

///Sorki za błędy i niespójności językowe

1. SYSTEMY STEROWANIA

1.1. Podstawowe pojęcia

Pierwsze prymitywne, a potem bardziej skomplikowane maszyny, ułatwiły pracę człowiekowi. Był to okres mechanizacji, który jednak nie wyeliminował człowieka z procesów wytwarzania. Dopiero automatyzacja bazująca na doskonałych maszynach umożliwia odsunięcie człowieka od procesu wytwarzania.

Automatyzacja procesów technologicznych (np. walcowania, mielenia, syntezy chemicznej itp.) obejmuje:

- sterowanie, jest to postępowanie, które na podstawie mierzonych sygnałów i zbioru reguł sterowania (algorytmu) wypracowuje sygnał wyjściowy,
- zabezpieczenia i blokady,
- sygnalizację stanów normalnej i awaryjnej pracy,
- pomiar wielkości fizycznych charakteryzujących dany proces (zbieranie informacji),
- przekazywanie wyników na odległość (transmisja danych),
- przetwarzanie wyników na inną dogodną postać.

Automatyzacja bazuje na dziedzinie wiedzy, jaką jest automatyka. Potocznie „automatyką” nazywa się zestaw urządzeń technicznych umożliwiających zautomatyzowanie danego procesu (maszyny). Urządzenia noszą nazwę elementów automatyki, które mogą być połączone w rozmaity sposób tworząc łącznie z obiektem sterowania (np. maszyną, komorą chłodniczą,) systemy automatyki. Można wyróżnić następujące elementy automatyki:

- pomiarowe (czujniki, przetworniki, termostaty, presostaty itd.),
- zadające (nastawniki, panele sterujące, klawiatury itp.),
- regulatory, które mogą być jako autonomiczne lub zrealizowane komputerowo,
- wykonawcze (siłowniki, silniki, styczniki, grzałki, pompy itp.)

Obok elementów automatyki, drugim pojęciem nierozdzielnie związanym z systemem automatyki jest sygnał. Sygnałem nazywamy dowolną, wielkość fizyczną, niosącą, informację. W systemach automatyki mamy do czynienia z sygnałami:

- elektrycznymi,
- hydraulicznymi,
- pneumatycznymi.

W taki sam sposób w zależności od nośnika informacji, możemy podzielić systemy automatyki.

W systemie automatyki sygnały” można podzielić na dyskretne i analogowe.

Sygnały dyskretne przyjmujące dwa stany „1” (załączony, ON, obecność napięcia, przepływu prądu) lub „0” (wyłączony, OFF, brak napięcia, brak przepływu prądu), nazywane są również sygnałami binarnymi, dwójkowymi. Najmniejszą ilością informacji cyfrowej jest jeden bit. Taka ilość informacji jest przenoszona przez sygnał przyjmujący dwa kolejne stany. W technice napięciem sterującym jest najczęściej napięcie 24 V. Napięcie o takiej wartości, a dokładniej to i przedziału około 14 do 26 V. podłączone na wejście jest interpretowane jako stan „1” na tym wejściu. Odpowiednio 0V a dokładniej to z przedziału 0 do 5 V. będzie interpretowane jako stan „0”

Zestawienie w uporządkowany sposób bitów zgodnie z wcześniejszą umową (określonym kodem) tworzy tzw. wektor informacji cyfrowej. Zestawienie n-bitów pozwala zapisać 2^n wartości. Zestawienie 8 bitów nazywane jest bajtem i pozwala zapisać 256 wartości tzn. od 0 do 255. Większość sterowników korzysta z zestawu 16 bitów, czyli dwóch bajtów nazywanych słowami. Stosuje się również słowo o podwójnej długości czyli o 32 bitach. Numeracja bitów np. w słowie jest od prawej do lewej strony i rozpoczyna się od numeru 0. bit ten ma nazwę najmniej znaczącego bitu (ang. LSB Least Significant Bit) a kończy się na

bicie 15. który ma nazwę najbardziej znaczącego bitu (ang. MSB Most Significant Bit). Ponadto stosuje się przedrostki dla powyższych jednostek: kilo (k lub K -1024). mega (M=1024x1024=16 777 216) i giga (G I 073 741 824).

