

# SCADA TIA Portal - Conectivitate

## • Obiective

- Configurare Web Server
- Realizarea de aplicatii Web Server
- Conectarea la o baza de date SQL
- Open Platform Communications (OPC)

## • Organizarea sarcinilor de lucru

- Parcurgeti cele trei capitole ale cursului.
- In cadrul fiecarui capitol urmariti exemplele ilustrative si incercati sa le realizati in medul de dezvoltare "Citect".
- Fixati principalele idei ale cursului, prezentate în rezumat.
- Completati testul de autoevaluare.
- Timpul de lucru pentru parcurgerea testului de autoevaluare este de 15 minute.

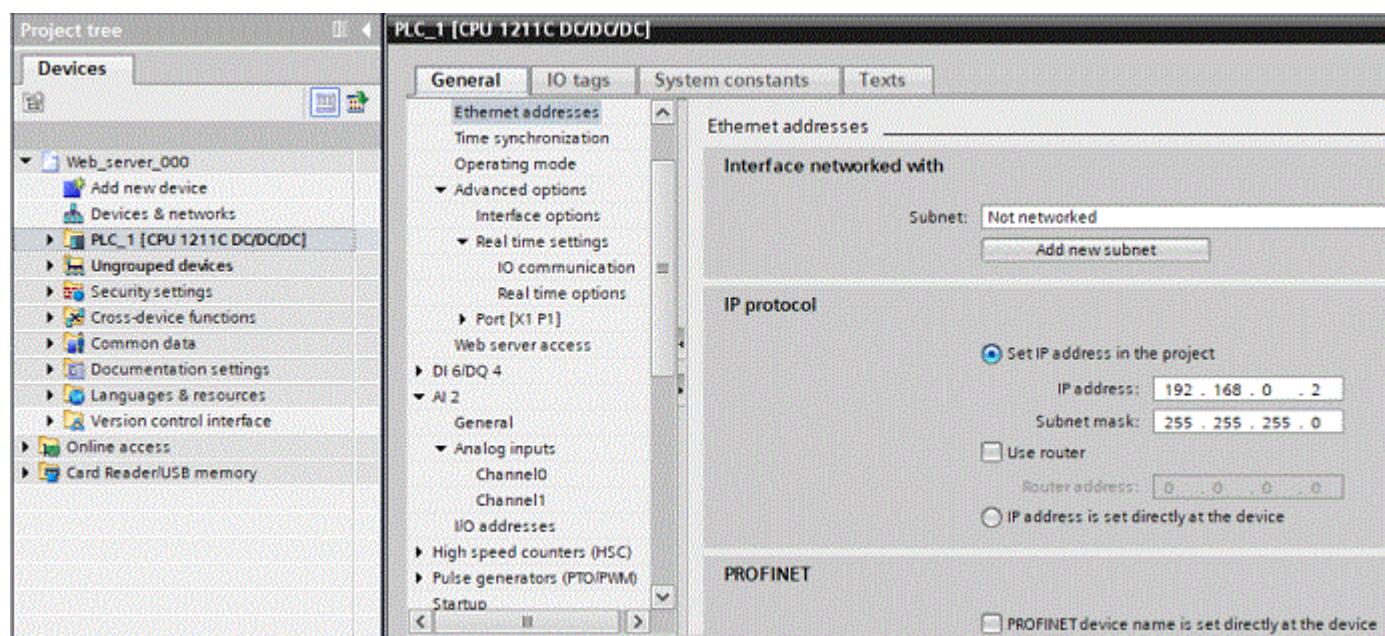
## 1. Web Server

### Configurare Web Server

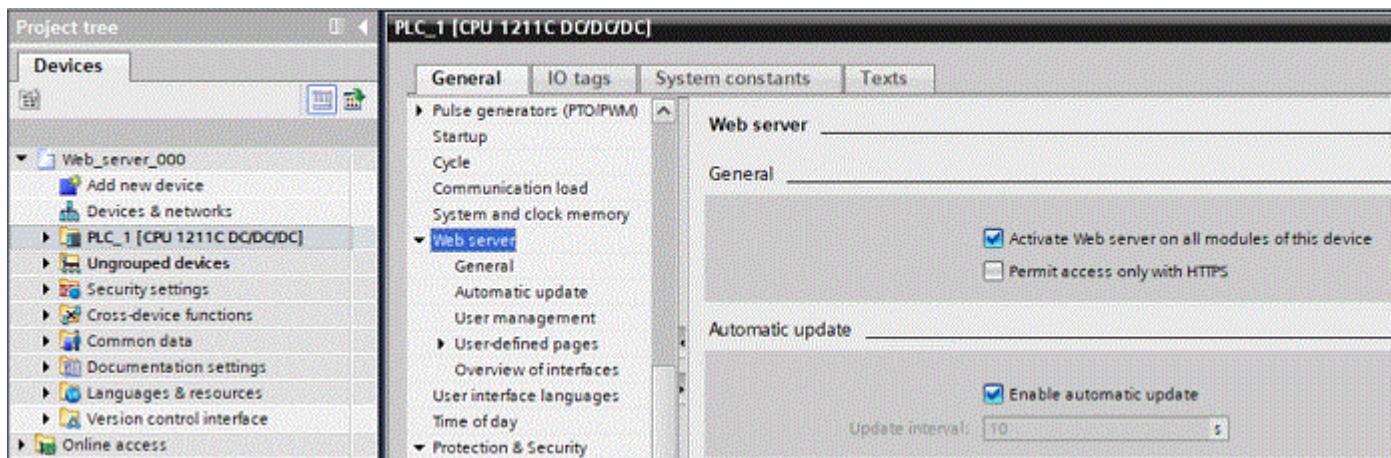
Vom dezvolta in continuare proiectul numit **Web server 000**

In cadrul acestui proiect se va configura un WebServer care va furniza paginile web din folder-ul: **Web pages**

