z arytmetyką zmiennoprzecinkową, z rozbudowanymi systemami komunikacyjnymi. Rosną moce obliczeniowe i możliwości sterowników. Spełniają one oczekiwania najbardziej wymagających użytkowników. Rynek PLC ciągle się rozwija i na razie nie widać przesłanek do zmiany tej tendencji.

Zakres zastosowań PLC jest bardzo szeroki od pojedynczych maszyn, np. pakujące, drukujące, wtryskarki, obrabiarki itp. poprzez gniazda produkcyjne, instalacje wodociągowe, klimatyzacyjne, wiatrowe, automatyzacje budynków, automatykę okrętową, systemy transportowe i logistyczne, zrobotyzowane linie montażowe po automatyzację kompletnych procesów technologicznych. Sterowniki PLC spotykamy w różnych branżach, przykładowo przemysł ciężki, chemiczny, spożywczy, drzewny, farmaceutyczny, samochodowy ciepłownictwo itd..

W ostatnich latach z grupy PLC wyodrębniła się nowa klasa urządzeń a mianowicie Programowalne sterowniki Automatyki (PAC - ang. Programmable Automation Controller), które oferują dodatkowe funkcje.

Slajd przedstawia zastosowania najprostszych sterowników PLC.



<b>Siemens</b>	<b>26 %</b>
<b>Schneider-Autom.</b>	<b>18 %</b>
<b>Allen-Bradley</b>	<b>17 %</b>
<b>GE Fanuc</b>	<b>8 %</b>
<b>Mitsubishi</b>	<b>11 %</b>
<b>Inne</b>	<b>20 %</b>



Najważniejsi producenci sterowników PLC