W większości aplikacji dominują sygnały dyskretne. natomiast trudno sobie wyobrazić zaawansowane systemy automatyki bez sygnałów analogowych. Sygnały te mogą przyjmować dowolne wartości z określonego przedziału np. 0÷10 V lub 4÷20 mA, Zastosowanie w systemie sygnału analogowego wymusza użycie specjalnych modułów analogowych, które w istocie są przetwornikami analogowo/cyfrowymi lub cyfrowo/analogowymi. Sygnał analogowy przetwarzany jest na sygnał cyfrowy (wektor). Im ten wektor ma większą liczbę bitów, tym dokładność przetwarzania jest większa. Liczba tych bitów określa rozdzielczość przetwornika. W większości modułów stosuje się przetworniki 11- lub 12- bitowe, co pozwala uzyskać odpowiednio 2048 lub 4096 wartości w ramach przedziału zmienności sygnału (zakresu). Dla sygnału wejściowego 0÷10 V dla przetwornika 11- bitowego zmiana napięcia o 4,88 mV powoduje zmianę wartości cyfry o 1. Występujący tu wektor informacji cyfrowej jest zakodowany. Kodowanie jest przyporządkowaniem każdemu elementowi w zbiorze symboli (cyfr, liter, znaków) jednego i tylko jednego wektora informacji cyfrowej. Obok kodu alfanumerycznego ASCII (ang. American Standard Code for Information Interchange) zawierającego 26 małych liter i 26 dużych liter alfabetu łacińskiego oraz 10 cyfr i około 25 symboli specjalnych. podstawowymi kodami są kody liczbowe.

Najpopularniejszym jest naturalny kod binarny (NKB) bazujący na dwóch elementach (0, 1); liczba elementów jest równa podstawie systemu ($p=2$). natomiast w wektorze informacji cyfrowej składającym się z k -bitów, każdej pozycji przyporządkowana jest „waga wynikająca z numeru pozycji i podstawy systemu a wynosi ona p^k . W NKB LSB będzie miał wagę $2^0=1$, natomiast 5 bit będzie miał wagę 2^{32} . Powyższa zależność w łatwy sposób umożliwia określenie wartości liczby zapisanej w NKB. W tym celu wystarczy dodać do siebie wagi bitów będących jedynekami np. liczba binarna 100011 to liczba $2^5+2^1+2^0=35$.

Popularnym kodem będącym skróconą formą NKB jest kod heksadecymalny (szesnastkowy). często oznaczany na końcu cyfry literką H. Bazuje on na szesnastu elementach (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F) czyli podstawą jest $p=16$, wzór do obliczenia wagi pozycji k jest taki sam, więc np. liczba heksadecymalna COO1H to liczba $C \times 16^4 + 0 \times 16^3 + 0 \times 16^2 + 1 \times 16^1 = 4097$.

Ostatnim powszechnie używanym kodem jest BCD (ang. Binary Coded Decimal). Jest to kod, w którym każda cyfra dziesiętna jest przedstawiona za pomocą czterech cyfr binarnych np. 87 w kodzie BCD to 10000111.

1.3. Komputerowe systemy sterowania

Procesy przemysłowe mogą mieć charakter [2]:

- ciągły,
- szarżowy, okresowy (ang. batch),
- testowy.

Eksploatacja ciągła jest używana w systemach, gdzie produkcja danego wyrobu trwa bez przerwy przez okres kilku miesięcy, a nawet lat i nie wymaga istotnych przestrojeń parametrów. Przykładem może być zakład przeróbczy, w którym po wykonaniu rozruchu (czyli jednorazowym ustawieniu parametrów) przez cały czas kopalina poddawana jest procesowi wzbogacania i sortowania.