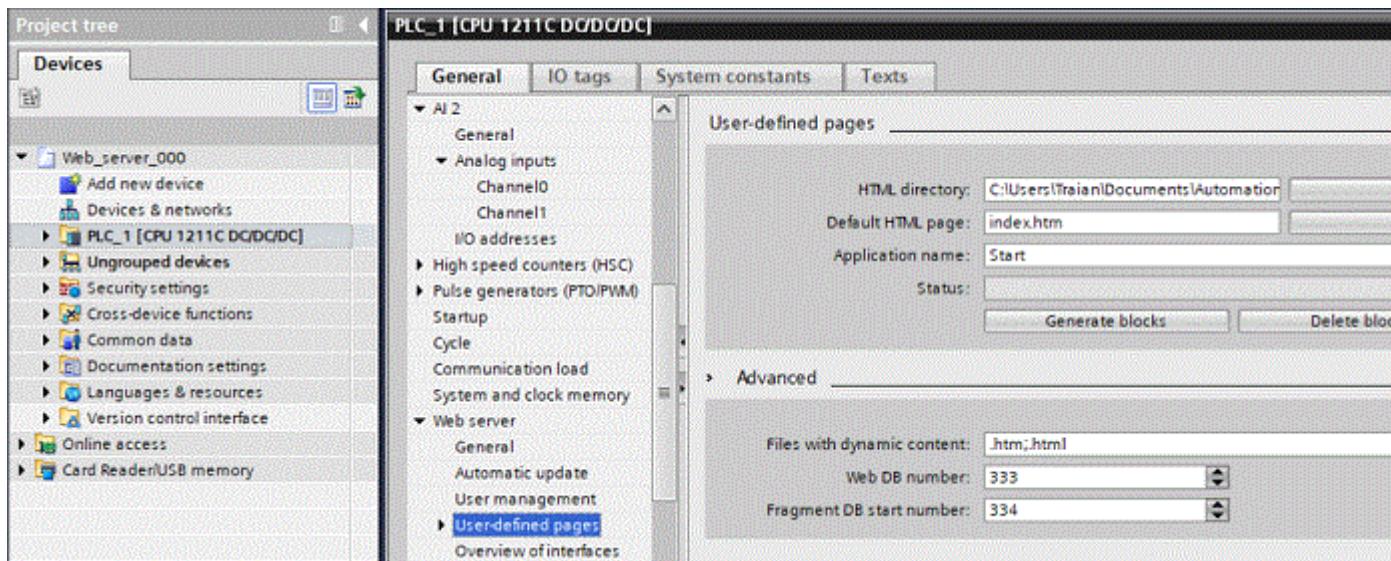
Configurare PROFINET Interface --> Ethernet addresses



Configurare Web Server --> General



Configurare Web Server --> User Defined pages



## Aplicatii Web Server

Program blocks --> Main

The screenshot shows the SIMATIC Manager software interface. On the left, the Project tree is displayed under the 'Devices' tab, showing a hierarchy including 'Web\_server\_000', 'PLC\_1 [CPU 1211C DC/DC/DC]', and 'Program blocks'. Under 'Program blocks', 'Main [OB1]' is selected. The main workspace on the right shows the 'Main' program block. The block title is "Main Program Sweep (Cycle)". The block contains a single network labeled "Network 1:". This network consists of a single instruction: a call instruction (CALL) with the label "WWW". The parameters for this call are: EN (333), CTRL\_DB (%MWD), and RET\_VAL ("Web\_cnt").

## Definirea Tag-urilor

Vom defini:

- 8 TAG-uri PLC de tip bool-ean, cu numele B00 - B07 aflate pe intrarile digitale DIGITAL INPUT 0.0 - 0.7
- 8 TAG-uri PLC de tip bool-ean, cu numele B08 - B15 aflate pe intrarile digitale DIGITAL INPUT 1.0 - 1.7.
- 8 TAG-uri PLC de tip bool-ean, cu numele C00 - C07 aflate pe iesirile digitale DIGITAL OUTPUT 0.0 - 0.7.
- 8 TAG-uri PLC de tip bool-ean, cu numele C08 - C15 aflate pe iesirile digitale DIGITAL OUTPUT 1.0 - 1.7.

Vom mai defini tag-urile:

Web\_server\_000 ▶ PLC\_1 [CPU 1215C DC/DC/DC] ▶ PLC tags

	Name	Tag table	Data type	Address	Retain	Access
41	Com	list taguri	DWord	%MD20	<input type="checkbox"/>	<input checked="" type="checkbox"/>
42	Conn_cnt	list taguri	Int	%MW32	<input type="checkbox"/>	<input checked="" type="checkbox"/>
43	Digi	list taguri	DWord	%MD24	<input type="checkbox"/>	<input checked="" type="checkbox"/>
44	i	list taguri	Int	%MW4	<input type="checkbox"/>	<input checked="" type="checkbox"/>
45	k	list taguri	Int	%MW6	<input type="checkbox"/>	<input checked="" type="checkbox"/>
46	Loc	list taguri	Bool	%M4.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>
47	Mod_I	list taguri	Int	%MW0	<input type="checkbox"/>	<input checked="" type="checkbox"/>
48	Plc_cnt	list taguri	Int	%MW30	<input type="checkbox"/>	<input checked="" type="checkbox"/>
49	Web_cnt	list taguri	Int	%MM28	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Pentru a realiza pagini Web in care sa accesam mai eficient Tag-urile de mai sus, vom crea un Program Blocks -- Cyclic Interrupt in care vom codifica binar-zecimal toate intrarile binare B01-B15 precum si iesirile digitale C00-C15, pentru a face mai eficienta comunicarea intre PLC si paginile web.

```

"Plc_cnt" := "Plc_cnt" + 1;
IF "Plc_cnt" > 100 THEN
    "Plc_cnt" := 0;
;
END_IF;

// Calculez valoarea "Digi"

"Digi" := 0;
IF "B00" THEN
    "Digi" := "Digi" + 1;
END_IF;
IF "B01" THEN
    "Digi" := "Digi" + 2;
END_IF;
IF "B02" THEN
    "Digi" := "Digi" + 4;
END_IF;
IF "B03" THEN
    "Digi" := "Digi" + 8;
END_IF;
IF "B04" THEN
    "Digi" := "Digi" + 16;
END_IF;
IF "B05" THEN
    "Digi" := "Digi" + 32;
END_IF;
IF "B06" THEN
    "Digi" := "Digi" + 64;
END_IF;

