

Jakub Piotrowski 266502

Mateusz Bogusz 266492

Sprawozdanie

1. Wstęp

Celem naszego projektu było zaprogramowanie sterownika programowalnego PLC Siemens LOGO! w celu symulacji procesu wjazdu samochodu do garażu. Realizacja tego celu została przeprowadzona poprzez dwa podejścia: tradycyjne programowanie logiczne bez użycia przerzutnika typu RS oraz podejście z wykorzystaniem przerzutnika typu RS. Celem analizy było nie tylko opracowanie funkcjonalnych symulacji, ale także porównanie obu metod pod względem efektywności, zużycia zasobów, prostoty i odporności na awarie.

Programowanie Logiczne bez użycia diagramu SFC polegało na bezpośrednim sterowaniu zachowaniami systemu poprzez użycie standardowych funkcji logicznych i sekwencji operacyjnych. To podejście wymagało szczegółowej analizy i predykcji różnych stanów systemu oraz uwzględnienia stanów pośrednich.

Zastosowanie diagramu SFC pozwoliło na modelowanie procesów jako sekwencje kroków i przejść, zapewniając wyższy poziom abstrakcji oraz upraszczając zarządzanie złożonymi warunkami i przejściami pomiędzy stanami systemu.

Obie metody przeszły testy symulacyjne w oprogramowaniu LOGO!Soft Comfort oraz testy praktyczne na rzeczywistej makiecie garażu.

2. Opis rzeczywistości

Przyjęte oznaczenia to:

I1 – auto na podjeździe

I2 – brama otwarta

I3 – auto w garażu

I4 - brama zamknięta

Q1 - włącz oświetlenie podjazdu

Q2 – otwieraj bramę

Q3 – zamykaj bramę

Q4 - włącz oświetlenie garażu

Symbol "I" oznacza czujniki, a "Q" reprezentuje działania, jakie mają być wykonane w określonych warunkach, opisane szczegółowo w tabelach logicznych poniżej.

| I1 | I2 | I3 | I4 | Q1 | Q2 | Q3 | Q4 |
|----|----|----|----|----|----|----|----|
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |

Tabela 1. Tabela logiczna reprezentująca podejście bez wykorzystania diagramu SFC.

| I1 | I2 | I3 | I4 | Q1 | Q2 | Q3 | Q4 |
|----|----|----|----|----|----|----|----|
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |

Tabela 2. Tabela logiczna reprezentująca podejście wykorzystujące diagram SFC.

3. Wyniki

Podjęcie bez wykorzystania diagramu SFC wymagało szczegółowego planowania i kompleksowej analizy potencjalnych scenariuszy. Wykorzystaliśmy 22 z 200 dostępnych bloków funkcyjnych, a program zużywał 232 z 3800 dostępnych bajtów pamięci. Podjęcie z diagramem SFC pozwoliło zredukować liczbę wykorzystywanych bloków funkcyjnych do 4 oraz zużycie pamięci do zaledwie 32 bajtów.

Obie strategie skutecznie symulowały proces wjazdu samochodu do garażu i zachowywały się prawidłowo po przerwach w zasilaniu. Podjęcie z użyciem diagramu SFC wymagało jednak aktywacji funkcji „Retentivity” dla zapewnienia właściwego zachowania programu po przywróceniu zasilania.

4. Wnioski

Analiza przeprowadzonego projektu uwydatnia różnice w implementacji i zarządzaniu systemem sterowania w zależności od wybranej metody programowania sterownika PLC Siemens LOGO!. Programowanie bez wykorzystania diagramu SFC, chociaż bardziej zrozumiałe i prostsze do wdrożenia w ograniczonych aplikacjach, wykazuje wyższe zużycie zasobów – zarówno pod względem wykorzystania bloków funkcyjnych, jak i pamięci. Ta metoda, mimo swojej bezpośredniości, może być niewystarczająca w przypadku systemów bardziej złożonych, gdzie wymagana jest większa elastyczność oraz możliwość łatwej modyfikacji czy rozbudowy.

Użycie diagramu SFC znacząco upraszcza proces programowania, zapewniając wyższy poziom abstrakcji i intuicyjne wizualizacje, co jest nieocenione w przypadku skomplikowanych lub dynamicznie zmieniających się systemów. Choć początkowo metoda ta wymaga większego nakładu czasu oraz specjalistycznej wiedzy, długoterminowe korzyści są znaczące. Oferuje ona zwiększoną efektywność operacyjną, mniejsze zużycie pamięci, a także redukuje liczbę wykorzystywanych bloków funkcyjnych. W rezultacie, możliwe jest zmniejszenie kosztów utrzymania oraz łatwiejsze wprowadzanie przyszłych modyfikacji.

Podczas testów obie strategie wykazały wysoką odporność na przerwy w zasilaniu, jednak podjęcie z użyciem SFC wymagało aktywacji funkcji „Retentivity” w celu zachowania ciągłości działania po przywróceniu zasilania. To podkreśla zdolność metody

SFC do zapewnienia niezawodności i ciągłości pracy systemów sterowania, co jest kluczowe w kontekście ich długoterminowego użytkowania.

Podsumowując, decyzja o wyborze metody programowania powinna być świadoma i dostosowana do specyfiki projektu. W przypadku prostych aplikacji, tradycyjne podejście może okazać się bardziej kosztowo efektywne i wystarczające. Natomiast złożone systemy, wymagające przyszłej rozbudowy lub modyfikacji, znacznie zyskują na implementacji diagramu SFC, mimo początkowo większych nakładów. Jego zalety w zakresie diagnozy, elastyczności i skalowalności stanowią solidną inwestycję w przyszłość, usprawniając zarządzanie złożonymi systemami sterowania.