Eksploatacja szarżowa albo okresowa wykorzystywana jest w systemach, gdzie produkt poddawany jest kilkukrotnemu przetworzeniu na tej samej instalacji. Wykonanie kolejnego przetworzenia produktu wymaga zmiany programu, a czasem narzędzi). Stąd czas na

wykonanie danego asortymentu składa się z czasu operacyjnego oraz czasu potrzebnego na zmiany (ang. set-up, down time, changeover time). Przykładem może być wiele instalacji w przemyśle spożywczym, farmaceutycznym oraz chemicznym.

Eksploatacja testowa używana jest wtedy, gdy komputer wykorzystywany jest do wykonywania pewnych złożonych, rutynowych testów inicjowanych przeważnie przez operatora. Wyniki testów są podstawą do wyboru sterowania.

Wobec powyższego, sterowanie jest to świadome oddziaływanie na proces, a dokładniej na wejścia systemu, aby zrealizować określony cel w ramach narzuconej dokładności sterowania (patrz rozdz. 3.5 - wskaźniki jakości sterowania). Natomiast narzędziami do realizacji tego sterowania są systemy, w których głównymi elementami systemu sterującego są komputery i stąd nazwa komputerowe systemy sterowania.

W komputerowych systemach sterowania stosuje się następujące rodzaje sterowania:

- sekwencyjne,
- wielozadaniowe (ang. multi-tasking),
- w czasie rzeczywistym (ang. real-time).

W sterowaniu sekwencyjnym kolejne ruchy (stany pracy) są uporządkowane i uzależnione od wcześniejszych stanów ich kolejności. Czas potrzebny na wykonanie ruchu nie odgrywa roli. Pewne fragmenty układu mogą wykorzystywać algorytmy regulatorów PID. W sterowaniu wielozadaniowym program składa się z kilku części, z których każda jest sekwencyjna, lecz całość realizuje określone zadanie i jest wykonywana współbieżnie (ang. concurrently) przez odpowiednie procedury synchronizujące. Również i w tych programach nie jest konieczne powiązanie z czasem rzeczywistym



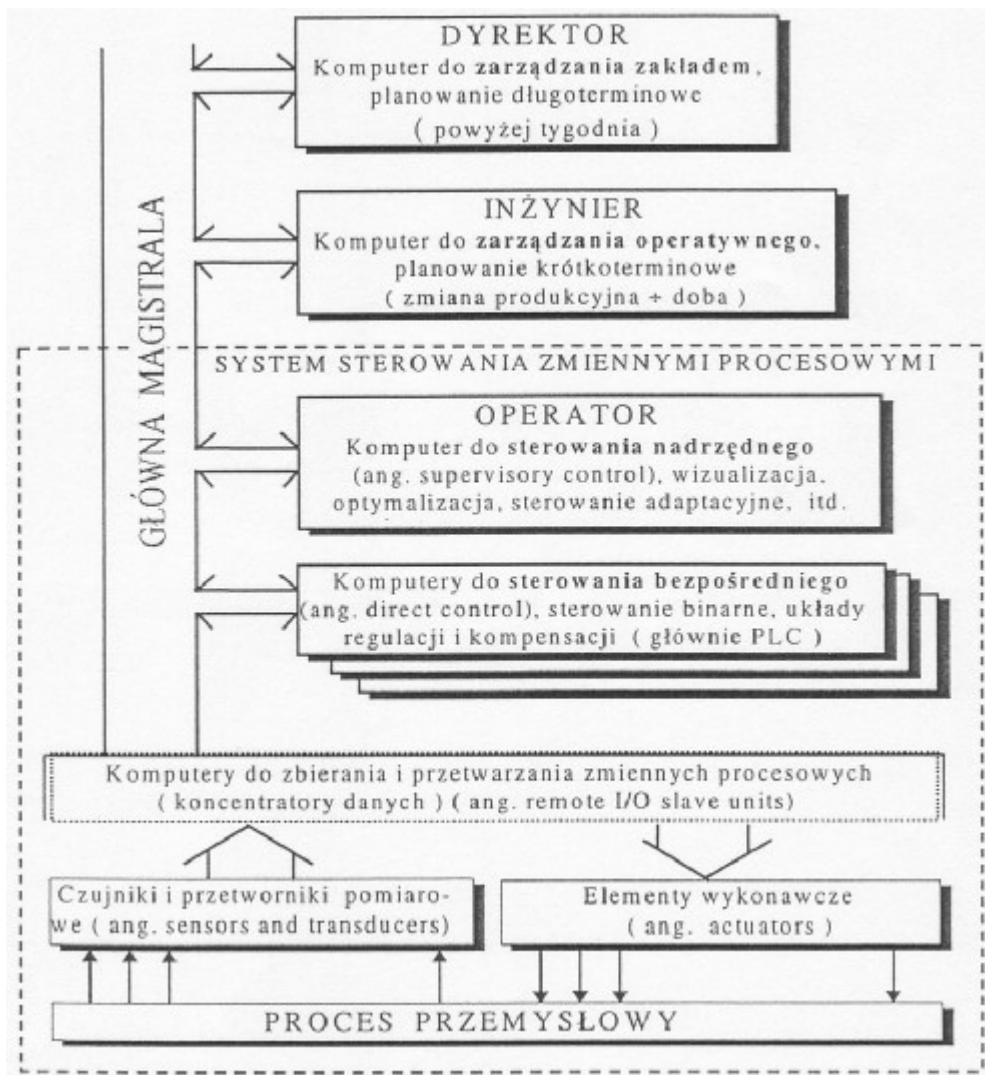
Rys. Piramidalny układ warstw sterowania

