

## Sprawozdanie z ćwiczenia 2 – UKŁAD REGULACJI TEMPERATURY

Bartosz Bryk, Maciej Kurcius, Jakub Piasek – zespół 3

### Opis ćwiczenia

Naszym zadaniem podczas zajęć było odpowiednie skonfigurowanie sterownika SIEMENS, tak aby dokonać pomiaru i regulacji temperatury w zbiorniku.

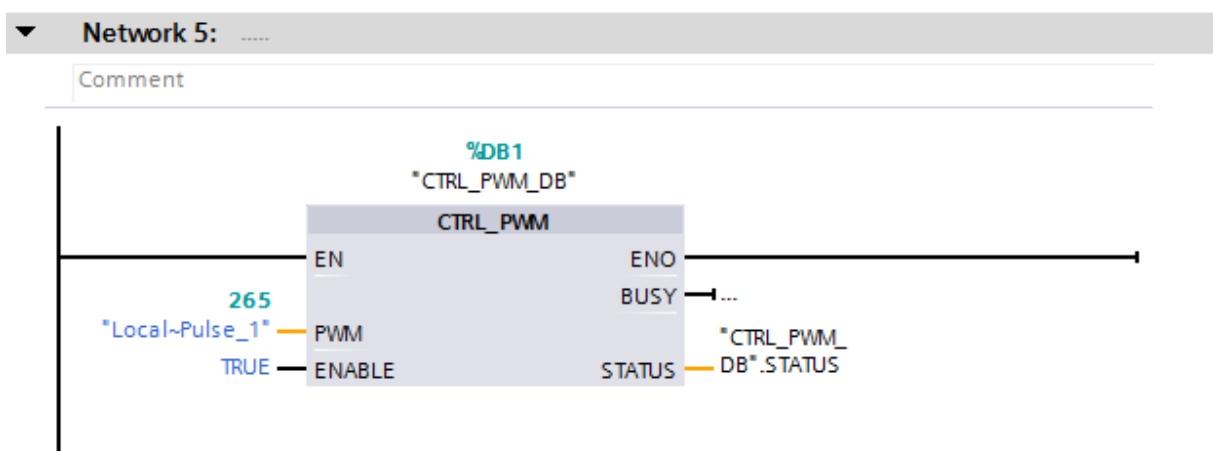
### Przebieg ćwiczenia

Ćwiczenie rozpoczęliśmy od wstępnej konfiguracji sterownika – tak jak w ćwiczeniu zerowym. Następnie zdefiniowaliśmy potrzebne w programie zmienne PLC Tags. Zmienne użyte w ćwiczeniu zaprezentowane zostały na zrzucie ekranu 1.

	Name	Tag table	Data type	Address	Retain	Visibl...	Acces...	Com
1	RTD_Input0	Default tag table	Word	%IW112		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
2	RTD_Input0_Tag	Default tag table	Int	%MW6		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
3	PMW_Output0	Default tag table	Int	%QW1000		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
4	Heater_Value	Default tag table	Int	%MW2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
5	PLO_Fan	Default tag table	Bool	%Q0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
6	Fan_ON	Default tag table	Bool	%M128.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
7	Manual	Default tag table	Bool	%M128.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
8	poziom_0	Default tag table	Int	%MW10		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
9	poziom_1	Default tag table	Int	%MW12		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
10	SP	Default tag table	Int	%MW14		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Zrzut ekranu 1 PLC Tags zdefiniowane przez nas

Następnie skonfigurowaliśmy wejście RTD oraz wejście PWM oraz skonfigurowaliśmy blok do obsługi PWM co pokazane jest na zrzucie ekranu 2.



Zrzut ekranu 2 Kontrola bloku PWM

W kolejnym etapie umieściliśmy w programie głównym bloki realizujące odczyt temperatury i przeliczenie jej na stopnie Celsjusza, a następnie stworzyliśmy funkcję, która spełniała rolę regulatora dwupołożeniowego bez histerezy. W funkcji tej użyliśmy zmiennych wewnętrznych zdefiniowanych tak jak na rzucie ekranu 3.

