

Sprawozdanie z Laboratorium Aparatury Automatykacji			
Nr. ćw. Ćw. 4	Temat laboratorium Konfiguracja i uruchomienie sieci przemysłowej PROFINET (SIEMENS)		
Wydział EAIIB	Kierunek Automatyka i Robotyka	Rok ///	
Zespół Zespół nr 7	Grupa Grupa 1, piątek 8:00	Data 13 stycznia 2023	
L.p.	Skład grupy ćwiczeniowej		
1	Jakub Szczypek		
2	Iwona Fąfara		
3	Dawid Antosz		

## Spis treści

1. Wstęp .....	1
2. Opis stanowiska .....	1
3. Wykonanie ćwiczenia .....	3
4. Wnioski .....	9

## 1. Wstęp

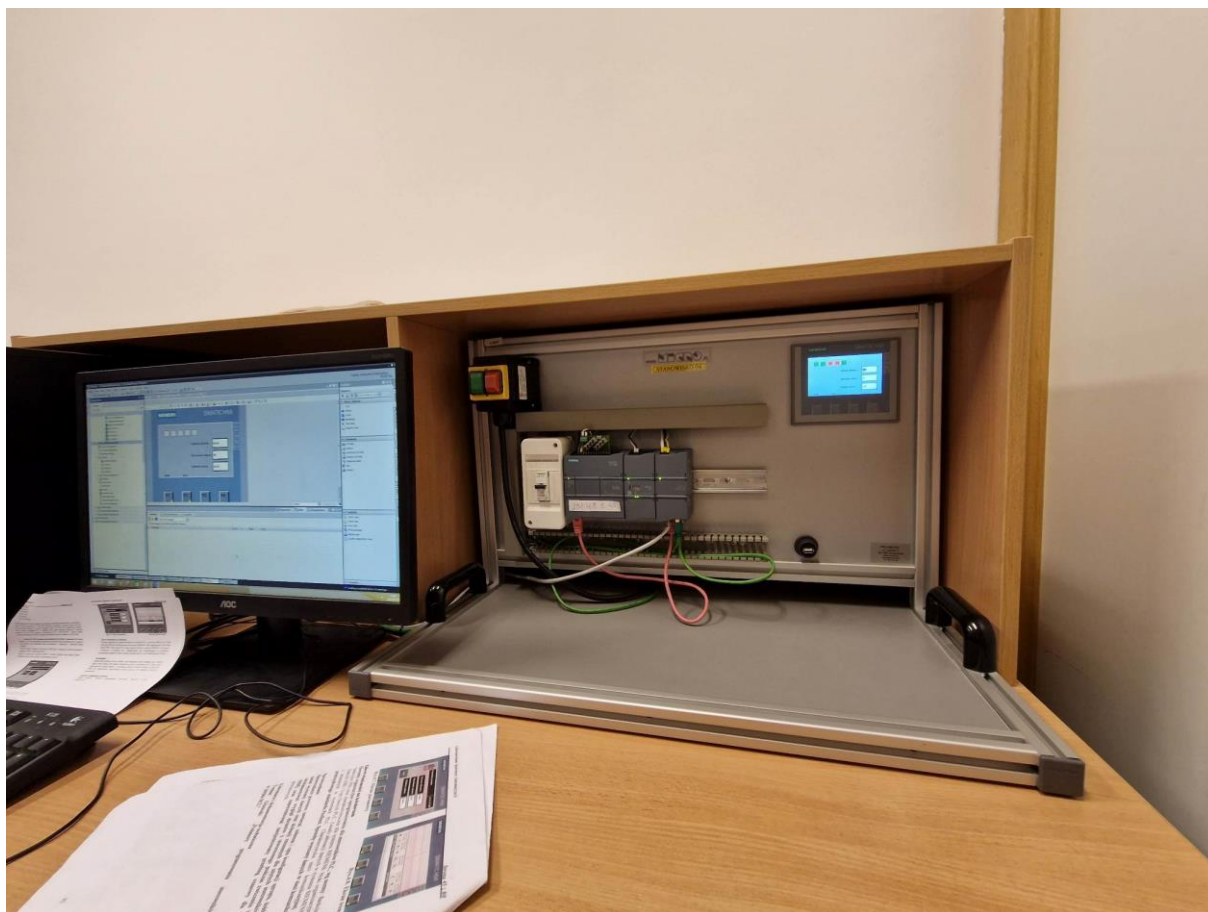
Celem ćwiczenia jest konfiguracja oraz uruchomienie rozproszonego systemu sterowania połączonego z użyciem sieci przemysłowej Profinet. Wykorzystaliśmy sterownik SIEMENS S7-1200 oraz panel operatorski do wizualizacji procesu automatycznej regulacji stanowiska do kontroli ciśnienia, wyposażonego w inny sterownik PLC tj. TURCK BL 20 PG EN V3 (użyty przez nas w ćwiczeniu nr. 1).