Przy programowaniu w czasie rzeczywistym sekwencje ruchów nie są określane przez projektanta lecz przez środowisko. Możemy wyróżnić trzy rodzaje systemów czasu rzeczywistego:

- a) czasowe, które bazują na zegarze sterującym przerwami,
- b) sprzętowe, które bazują na czujnikach reagujących na pewne wydarzenia i uruchamiają kolejne sekwencje procesu wykorzystując przerwania,
- c) interakcyjne np. automatyczny kasjer, system rezerwacji. Cechuje je określona wartość czasu (np. 20 s) na odpowiedź i udzielanej przeważnie człowiek.

Pod względem struktury komputerowe systemy sterowania można podzielić na:

- scentralizowane (przeważnie jednokomputerowe),
- rozproszone, zdecentralizowane (wielokomputerowe) (ang. distributed computer control).



Rys. System hierarchiczny i rozproszony

Bezsporna przewaga systemów rozproszonych wynikająca z bardzo niskich kosztów sprzętu powoduje, że systemy te są szeroko stosowane. Główne założenia dla tych systemów to: każdy komputer jest w stanie realizować zadania jak inne komputery, w przypadku awarii istnieje możliwość wykorzystania sąsiedniego komputera. Wymaga to szerokopasmowej magistrali do przenoszenia wszystkich informacji. Dodatkową cechą systemów rozproszonych jest ich hierarchiczność (wielopoziomowość). W licznych materiałach reklamowych struktura takiego systemu rysowana jest w kształcie piramidy (rys. 1.4). Schemat ogólny systemu zdecentralizowanego i wielopoziomowego przedstawiono na rys. 15. Najniżej znajduje się warstwa sterowania bezpośredniego z największą liczbą komputerów, przeważnie PLC, bezpośrednio połączonych z obiektem lub przy użyciu koncentratorów. Nad nią jest warstwa sterowania nadrzędnego, w której można wyróżnić: warstwę sterowania sekwencyjnego np. ciągiem maszyn oraz warstwy adaptacji i optymalizacji. Na wierzchołku wspomnianej piramidy są warstwy: zarządzania operatywnego (produkcją) i zarządzania zakładem.