The screenshot shows the 'REGULATOR' function block with the following internal variables:

	Name	Data type	Default value	Comment
1	Input			
2	temp_zadana	Int		
3	temp_aktualna	Int		
4	ster_0	Int		
5	ster_1	Int		
6	Output			
7	sterowanie	Int		
8	InOut			
9	<Add new>			
10	Temp			
11	<Add new>			
12	Constant			
13	<Add new>			
14	Return			
15	REGULATOR	Void		

Below the table, the LAD logic is shown:

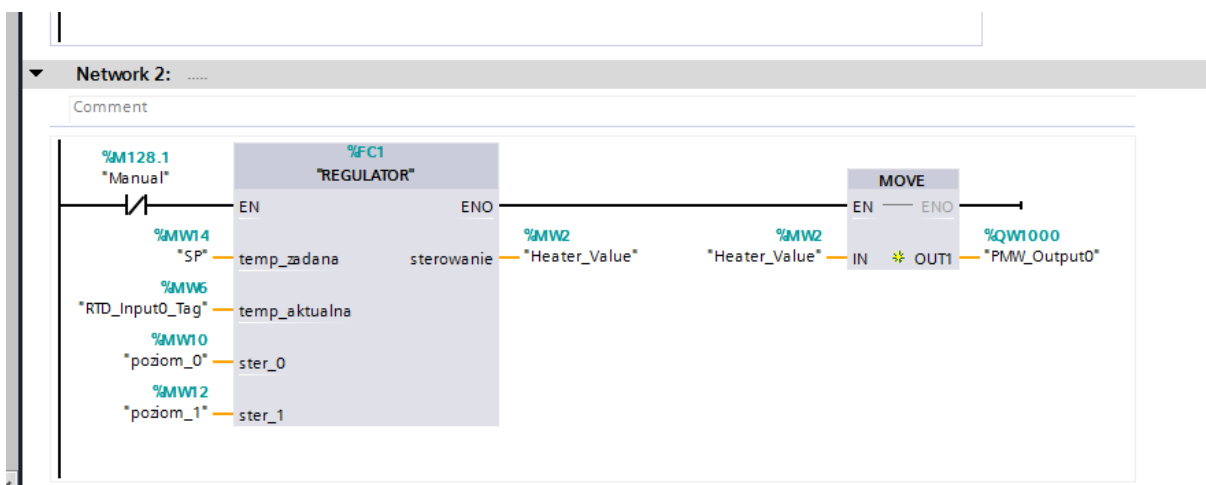
```

1 IF (#temp_aktualna - #temp_zadana) > 0 THEN
2   #sterowanie := #ster_0
3   ;
4 ELSE
5   #sterowanie := #ster_1
6   ;
7 END_IF;
8
9