## 2. Opis stanowiska

Budowa stanowiska:

- Sterownik SIEMENS S7-1200
- Panel operatorski HMI
- Wyjścia analogowe AQ2x14BIT
- Sterownik TURCK BL 20
- Switch sieciowy CSM 1277 SIMATIC NET

Wygląd stanowiska przedstawiamy na rysunku poniżej:



**Rysunek 1.** Wygląd stanowiska laboratoryjnego

Profinet IO to sieć przemysłowa, powstała na podwalinach sieci Profibus DP oraz Industrial Ethernet, stanowiąc ich rozwinięcie. Najnowsze linie technologiczne i instalacje wykorzystują Profinet do komunikacji między urządzeniami systemu sterowania. Wykorzystując standard Profinet można przesyłać:

- Dane niekrytyczne czasowo – konfiguracja, parametry
- Dane krytyczne czasowo – komunikacja w czasie rzeczywistym RT
- Dane z synchronizacją czasu – komunikacja izochroniczna w czasie rzeczywistym IRT

Cykliczna wymiana danych jest komunikacją deterministyczną – posiada zdefiniowany cykl, w którym odbywa się wymiana danych. Dla sieci RT wymiana może odbywać się z cyklem 1,2,3,4,...,512 [ms] (najszybciej 1 [ms], najwolniej 512 [ms]). Dla trybu izochronicznego IRT częściej niż 1 [ms] wykorzystuje się: 500, 250, 125 [ $\mu$ s]. Urządzenia w sieci pracują w trybie full-duplex z prędkością 100 Mb/s. Profinet wykorzystuje między innymi protokoły TCP/IP i standardy IT. Komunikacja w sieci inicjalizowana jest za pomocą protokołu TCP/UDP, a diagnostyka przysyłana jest acyklicznie. Jednocześnie dane procesowe przesyłane są cyklicznie. Maksymalna liczba urządzeń w sieci to 256.

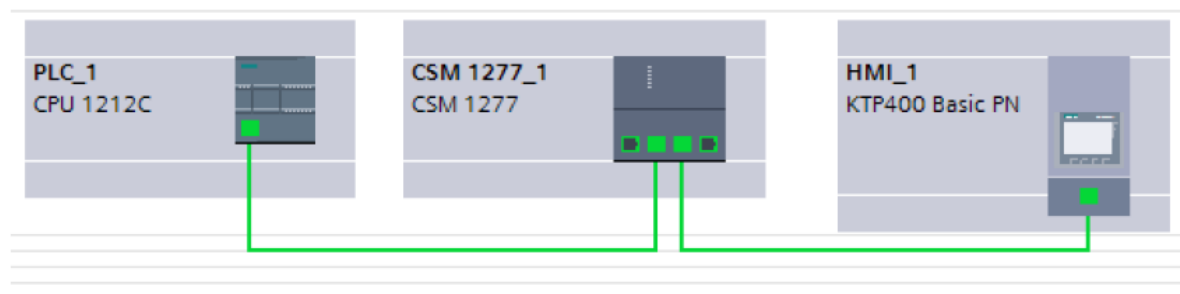
### 3. Wykonanie ćwiczenia

Na samym początku musieliśmy dokonać konfiguracji sprzętu zgodnie z tabelą 1 zamieszczoną poniżej.