1.4. Komputery i ich programowanie

System komputerowy składa się ze sprzętu (ang. hardware) i oprogramowania (ang. software). Pod pojęciem sprzętu rozumie się komputery wraz z urządzeniami peryferyjnymi. Dzielimy je na:

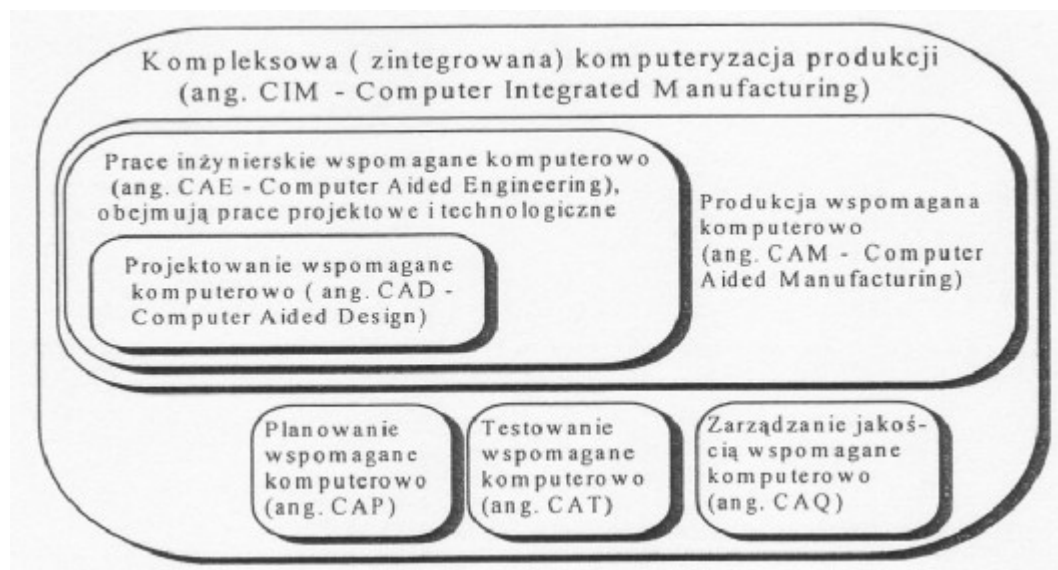
- domowe (osobiste); są to małe komputery, które można połączyć z telewizorem, magnetofonem i można nimi sterować radiem, telewizorem, kuchenką elektryczną, temperaturą w pokoju, itp.
- stacje robocze (ang. work station); są to stanowiska pracy wyposażone w sprzęt komputerowy do realizacji określonych zadań. Mogą być autonomiczne lub terminalami większego systemu komputerowego,
- duże systemy komputerowe (superkomputery),
- sterowniki jednopłytkowe, przeważnie budowane przy użyciu procesorów jednoukładowych,
- sterowniki przemysłowe charakteryzują się w większości budową modułową i specjalnymi zabezpieczeniami przed szkodliwością środowiska.

W zależności od potrzeb w sterownikach (ang. controllers) stosuje się cyfrowe sterowanie:

a) w układzie otwartym, realizujące funkcje logiczne zastępujące dotychczasowe układy stycznikowe. Ich rozwiązania sprzętowe noszą nazwę programowanych sterowników przemysłowych (ang. PC - Programmable Controllers, PLC - Programmable Logic Controllers), stosowana jest również nazwa sterowniki sekwencyjne.