```

```

IF "B07" THEN
    "Digi" := "Digi" + 128;
END_IF;
IF "B08" THEN
    "Digi" := "Digi" + 256;
END_IF;
IF "B09" THEN
    "Digi" := "Digi" + 512;
END_IF;
IF "B10" THEN
    "Digi" := "Digi" + 1024;
END_IF;
IF "B11" THEN
    "Digi" := "Digi" + 2048;
END_IF;
IF "B12" THEN
    "Digi" := "Digi" + 4096;
END_IF;
IF "B13" THEN
    "Digi" := "Digi" + 8192;
END_IF;
IF "B14" THEN
    "Digi" := "Digi" + 16384;
END_IF;
IF "B15" THEN
    "Digi" := "Digi" + 32768;
END_IF;

// Activez cele doua led-uri

"k" := "k" + 1;
IF "k" > 10 THEN
    "k" := 1;
END_IF;
IF ("k" MOD 2) = 0 THEN
    "C16" := TRUE;
    "C17" := FALSE;
ELSE
    "C16" := FALSE;
    "C17" := TRUE;
END_IF;

// Actualizez C0-C15 din valoarea "Com"
IF (NOT "Loc") THEN
    #vl := "Com";
    FOR #j := 0 TO 15 DO
        IF (#vl MOD 2) = 0 THEN
            #b := FALSE;
        ELSE
            #b := TRUE;
        END_IF;
        #vl := #vl / 2;
        CASE #j OF
            0:
                "C00" := #b;
            1:
                "C01" := #b;
            2:
                "C02" := #b;
            3:

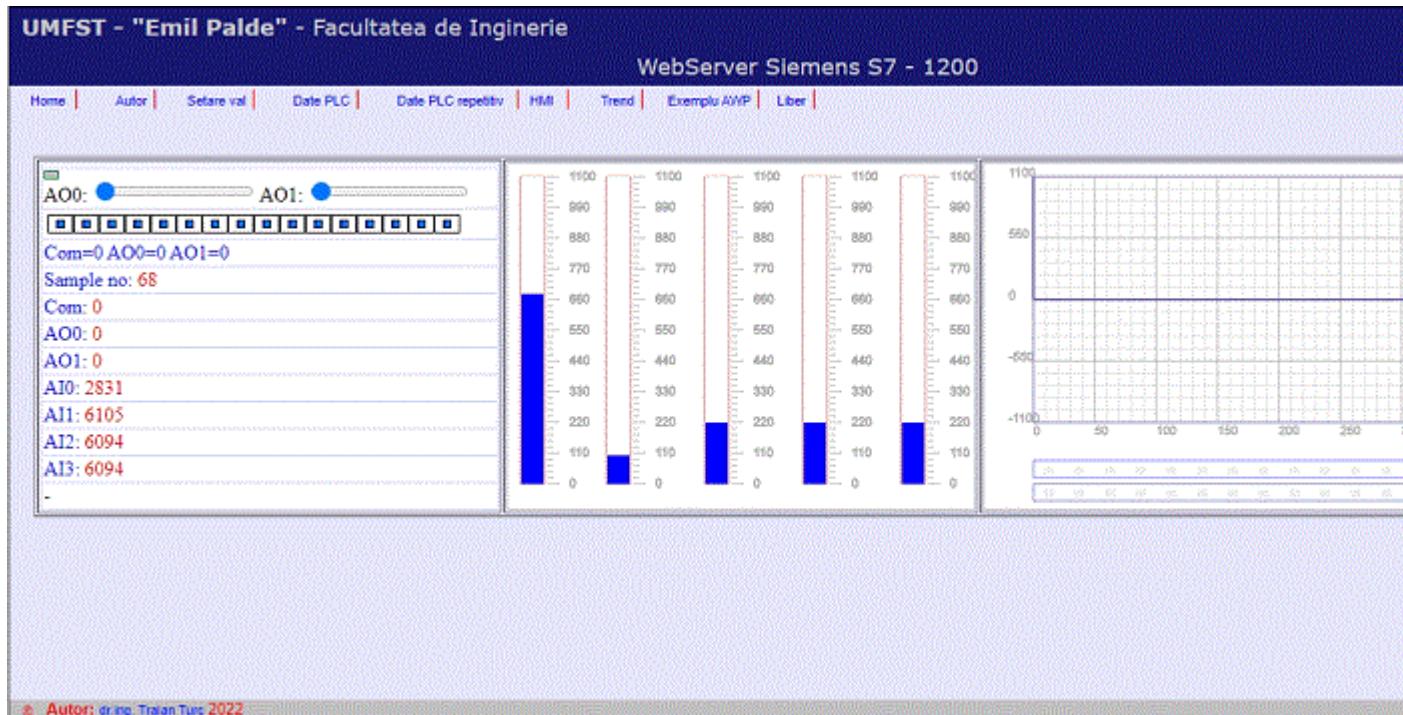
```

```

        "C03" := #b;
4:      "C04" := #b;
5:      "C05" := #b;
6:      "C06" := #b;
7:      "C07" := #b;
8:      "C08" := #b;
9:      "C09" := #b;
10:     "C10" := #b;
11:     "C11" := #b;
12:     "C12" := #b;
13:     "C13" := #b;
14:     "C14" := #b;
15:     "C15" := #b;
    END_CASE;
  END_FOR;
END_IF;

```

Vom realiza in continuare o pagina Web care acceseaza tag-urile definite.



Paginile web sunt realizate in HTML si AWP(Automation Web Programming). Comenzile AWP sunt inserate sub forma de comentarii HTML. Comenzile AWP pot fi plasate in orice

pozitie din fisierul HTML. Cu toate acestea, din motive de claritate, se recomanda plasarea comenzilor AWP la inceputul fisierului HTML.

Formatul comenzi AWP pentru declararea unei variabile:

```
<!-- AWP_In_Variable Name='''Nume_var''' -->
```

Toate variabilele transferate spre CPU trebuie sa fie definite ca **AWP\_In\_Variable**. Variabila este scrisa intre ghilimele duble si incadrate in ghilimele simple ("..."). Toate variabilele citite din CPU trebuie sa aiba forma:

```
:='''Nume_var'''
```

Sa presupunem ca avem definite PLC Tag-urile:Com, AO0, AO1.

Pagina web **set\_val.html**, care permite modificarea Tag-urilor Com, AO0, AO1 va fi:

```
<!-- AWP_In_Variable Name='''Com''' -->
<!-- AWP_In_Variable Name='''AO0''' -->
<!-- AWP_In_Variable Name='''AO1''' -->

<form method="get" action="set_val.html">
    <input type="submit" value="Modific">
    Com <input type="text" name='''Com'''' value=':=Com:'>
    AO0 <input type="text" name='''AO0'''' value=':=AO0:'>
</form>
```

Pagina **date\_v0.html** este un exemplu simplu in care se modifica respectiv se afiseaza vlorile controlate de PLC prin intermediul Tag-urilor definite mai sus.