```

Zrzut ekranu 3 Zmienne wewnętrzne funkcji

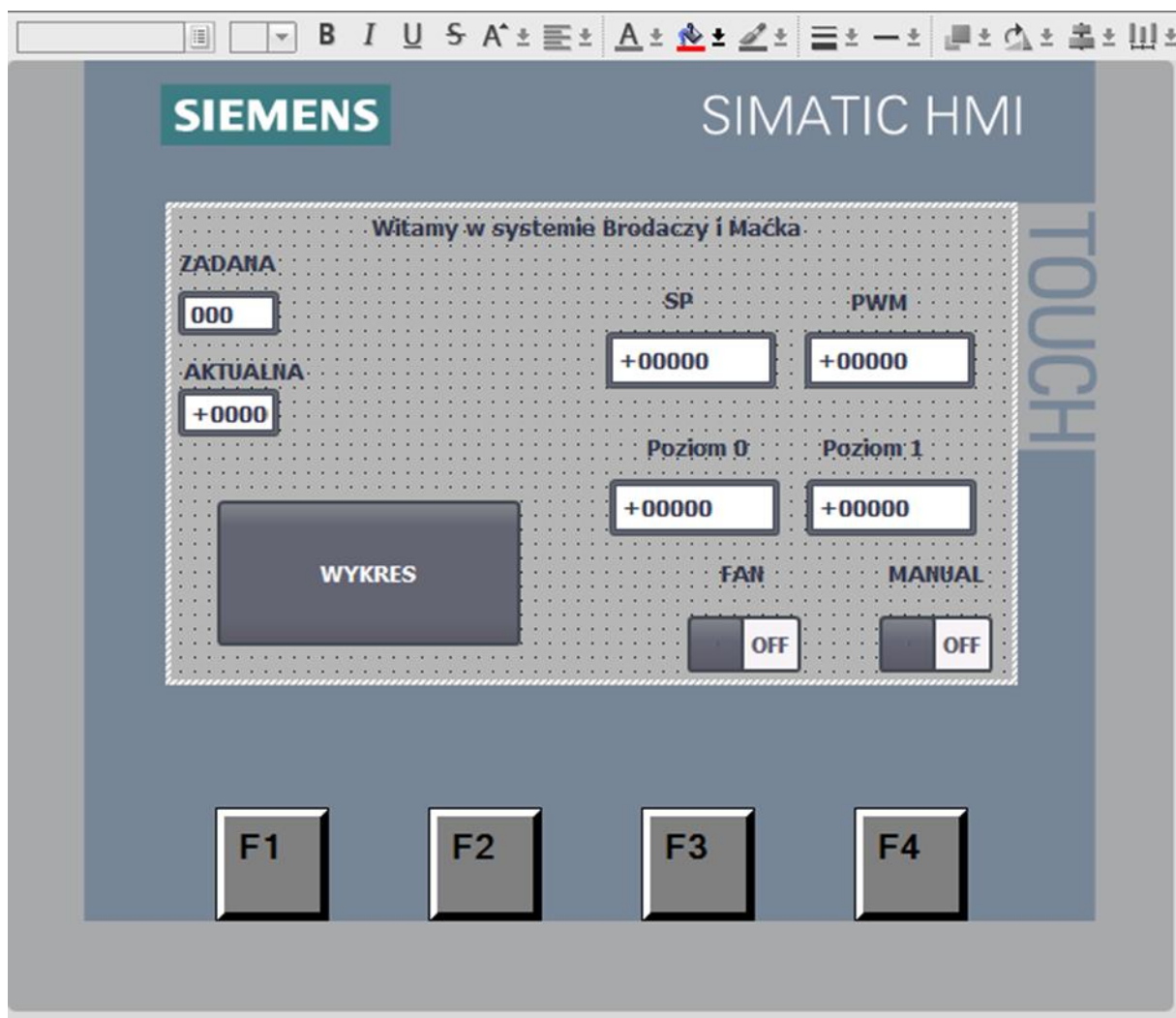
W programie głównym umieściliśmy blok realizujący tą funkcję pokazany na rzucie ekranu 4.