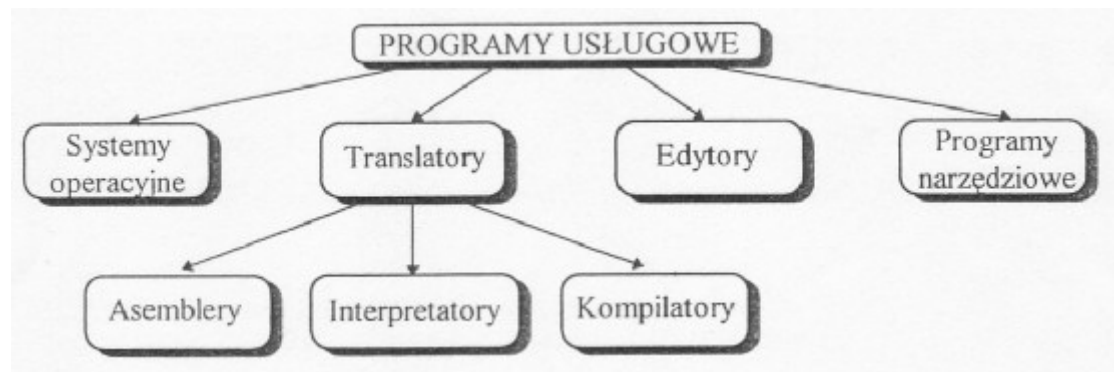
a) w układzie zamkniętym, stosowane w procesach ciągłych, co szczegółowo omówione zostało w rozdziale 3, dotyczącym bezpośredniego sterowania cyfrowego. Wszystkie czołowe firmy produkujące sterowniki PLC wyposażają je w moduły do sterowania w układzie zamkniętym.

Systemy komputerowe wykorzystuje się wg następujących potrzeb



Oprogramowanie można podzielić na programy usługowe (rysi.?), w których stosuje się polecenia (ang. command) i programy użytkowe, które przy dużej liczbie plików noszą nazwę pakietów np.: ACAD, MATLAB itp.

Do napisania powyższego oprogramowania wykorzystuje się języki. Program źródłowy (ciąg instrukcji lub rozkazów) napisany w danym języku jest tłumaczony (przekształcony) na program wynikowy, docelowy (język maszynowy) przy pomocy translatora danego języka.



Rys. Podział programów usługowych.

Języki dzielimy na:

a) maszynowe (wewnętrzne, kody maszynowe, listy rozkazów procesorów) (ang. machine code, machine language). Język ten rozumiały jest tylko dla komputera, gdyż rozkazy, adresy, dane itd. są zapisywane w postaci binarnej. Dlatego język ten nie jest stosowany do tworzenia programów.

b) Symboliczne:

- zorientowane systemowo (montujące, niższego poziomu, języki assemblera); jednej instrukcji odpowiada jeden rozkaz (ang. instruction), np. język assemblera systemu procesora INTEL 80486,

- zorientowane proceduralnie o ogólnym przeznaczeniu [23], (wyższego poziomu): jednej instrukcji (ang. statements) odpowiada wiele rozkazów np.: 13A51C, PASCAL, ADA, C,

- zorientowane aplikacyjnie (problemowo [23], zadaniowo) np.: COBOL - do zagadnień ekonomicznych i biurowych, CUTLASS - do sterowania cyfrowego, LISP, PROLOG - do zagadnień z zakresu sztucznej inteligencji, STEP 7 - do programowania sterowników.

Programowanie jest to metodyka opracowywania, organizacji i pisania programów komputerowych. Rozróżniamy programowanie strukturalne (ang. structured programming) oparte na następujących konstrukcjach językowych (rys. 1,8) [17]:

- sekwencja działań,

- decyzja IF...THEN...ELSE (jeżeli...to. . . w przeciwnym razie); służy do tworzenia rozgałęzień,

- wybór,

- powtórzenie DO...,WHILE, REPEAT..UNTIL (wykonuje się je do chwili spełnienia warunku); wykorzystywana jest do tworzenia pętli.

Program jest sekwencją powyższych konstrukcji językowych.

Obecnie w fazie wielkiego rozwoju jest programowanie współbieżne (ang. concurrent programming). Niektóre języki mają współbieżność jako część języka np.: ADA, MODULA 2. Zbudowane przy jego użyciu systemy operacyjne oraz systemy czasu rzeczywistego pozwalają przełączać pracę komputera pomiędzy kilkoma obliczeniami (wieloprogramowość) lub kilkoma komputerami (wieloprocessorowość) przy równoczesnym rozwiązaniu problemów związanych z synchronizacją i komunikacją.