0	
AO0	0
AO1	0
<input type="button" value="Modific"/>	

Contor Plc- :="Plc\_cnt":

AI0 ----- :="AI0":

AI1 ----- :="AI1":

AI2 ----- :="AI2":

AI3 ----- :="AI3":

Com ----- := "Com";  
Digi ----- := "Digi";  
AO0 ----- := "AO0\_w";  
AO1 ----- := "AO1\_w";

---

Afisarea acestei pagini se va face utilizand orice browser in care se apeleaza url-ul:  
**http://192.168.0.11/awp//index.htm**

**UMFST - "Emil Palde" - Facultatea de Inginerie**

**WebServer Siemens S7 - 1200**

Home | Autor | Setare val | Date PLC | Date PLC repetitiv | HMI | Trend | Exemplu AWP | Liber |

Com   
AO0   
AO1

Contor Plc- 15  
AI0 ----- 2832  
AI1 ----- 6104  
AI2 ----- 6095  
AI3 ----- 6095  
Com ----- 0  
Digi ----- 0  
AO0 ----- 0  
AO1 ----- 0

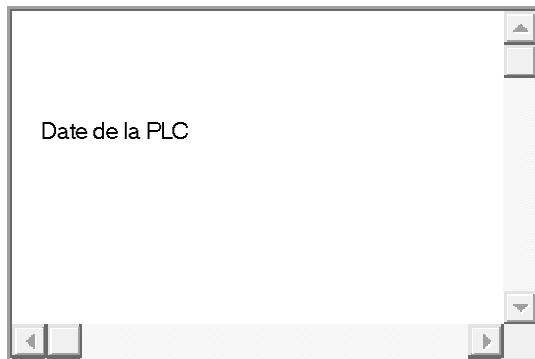
---

[Exemplu awp](#)

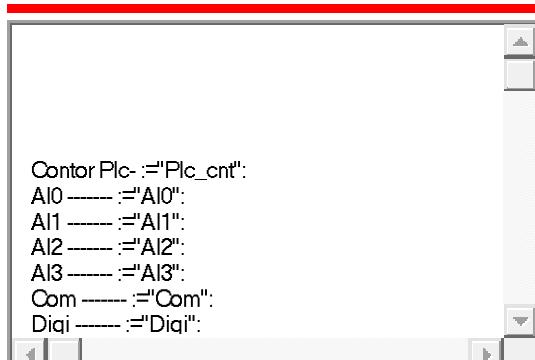
Datele provenite de la PLC sunt dinamice. Pentru a afisa un nou set de date pagina trebuie reincarcata.

Aplicatia **date\_r.html** afiseaza repetitiv datele provenite de la PLC





Continutul **date\_v1.html** find:



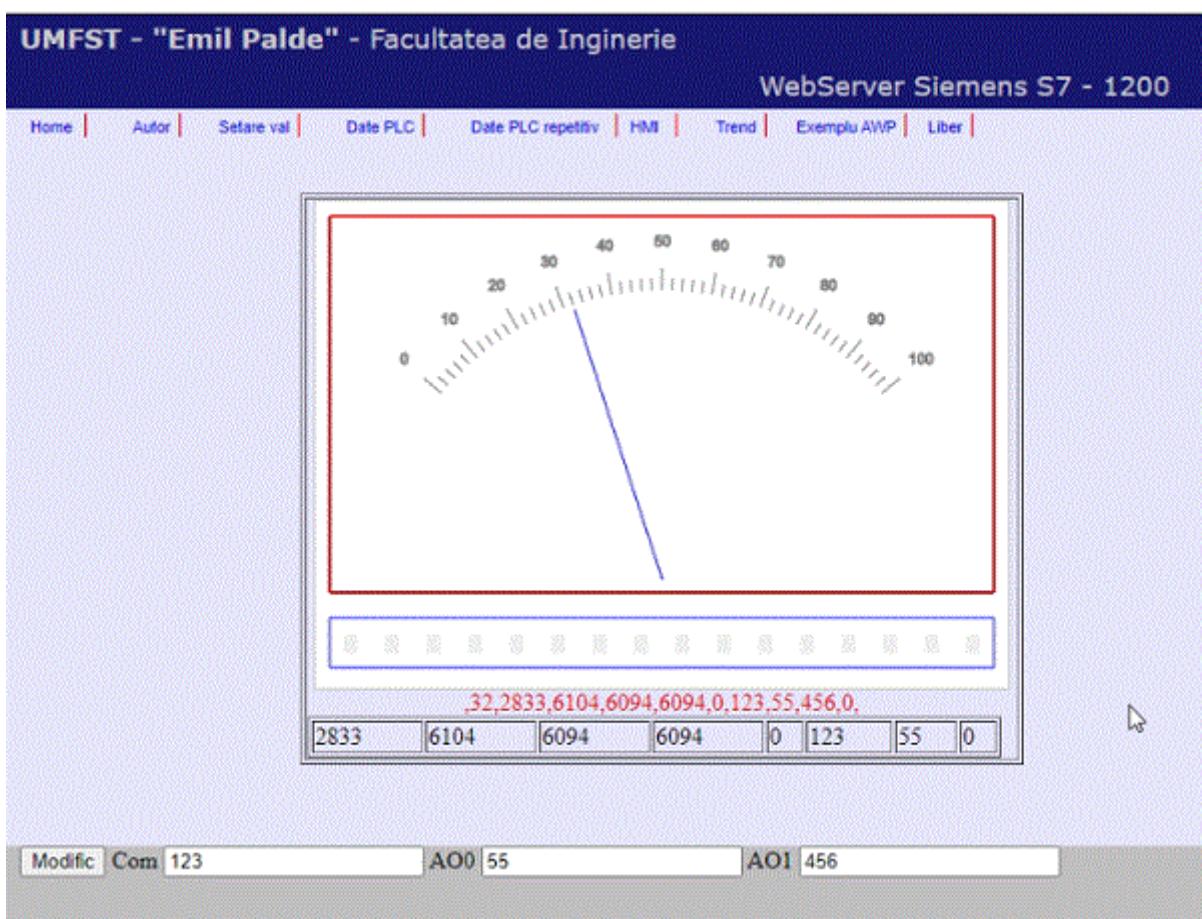
Aplicatia **date\_r.html** afiseaza dinamic datele de la PLC insa dezavantajul este ca pagina "clipeste" la fiecare reincarcare.

Cel mai potrivit ar fi sa folosim tehnologia "Ajax" si sa aducem repetitiv numai datele nemaifiid necesara reincarcarea paginii.

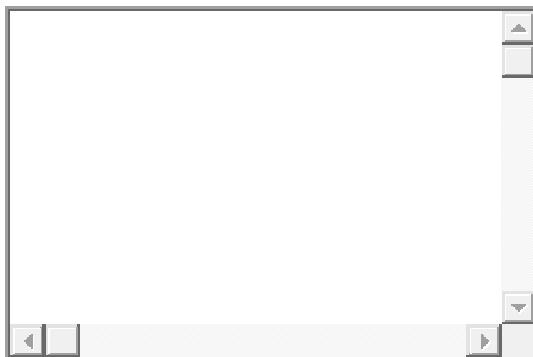
Vom crea pagina **date.html** care trimite datele in vederea preluarii acestora prin tehnologia "Ajax"

```
,:"Plc_cnt",:"AI0",:"AI1",:"AI2",:"AI3",:"Digi"
```

Dupa cum se observa valorile Tag-urilor  
"Plc\_cnt","AI0","AI1","AI2","AI3","Digi","Com","AO0\_w","AO1\_w","Conn\_cnt" sunt trimise sub forma de text delimitate prin separatorl ",".  
Aplicatia **hmi\_ajax.html** afiseaza dinamic valorile primite de la PLC.