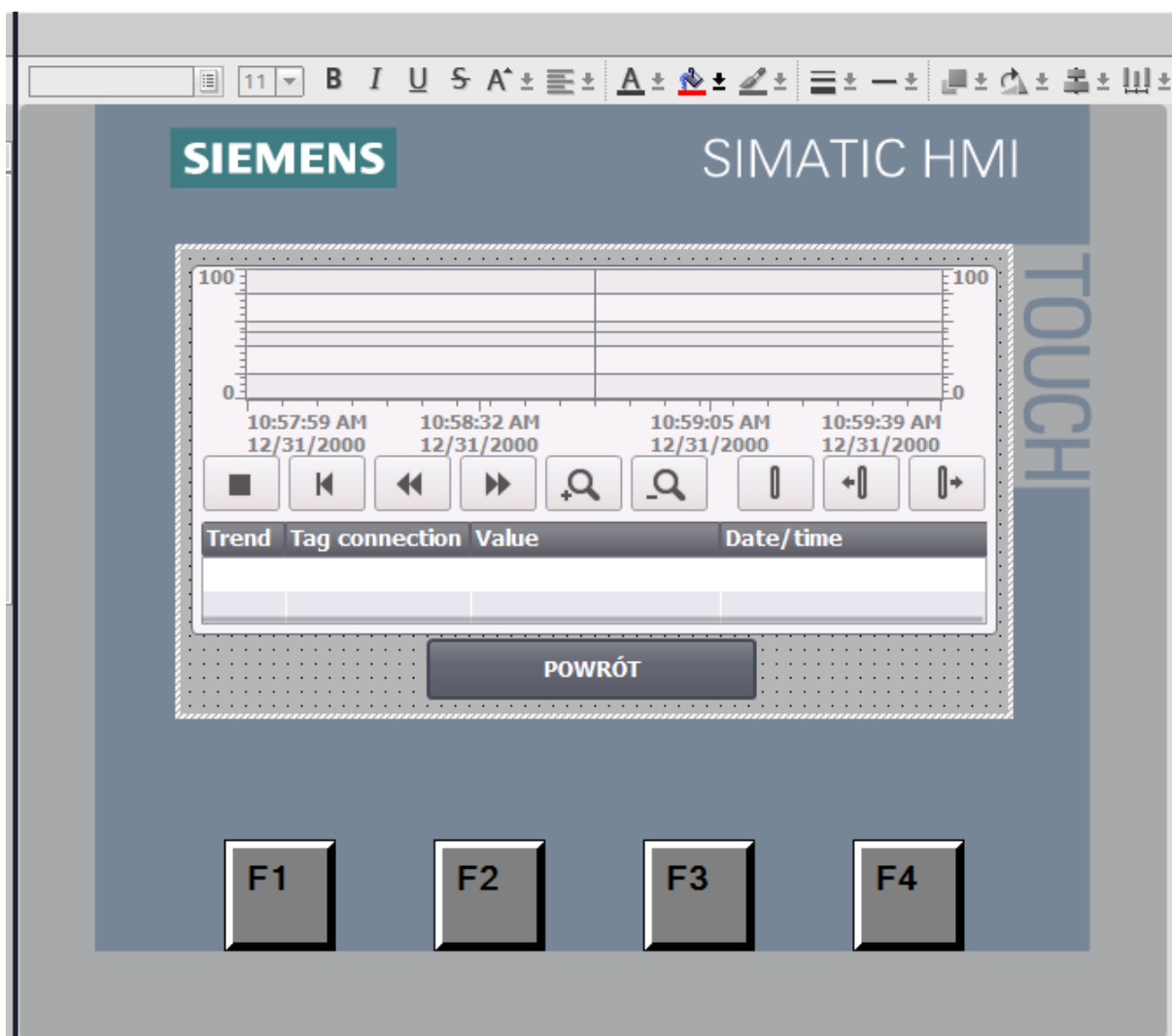
Zrzut ekranu 4 Funkcja regulatora 2 położeniowego bez histerezy w bloku głównym

W programie zamieściliśmy jeszcze dodatkowe funkcje takie jak np. ręczna kontrola wiatraka, czy możliwość manualnej regulacji wartości.

Na koniec utworzyliśmy panel sterowania SCADA w celu ręcznego sterowania procesem oraz panel umożliwiający odczyt wartości, pokazane odpowiednio na zrzucie ekranu 5 i 6. Przyciski WYKRES oraz POWRÓT umożliwiały przełączanie się pomiędzy poszczególnymi ekranami.



Zrzut ekranu 5 Panel sterujący SCADA



*Zrzut ekranu 6 Monitoring wartości w programie*

## Wnioski

Ćwiczenie udało nam się wykonać do końca. Program zadziałał poprawnie wraz z panelem operatorskim, który go obsługiwał. Poznaliśmy nowe funkcje portalu TIA portal oraz zapoznaliśmy się z bardziej zaawansowanymi technikami budowy panelu operatorskiego.