Programy tworzymy według następującego schematu (rys. 1.9). Pełna analiza problemu pozwala napisać algorytm, który jest sekwencją ściśle zdefiniowanych operacji lub metod postępowania, podejmowanych w celu rozwiązania określonego zadania w skończonej liczbie kroków. Tworzymy go przy znanych podprogramach od szczegółu do ogółu (ang. bottom up), a przy nieznanym problemowi od ogółu do szczegółu (ang. top down).



Graficznie program zapisujemy wykorzystując sieci działań (ang. flow diagram, programming flowchart) lub diagramy strukturalne (rys. 1.8). Diagramy strukturalne, chociaż mniej popularne wymuszają jednak przejrzystą strukturę i nie dopuszczają do tworzenia nieprawidłowych pętli [2]. Do definiowania sposobu zapisu instrukcji w opisach języków stosuje się trzecią formę tzw. diagramy składniowe.

Oprogramowanie po napisaniu musi być:

- jednoznaczne tzn. wielokrotne zastosowanie dla tych samych danych daje identyczne wyniki,
- kompletne tzn. uwzględnia wszystkie przypadki, jakie mogą zaistnieć podczas jego wykonywania.

1.5. Oprogramowanie zorientowane aplikacyjnie

Oprogramowanie zorientowane aplikacyjnie (ang. application - oriented software) jest budowane w tym celu, aby po krótkim przeszkoleniu mógł z niego korzystać każdy inżynier znający daną technologię. Nie wymaga to zatrudniania na etacie informatyka oraz unika się wielu błędów podczas uruchamiania i przestrajania.

Oprogramowanie to można podzielić na [2]:

- pobudzone tablice (ang table-driven),
- oprogramowanie blokowo - strukturalne.
- specjalizowane języki (języki symboliczne zorientowane aplikacyjnie).

Pobudzone tablice często korzystają z wcześniej napisanych programów w języku assemblera, aczkolwiek mogą to być również programy napisane w językach wyższego poziomu przystosowane do współpracy z pobudzonymi tablicami. Decydującym elementem świadczącym o przydatności tego oprogramowania jest metoda zadawania i modyfikowania danych w tablicy. Można tu wyróżnić trzy podejścia:

- bezpośrednie wprowadzanie danych z klawiatury (języki klawiaturowe), gdzie poszczególnym klawiszom przyporządkowane są instrukcje, które są bezpośrednio związane z daną technologią np.: wciśnięcie klawisza „PRĘDKOSC POWIETRZA” i klawisza „8” oznacza w danej technologii, że ustawiono prędkość przepływu powietrza 8 m/s, klawisz „NAND” oznacza, że zastosowano operację logiczną typu NAND, itp.,
- wypełnianie blankietów (języki blankietowe) polegające na wprowadzaniu danych lub modyfikacji uaktywnionych okienek na ekranie,
- używanie instrukcji DATA stosowanej głównie w dużych systemach.

Oprogramowanie typu pobudzone tablice jest bardzo proste i bezpieczne w użyciu, ale nieelastyczne, gdyż system jest konfigurowany podczas uruchamiania, między innymi co do liczby zmiennych i użytkownik nie może go już później zmienić.

Tej wady pozbawione jest oprogramowanie blokowo - strukturalne. Zawiera ono bibliotekę funkcjonalnych bloków (np.: regulator PID, oprogramowanie standardowe (ang. routines) dla wyjść, funkcje arytmetyczne, skalowanie bloków, oprogramowanie monitora) oraz oprogramowanie nadrzędne do manipulowania pomiędzy blokami. Inżynier wybiera odpowiednie bloki i łączy w zależności od potrzeb. Realizowane jest to przy użyciu odpowiednich pakietów na monitorach ekranowych przeważnie o podwyższonej rozdzielczości. Przykładem tego oprogramowania jest SIMULINK.

Specjalizowane języki mają składnię syntaktyczną przystosowaną do programowania np. określonego sprzętu w tym przypadku sterowników przemysłowych. Szeroko omówione one będą w rozdziale szóstym.