Your browser does not support the HTML5 canvas tag.



Clasa voltm fiind: [voltm](#) Exemplu de utilizare: [ex1](#)

## 2. conexiune OPC

Capitol realizat cu contributia asist: Tudor Covrig

### Configurare IP Calculator

- Se verifica daca PC se afla in aceea?i retea cu PLC-ul:

A screenshot of a Windows Command Prompt window. The title bar says "C:\ Linie de comandă". The window contains the following text:

```
Microsoft Windows [Version 10.0.19044.1706]
(c) Microsoft Corporation. Toate drepturile rezervate.

C:\Users\Tudor>ping 192.168.0.11
```

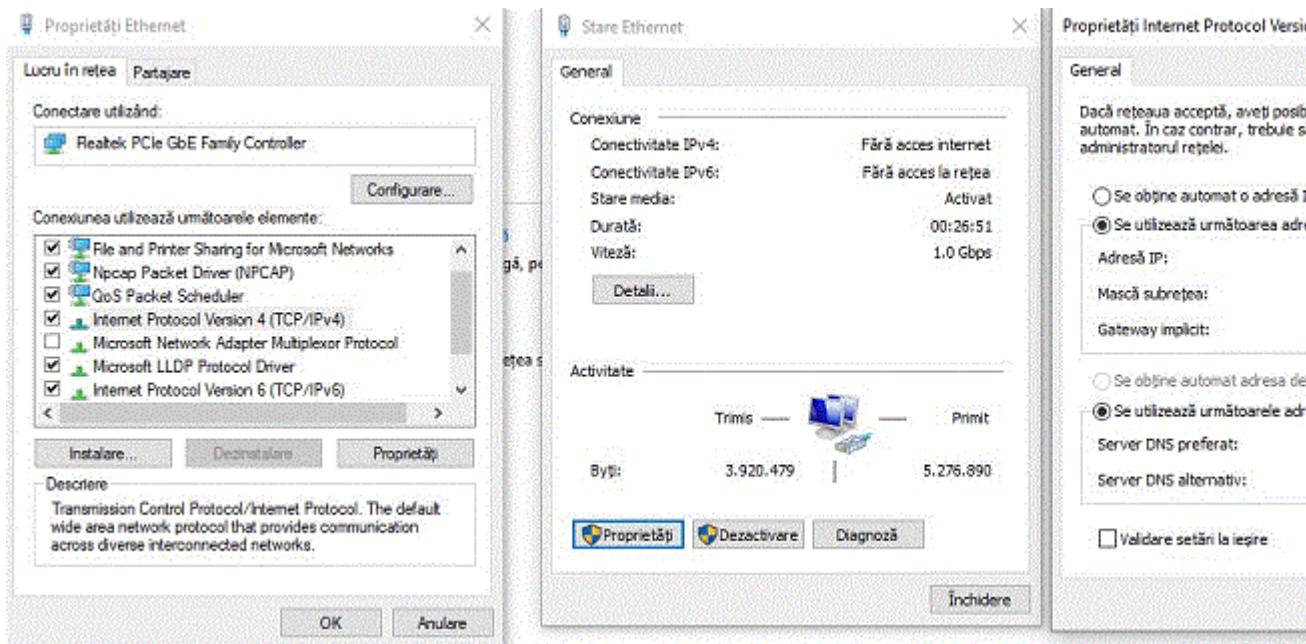
The command "ping 192.168.0.11" is highlighted with a red rectangle.

```
Pinging 192.168.0.11 with 32 bytes of data:
Reply from 192.168.0.11: bytes=32 time=3ms TTL=255
Reply from 192.168.0.11: bytes=32 time=1ms TTL=255
Reply from 192.168.0.11: bytes=32 time=1ms TTL=255
Reply from 192.168.0.11: bytes=32 time=1ms TTL=255

Ping statistics for 192.168.0.11:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 3ms, Average = 1ms

C:\Users\Tudor>
```

- Daca cele doua nu sunt in aceeasi retea, se modifica adresa IP a calculatorului:

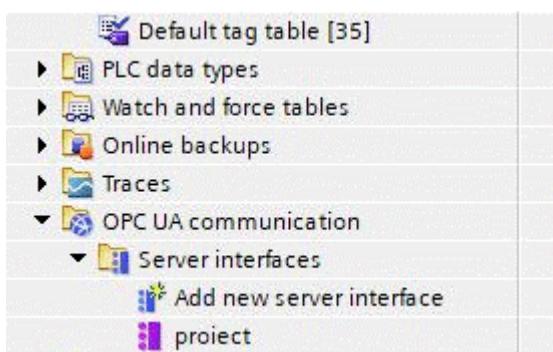


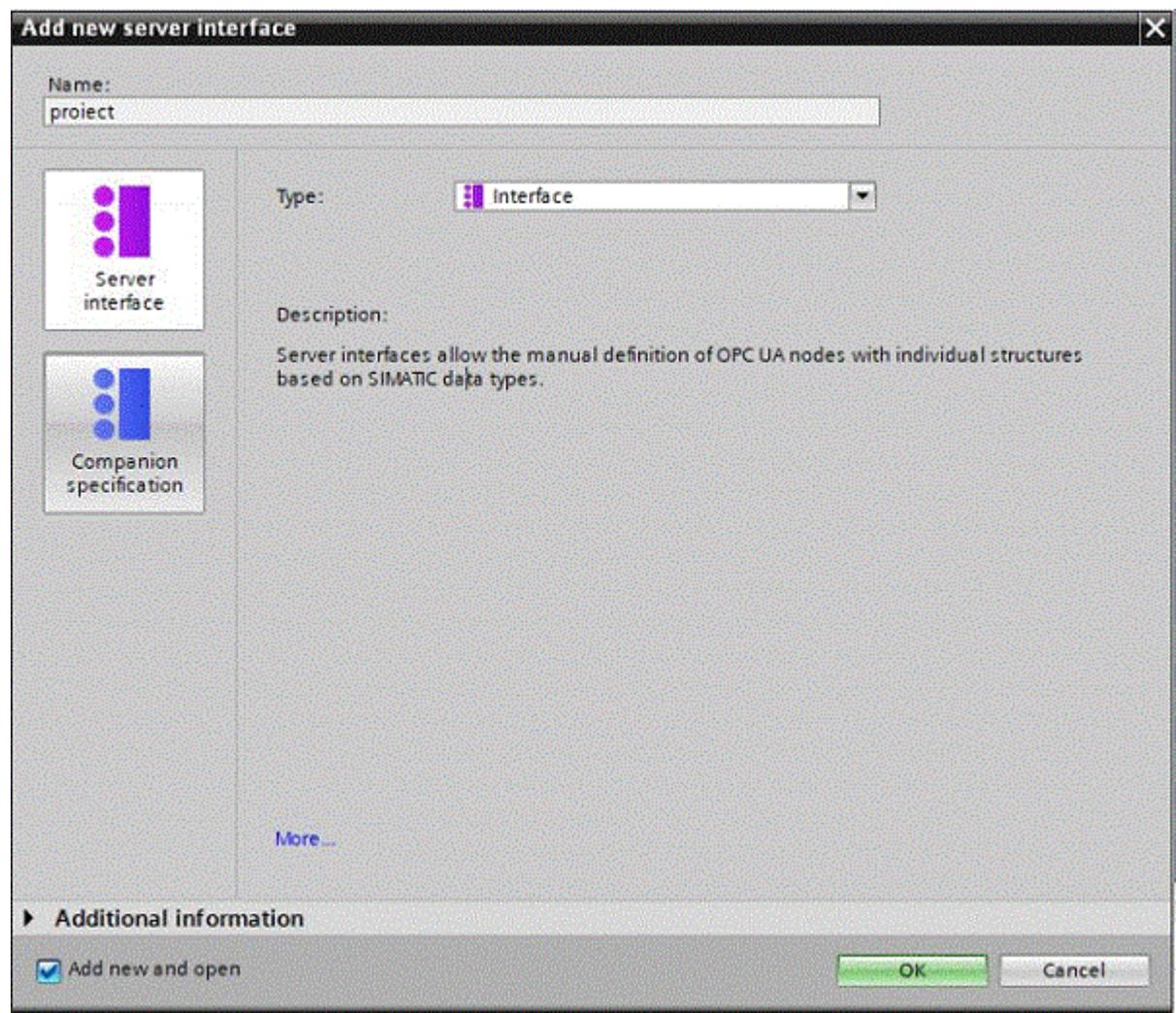
## Configurare OPC

- Se creaza variabilele tag care urmeaza sa fie exportate:

	Name	Tag table	Data type	Address	Retain	Acc
1	led1	Default tag table	Bool	%Q0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>
2	led2	Default tag table	Bool	%Q0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>
3	led3	Default tag table	Bool	%Q0.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>
4	<Add new>					

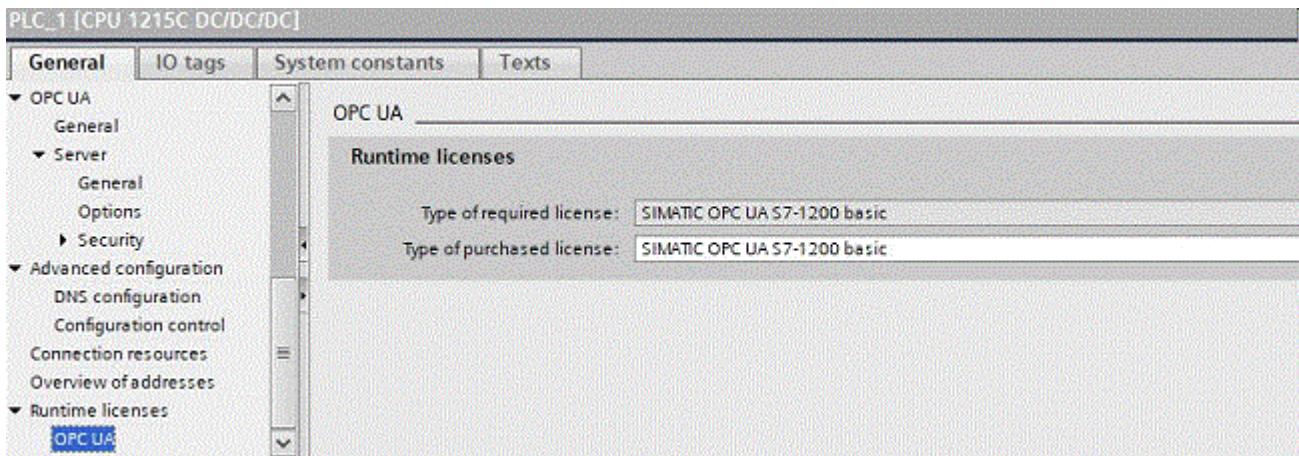
- Se adauga un nou tabel in care se salveaza variabilele pe care dorim sa le exportam





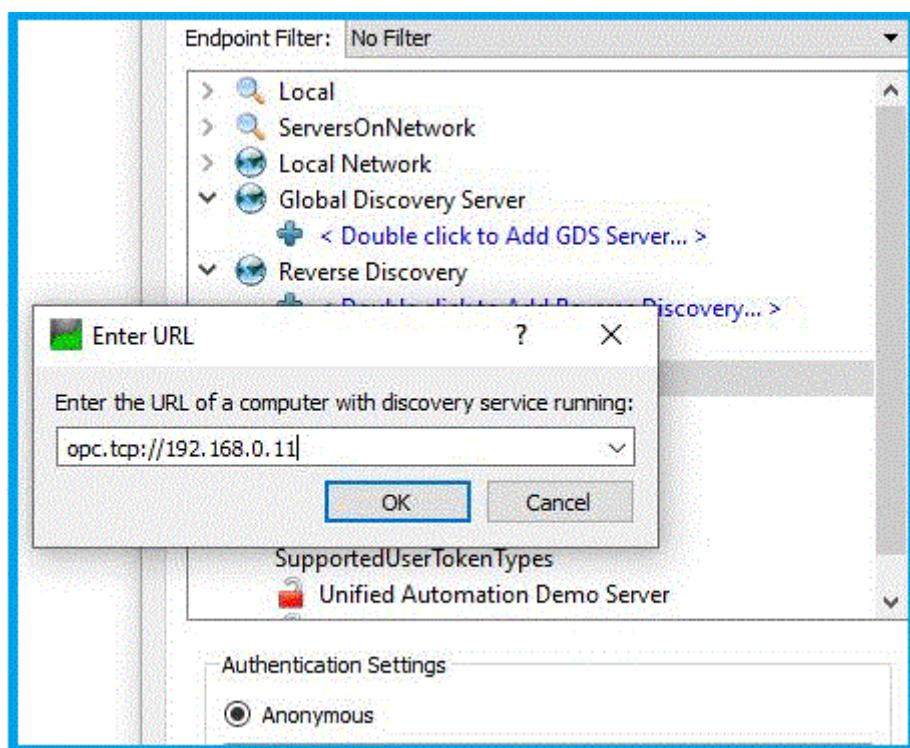
OPC UA server interface				
Browse name	Node type	Local data	Data type	OPC L
1 project	Interface			1
2 led1	BOOL	"led1"		2
3 led2	BOOL	"led2"		3
4 led3	BOOL	"led3"		4
5 <Add new>				5
				6
				7
				8