**Tabela. 1** Konfiguracja sprzętu

Slot nr	Funkcja modułu	Oznaczenie	Numer katalogowy 6ES7...
1	Jednostka centralna	CPU	212-1BE40-0XB0
2	Panel operatorski	HMI	6AV2 123-2DB03-0AX0
3	Wyjścia analogowe	AQ2x14BIT	232-4HB32-0XB0
4	Sterownik TURCK BL 20		
	Switch sieciowy	CSM 1277 SIMATIC NET	6GK7-277-1AA10-0AA0

Dla jednostki centralnej ustawiliśmy prawidłowy adres IP - **192.168.1.45.** , a także musieliśmy zaznaczyć opcję generate Profinet device name automatically. Dla panelu operatorskiego musieliśmy przypisać odpowiedni adres IP - **192.168.1.2.**



**Rysunek 2.** Konfiguracja sieci

Następnie musieliśmy dodatkowo zainstalować moduły dla urządzenia sieciowego TURCK-CDS3. Gdy to już było gotowe, mogliśmy przejść do tworzenia programu sterującego. Musieliśmy stworzyć odpowiednią funkcję odpowiedzialną za komunikację ze sterownikiem TURCK, która to wywoływana byłaby w pętli głównej programu, ale najpierw utworzyliśmy odpowiednią tabelę z nazwami symbolicznymi: tagami, która w przyszłości pozwoliła nam uniknąć konieczności ciągłego odwoływania się do miejsc w pamięci: wystarczyło używać nazw nadanych właśnie tym miejscom w pamięci.

PLC tags							
	Name	Data type	Address	Retain	Visible in HMI	Accessible from HMI	Comment
❏	In0	Bool	%I2.0	False	True	True	
❏	In1	Bool	%I2.1	False	True	True	
❏	In2	Bool	%I2.2	False	True	True	
❏	In3	Bool	%I2.3	False	True	True	
❏	In4	Bool	%I2.4	False	True	True	
❏	In5	Bool	%I2.5	False	True	True	
❏	In6	Bool	%I2.6	False	True	True	
❏	In7	Bool	%I2.7	False	True	True	
❏	Out0	Bool	%Q2.0	False	True	True	
❏	Out1	Bool	%Q2.1	False	True	True	
❏	Out2	Bool	%Q2.2	False	True	True	
❏	Out3	Bool	%Q2.3	False	True	True	
❏	Out4	Bool	%Q2.4	False	True	True	
❏	Out5	Bool	%Q2.5	False	True	True	
❏	Out6	Bool	%Q2.6	False	True	True	
❏	Out7	Bool	%Q2.7	False	True	True	
❏	DQ0	Bool	%Q0.5	False	True	True	
❏	Pressure_PV	Word	%IW3	False	True	True	
❏	Pressure_SP	Word	%IW5	False	True	True	
❏	Valve_PV	Word	%IW7	False	True	True	

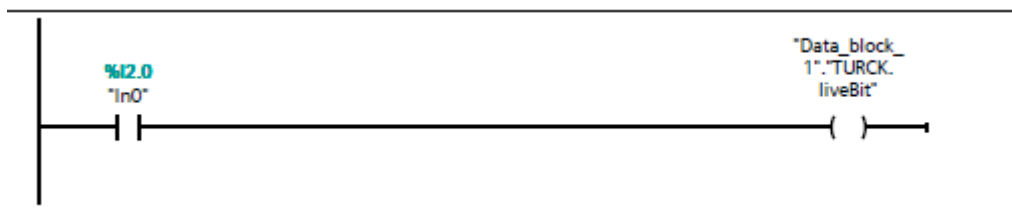
Rysunek 3. Utworzone przez nas Tag Table

Następnie stworzyliśmy blok danych (DB) przechowujący w pamięci informacje o zmiennych procesowych i statusie sterownika TURCK.

