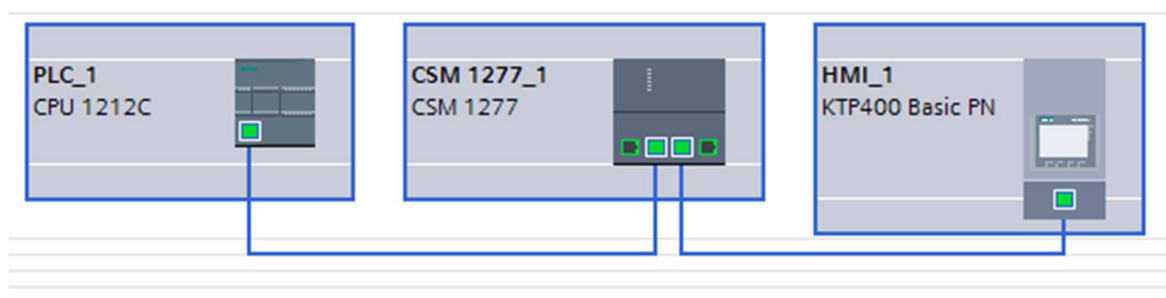


Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie Laboratorium Aparatury Automatyizacji			
Numer i temat ćwiczenia: Ćwiczenie 0. SIEMENS S7 1200 z panelem operatorskim – konfiguracja sprzętu			
Grupa ćwiczeniowa: Wtorek 17:00-19:15, Zespół: 3			
Lp.	Imię i nazwisko	Ocena	Podpis
1.	Katarzyna Wątorska		
2.	Sonia Wittek		
3.	Karolina Świerczek		
Data wykonania ćwiczenia: 12.03.2019			

1. Schemat i opis konfiguracji systemu:

Celem ćwiczenia było zapoznanie się ze środowiskiem TIA PORTAL poprzez przeprowadzenie podstawowej konfiguracji sterownika PLC SIEMENS S7 1200 z panelem operatorskim oraz uruchomienie na nim prostego programu sterowania logicznego, z wykorzystaniem aplikacji SCADA. Konfigurację rozpoczęliśmy od dodania do utworzonego w TIA PORTAL projektu sprzętu znajdującego się na stanowisku laboratoryjnym, tj. jednostki centralnej CPU 1212 C, switcha sieciowego CSM 1277 oraz panelu operatorskiego KTP400 Basic PN. Następnie elementy te odpowiednio skonfigurowaliśmy i połączyliśmy zgodnie z instrukcją, czego efekt przedstawia poniższy schemat:









Rys 1. Połączone elementy sterownika PLC.

2. Prosty program sterowania logicznego:

Stworzyliśmy algorytm sterowania logicznego, zaczynając od zdefiniowania nazw zmiennych w tabeli z nazwami symbolicznymi PLC tags oraz ich zaadresowania, w wyniku czego otrzymaliśmy:

PLC tags

PLC tags				
	Name	Data type	Address	Retain
	start	Bool	%M0.0	False
	gotow	Bool	%M0.1	False
	awaria	Bool	%M0.2	False
	stop	Bool	%M0.3	False
	naped ON	Bool	%Q0.0	False
	praca	Bool	%Q0.1	False

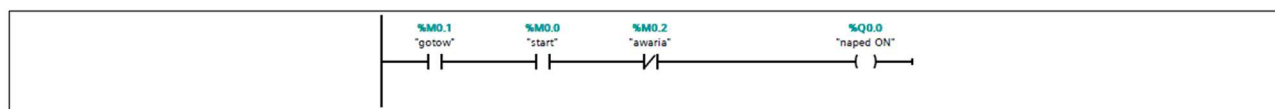
Rys 2. Zdefiniowane i zaadresowane zmienne.

Jako Data type użyliśmy dla wszystkich zmiennych bool, ponieważ wykorzystaliśmy je następnie do realizacji algorytmu logicznego, opisanego równaniami:

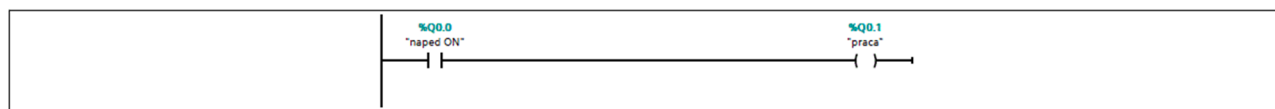
$$\begin{aligned} \text{naped ON} &= \text{gotow} \& \text{start} \& \text{NOT awaria} \\ \text{praca} &= \text{naped ON} \end{aligned}$$

Używając języka drabinkowego, zaprogramowaliśmy algorytm w pliku źródłowym bloku organizacyjnego OB1:

Network 1:



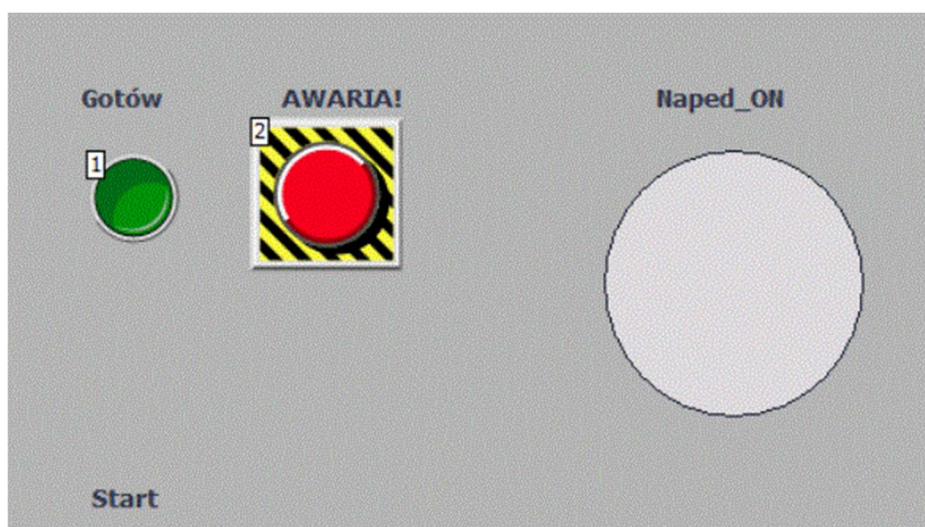
Network 2:



Rys 3. Algorytm sterowania zaprogramowany w języku drabinkowym.

3. Aplikacja SCADA na panelu operatorskim:

Aby zaobserwować działanie zaprogramowanego algorytmu w przejrzystej, graficznej formie, zdefiniowaliśmy na ekranie HMI możliwość zadawania wartości zmiennych gotow, start i awaria, czego wynik przedstawia poniższy zrzut ekranu:



Rys 4. Zrzut ekranu panelu operatorskiego.

Zmienną start powiązaliśmy z przyciskiem funkcyjnym F1, który w rzeczywistości znajdował się poniżej napisu „Start” widocznego na rysunku 4. Zmiennej gotow przypisaliśmy standardowy przycisk oznaczony na rysunku 4. numerem 1, zmiennej awaria natomiast przypisaliśmy tzw. grzybek bezpieczeństwa oznaczony numerem 2. Wyjście naped ON zostało powiązane z lampką sygnalizacyjną zbudowaną przez nas za pomocą podstawowego obiektu typu elipsa. Dodaliśmy do niego animację – wartość 0 zmiennej wyjściowej naped ON została skojarzona z kolorem czerwonym, a wartość 1 z kolorem zielonym. Następnie odpowiednio zdefiniowaliśmy podpisy do tych elementów. Sposób powiązania zmiennych z przyciskami obrazuje poniższy zrzut ekranu:

HMI_1 [KTP400 Basic PN] / HMI tags

Default tag table [4]

gotow

Name	gotow	Address		Connection	HMI_Connection_1
Data type	Bool	Length	1		

start

Name	start	Address		Connection	HMI_Connection_1
Data type	Bool	Length	1		

awaria

Name	awaria	Address		Connection	HMI_Connection_1
Data type	Bool	Length	1		

naped ON

Name	naped ON	Address		Connection	HMI_Connection_1
Data type	Bool	Length	1		

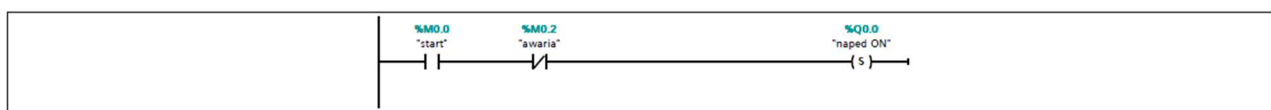
Rys 5. Zmienne wykorzystane przy tworzeniu HMI.

Następnym krokiem było wgranie algorytmu na sterownik i sprawdzenie jego działania za pomocą ekranu animującego wartości wyjścia na panelu operatorskim oraz przez obserwację w trybie online zachowania programu w tym samym oknie, w którym został on napisany.

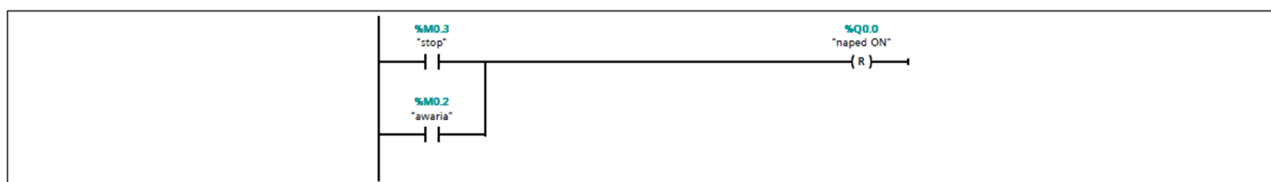
4. Alternatywna realizacja algorytmu sterowania:

Tą samą funkcję logiczną zrealizowaliśmy z wykorzystaniem cewki ustawiającej i kasującej. W tym celu przyjęliśmy założenie, że załączenie odbywa się z użyciem przycisku start, a wyłączenie z użyciem przycisku stop (powiązanego ze zmienną stop). Zmienną napęd ON powiązaliśmy przy starcie z cewką ustawiającą Set (S), a przy zatrzymaniu z cewką kasującą Reset (R). Analogicznie jak poprzednio, przeszliśmy przez etapy definiowania i adresowania zmiennych, pisania algorytmu sterowania w języku drabinkowym oraz dostosowywania panelu operatorskiego. Wyniki tych działań przedstawiają poniższe rysunki:

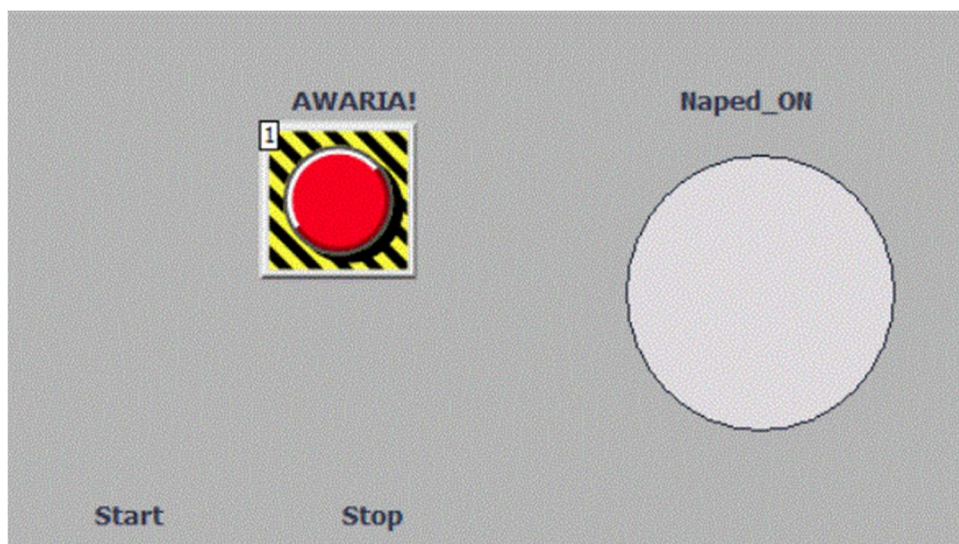
Network 1:



Network 2:



Rys 6. Algorytm sterowania zaprogramowany w języku drabinkowym.



Rys 7. Zrzut ekranu panelu operatorskiego.

HMI_1 [KTP400 Basic PN] / HMI tags

Default tag table [5]

gotow

Name	gotow	Address		Connection	HMI_Connection_1
Data type	Bool	Length	1		

Dynamizations\Event					
Event name	Value change				

Function list\SetBit					
----------------------	--	--	--	--	--

Tag	gotow				
-----	-------	--	--	--	--

start

Name	start	Address		Connection	HMI_Connection_1
Data type	Bool	Length	1		

awaria

Name	awaria	Address		Connection	HMI_Connection_1
Data type	Bool	Length	1		

naped ON

Name	naped ON	Address		Connection	HMI_Connection_1
Data type	Bool	Length	1		

stop

Name	stop	Address		Connection	HMI_Connection_1
Data type	Bool	Length	1		

Rys 8. Zmienne wykorzystane przy tworzeniu HMI.

5. Wnioski:

Na zajęciach nauczyliśmy się konfigurować sterownik PLC oraz uruchamiać na nim prosty, napisany przez nas algorytm. Podczas definiowania zmiennych w tabeli PLC tags, nauczyliśmy się poprawnie i efektywnie je adresować, z wykorzystaniem zapisu symbolicznego „%M” oraz „%Q”, odpowiednio dla zmiennych wewnętrznych oraz wyjściowych. Przyswoiliśmy sobie zasady programowania w języku drabinkowym, np. że iloczyn logiczny realizujemy jako szeregowe połączenie styków, a sumę realizujemy jako połączenie równoległe, zmienna wejściowa wprost to styk normalnie otwarty, zmienna zanegowana to styk normalnie zamknięty, zmienna wyjściowa wprost to cewka zwykła, zmienna wyjściowa zanegowana to cewka negująca, a każde wyjście to osobny szczebel w schemacie drabinkowym. Nabyliśmy również umiejętności związane z dostosowywaniem do danego zadania panelu operatorskiego i jego obsługą. Dzięki wykonaniu alternatywnej wersji algorytmu sterowania nauczyliśmy się podchodzenia do danego zagadnienia z różnych stron. Przy tym zaobserwowaliśmy także różnice w działaniu obu algorytmów, a mianowicie przy wersji pierwszej warunek musi być spełniony cały czas, w wersji drugiej natomiast należy załączyć jednym przyciskiem, a wyłączyć drugim.