- Se activeaza licenta OPC si se incarca programul în PLC:

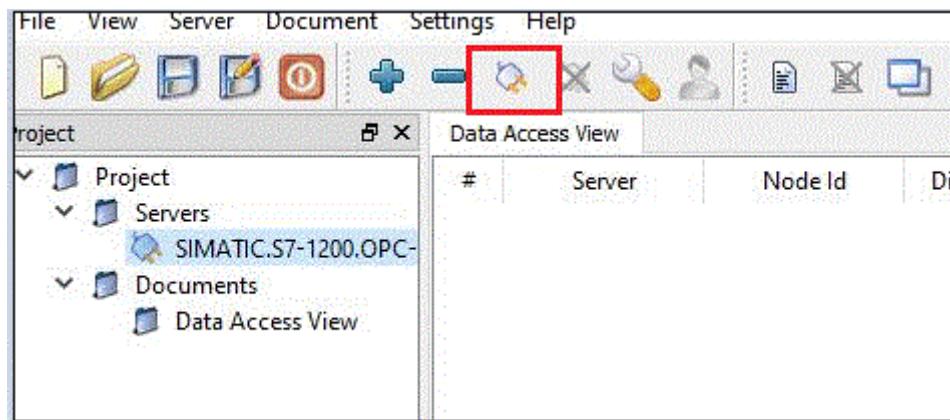
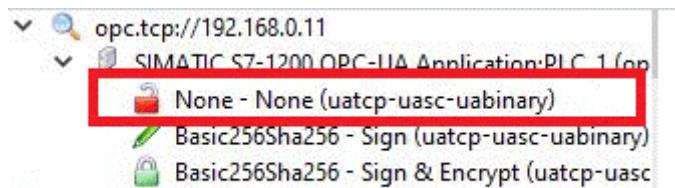


## Instalarea aplicatiei UaExpert

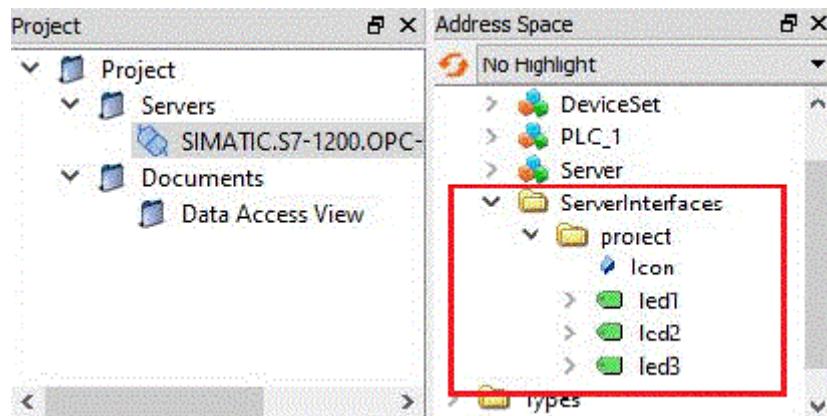
- Se descarca aplicatia [UaExpert](#)
- Se ruleaza aplicatia si se selecteaza meniul „Add server” dupa care se adauga adresa IP a PLC-ului cu care dorim sa comunicam.



- Se selecteaza „None-None (uatcp-uasc-uabinary) ?i „Connect Server”:



- Se selecteaza „ServerInterfaces”, folderul cu denumirea tabelului ce contine variabilele exportat din PLC („proiect”) dupa care se copiaza variabilele in meniul „Data Access View”. In final, starea acestor variabile poate fi modificata, iar schimbarea este vizibila in PLC.



#	Server	Node Id	Display Name	Value	Datatype
1	SIMATIC.S7-120...	NS4 Numeric 2	led1	false	Boolean
2	SIMATIC.S7-120...	NS4 Numeric 3	led2	false	Boolean
3	SIMATIC.S7-120...	NS4 Numeric 4	led3	false	Boolean

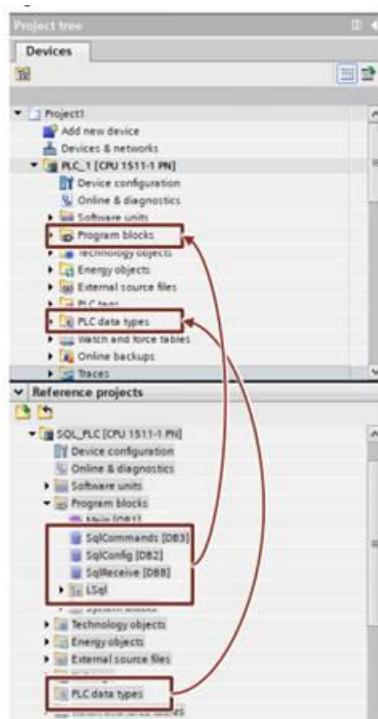
### 3. Conectarea PLC-ului la baza de date de tip SQL

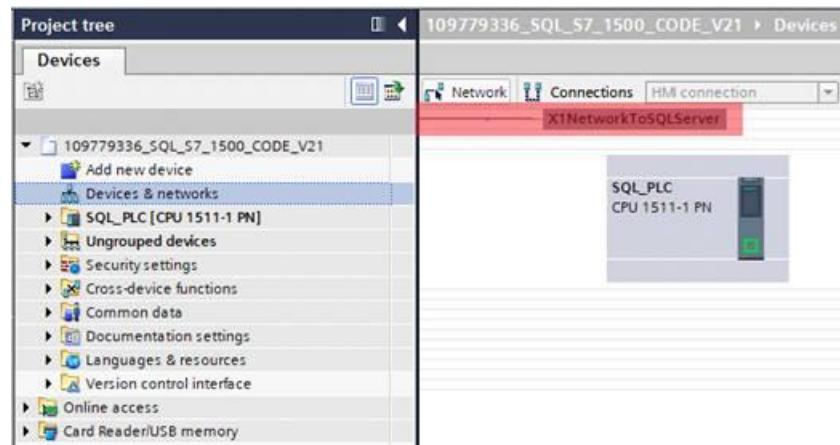
Capitol realizat cu contributia stud:Iordache Andra-Bianca- AIA IV