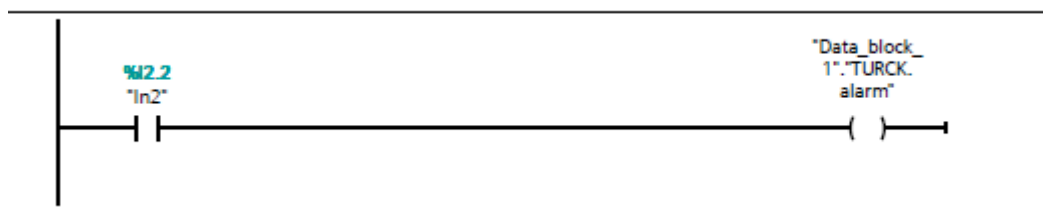
Data_block_1 Properties									
General									
Name	Data_block_1	Number	1	Type	DB	Language	DB	Numbering	automatic
Information									
Title		Author		Comment		Family		Version	0.1
User-defined ID									
Name	Data type	Start value	Retain	Accessible from HMI	Visible in HMI	Setpoint	Comment		
▼ Static									
TURCK.liveBit	Bool	false	False	True	True	False			
TURCK.mode	Bool	false	False	True	True	False			
TURCK.alarm	Bool	false	False	True	True	False			
TURCK.PV	Real	0.0	False	True	True	False			
TURCK.SP	Real	0.0	False	True	True	False			
TURCK.VALVE	Real	0.0	False	True	True	False			

Rysunek 4. Blok Danych

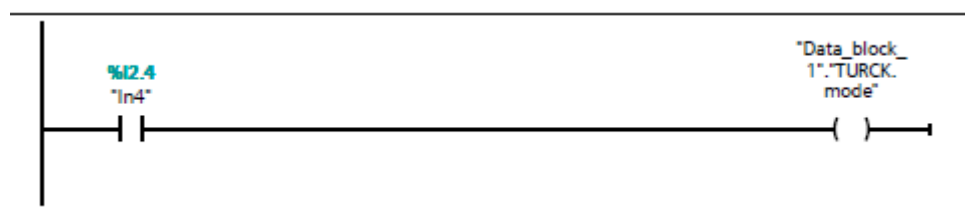
Następnie przeszliśmy do pisania funkcji (FC), która miała wykonywać się w bloku głównym programu (main). Stworzyliśmy odpowiednie funkcje odczytujące oraz funkcję które miały za zadanie przeliczać wartości zmiennych procesowych ze sterownika TURCK. Aby tego dokonać korzystaliśmy z bloczka „Calculate”.



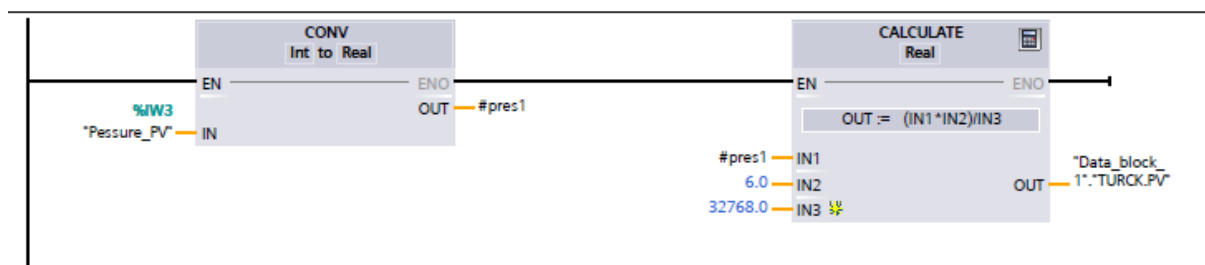
Rysunek 5. Kod alokujący wartość "LiveBit" sterownika TURCK



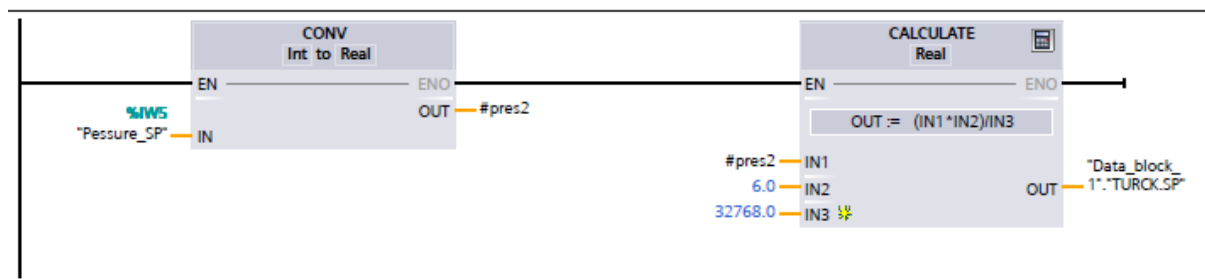
Rysunek 6. Kod alokujący bit alarmu



Rysunek 7. Kod alokujący bit sterowania manualnego



Rysunek 8. Kod konwertujący wartość odczytaną ciśnienia

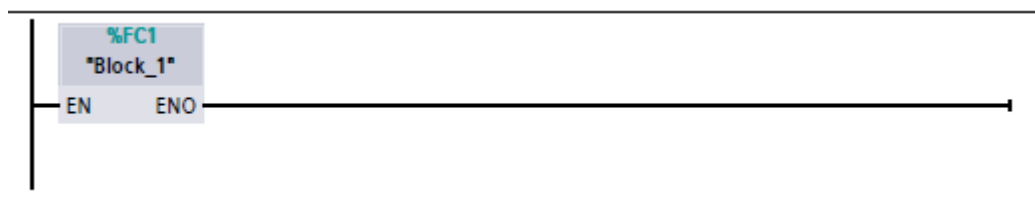


Rysunek 9. Kod konwertujący wartość zadaną ciśnienia



Rysunek 10. Kod konwertujący wartość odczytaną ciśnienia

Powyżej przedstawiona została funkcja FC. Poniżej przedstawiamy jej wywołanie w bloku głównym programu:



**Rysunek 11. Bloczek Main**

Aby powyższa funkcja działa poprawnie (funkcje Calculate), stworzyliśmy zmienne lokalne w bloku FC1.

Name	Data type
Input	
Output	
InOut	
▼ Temp	
pres1	Real
pres2	Real
pres3	Real
Constant	
▼ Return	
Block_1	Void

**Rysunek 12. Zmienne lokalne w bloku FC1**

Następnym naszym zadaniem było stworzenie panelu operatorskiego. Nasz panel operatorski HMI składał się z trzech różnych ekranów:

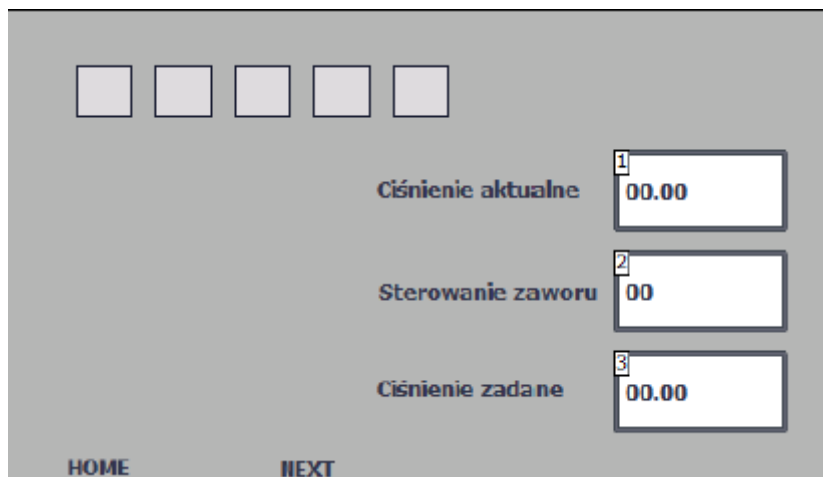
- Ekran główny – do tego ekranu wracało się za pomocą przyciśnięcia klawisza F1



**Rysunek 13. Ekran główny**

Niestety z niewiadomych przyczyn podczas wciskania klawisza F1 tylko na kilka chwil pokazywał nam się ekran główny, a następnie automatycznie zmieniał się na pierwszy pod-ekran. Tak samo działało się podczas załadowywania naszego programu na HMI. Główny ekran wyświetlał się na kilka chwil a następnie automatycznie przełączał się na pierwszy pod-ekran.

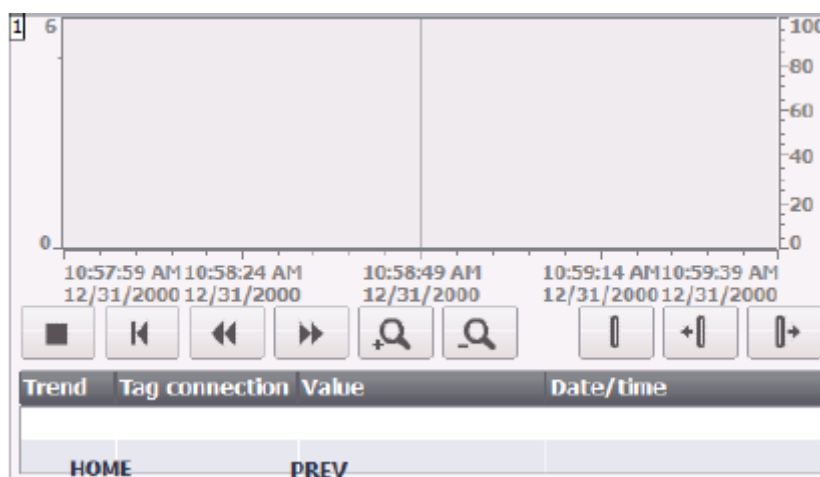
- Pierwszy pod-ekran – ekran zawierający podgląd wartości z procesu regulacji ciśnienia w zbiorniku (Pressure PV, Pressure SP, Value PV). Za pomocą klawisza F2 można było przełączyć się na drugi pod-ekran.



**Rysunek 14.** Pod-ekran podglądu wartości

Na powyższym ekranie możemy zauważyć pięć prostokątów. Są to diody sygnalizujące działanie odpowiednich procesów. Zaczynając od lewej strony:

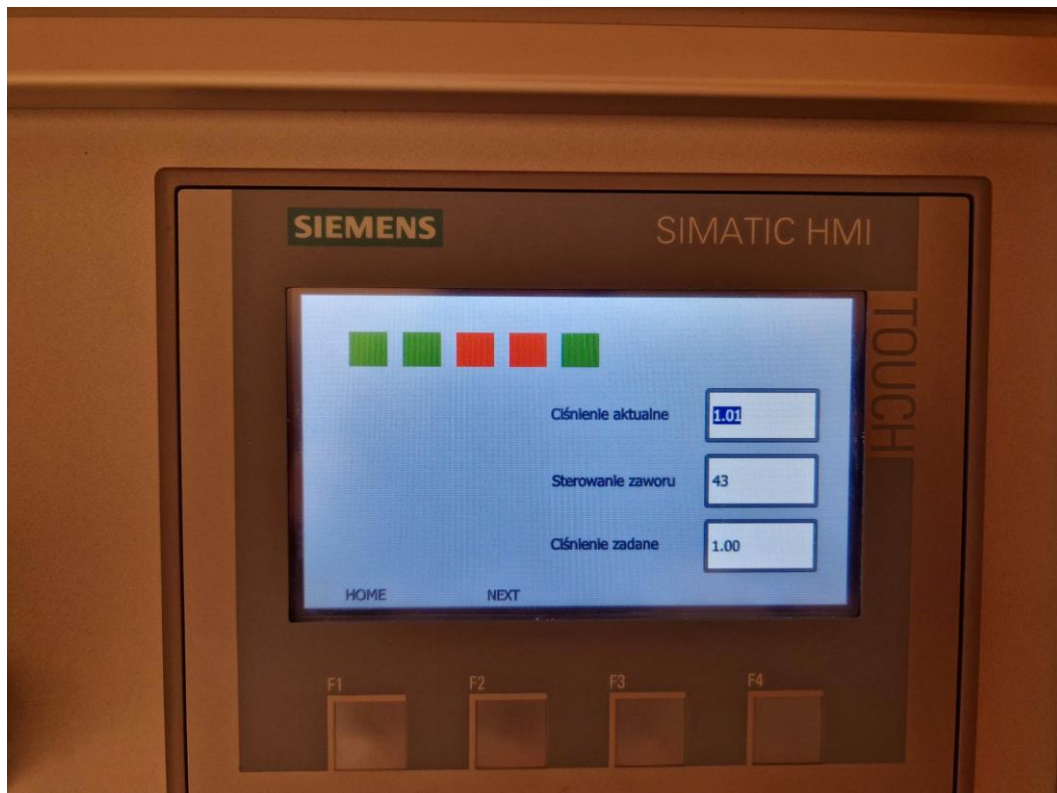
- Dioda 1 – sygnalizowała „LiveBit” ze sterownika TURCK
  - Dioda 2 – sygnalizowała czy sterownik pracuje bez błędów
  - Dioda 3 – sygnalizowała czy stanowisko pracuje w trybie sterowania ręcznego
  - Dioda 4 – sygnalizowała czy wartość aktualna nie jest równa wartości zadanej
  - Dioda 5 – sygnalizowała czy wartość aktualna = wartość zadana
- Drugi pod-ekran – ekran zawierający obserwację przebiegów czasowych z wartości zmiennych procesowych (SP, PV) - zawiera graf trendu. Za pomocą klawisza F2 można było przełączyć się na pierwszy pod-ekran.



**Rysunek 15.** Pod-ekran grafu trendu

Niestety z niewiadomych przyczyn na ekranie grafu nie wyświetlała nam się wartość aktualnego ciśnienia, pomimo że na pierwszym pod-ekranie wartość ta normalnie się nam wyświetla.

Poniżej na rysunkach 15 i 16 przedstawiamy wyniki które uzyskaliśmy tworząc nasz projekt. Wyniki są zwizualizowane na wyżej omówionych pod ekranach.



**Rysunek 16.** Pod-ekran podglądu wartości na żywo



**Rysunek 17.** Pod-ekran grafu trendu na żywo



## 4. Wnioski

Dzięki stworzeniu i skonfigurowaniu prostego, rozproszonego systemu sterowania mogliśmy w ciekawy i efektywny sposób zapoznać się z siecią przemysłową Profinet. Profinet nie działa na zasadzie „master-slave” jak Profibus oraz posiada full-duplex (dane mogą jednocześnie być przesyłane w obu kierunkach) co pozwala na budowę bardziej złożonych sieci, niż po prostu magistrala z urządzeniami sterowanymi, jak w przypadku Profibus. W naszym przypadku była to sieć złożona z dwóch sterowników. W trakcie wykonywania laboratorium wykorzystywaliśmy funkcje FC oraz bloki danych DB co pozwoliło nam lepiej zapoznać się z ich funkcjonalnościami. Stworzyliśmy także panel operatorski HMI, który był złożony z ekranu głównego i 2 pod-ekranów, czego wcześniej nie robiliśmy. Uważamy że takie zastosowanie panelu operatorskiego jest na pewno bardzo przydatne podczas pracy z większą ilością sterowników – program który napisaliśmy staje się bardziej przejrzysty i łatwiejszy do zrozumienia.