Se va realiza in continuare conectarea PLC-ului Siemens S7-1200 la o baza de date de tip SQL. Pentru a realiza conectarea s-a utilizat libraria LSQl, care poate fi descarcata de la adresa: <https://support.industry.siemens.com/cs/document/109779336/connecting-a-s7-1200-s7-1500-to-a-sql-database?-dti=0&lc=en-WW>

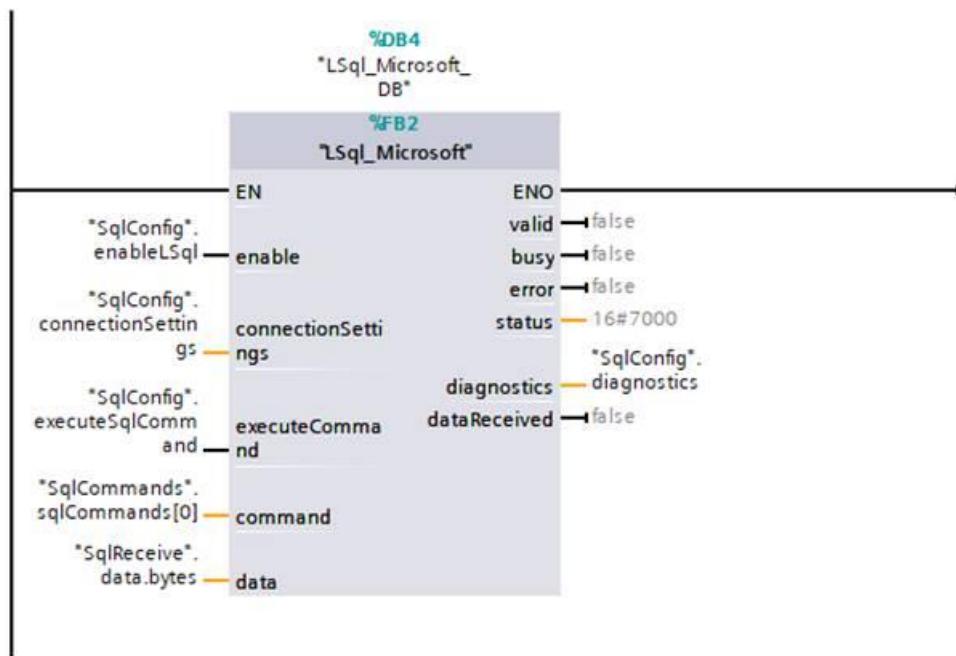
#### Configurarea PLC-ului:

1. Se descarca libraria LSQl
  2. Se creeaza un nou proiect in TIA Portal, unde se adauga configuratia PLC-ului. In paralel se dezarchivaza proiectul descarcat si se deschide. Din proiectul descarcat se copiază urmatoarele elemente in noul proiect creat:
    - "LSQl" block folder
    - Data block "SqlConfig"
    - Data block "SqlCommand"
    - Data block "SqlReceive"
    - PLC data types folder
    - X1NetworkToSQLServer (din meniul Device and Networks)





3. Blocul LSql\_Microsoft este apelat intr-un block ciclic (de exemplu Main) si se conecteaza intrarile si iesirile conform figurii urmatoare:



Acest bloc este controlat prin intermediul intrarilor "enableLSql" si "executeSqlCommand". Astfel cand intrarea "enableLSql" este setata "TRUE" se realizeaza conexiunea la baza de date iar cand "executeSqlCommand" este setata "TRUE" se executa query-ul selectat.

4. Se seteaza urmatorii parametrii in blocul de date SqlConfig:

Name	Data type	Start value	Retain	Accessible f...	Write ...	Visible in ...	Setpoint	Comment
connectionSettings	"LSql_typeConnecti...							
interfaceSettings	"TCP/IP_v4							
Interfaced	HWR_ANY	64						
ID	CONN_OUC	16#10						
ConnectionType	Byte	16#0B						
ActiveEstablish...	Bool	true						
RemoteAddress	IP_V4							
ADDR	Array[1..4] of Byte							
ADDR[1]	Byte	192						
ADDR[2]	Byte	168						
ADDR[3]	Byte	0						
ADDR[4]	Byte	109						
RemotePort	UInt	1433						
LocalPort	UInt	0						
loginInformation	"LSql_typeLogininf...							
hostname	String							
username	String	'SQL_571200'						
password	String	'sqlexpress'						
appName	String	'Database_Test'						
serverName	String	'SQLEXPRESS'						
libraryName	String							
local	String							
databaseName	String	'SQLFromPLC'						
spn	String							
attachDbFile	String							
changePassword	String							
enableLsql	Bool	false						
diagnostics	"LSql_typeDiagnos...							
status	Word	16#0						
subfunctionStatus	Word	16#0						
stateNumber	UInt	0						
executeSqlCommand	Bool	false						

## Configurarea Server-ului Microsoft SQL Server Express

Se descarca si se instaleaza Microsoft SQL Server basics de la adresa:

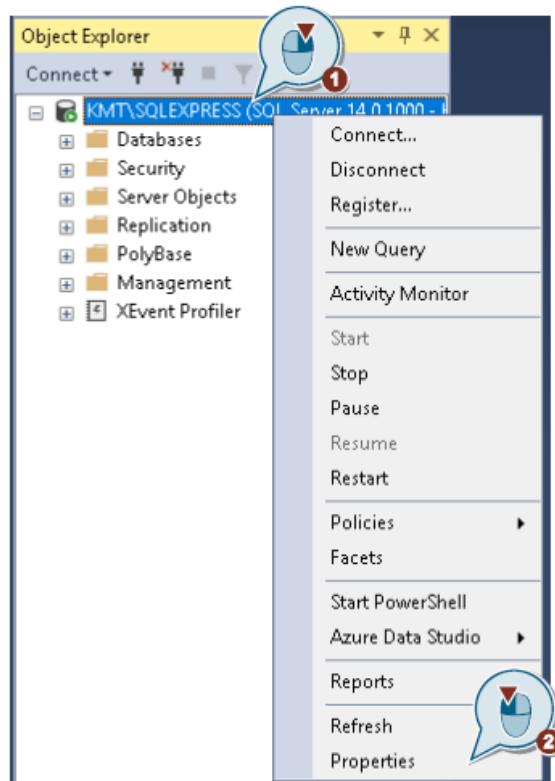
<https://www.microsoft.com/en-US/download/details.aspx?id=55994>

Se descarca si se instaleaza Microsoft SQL Server Management Studio

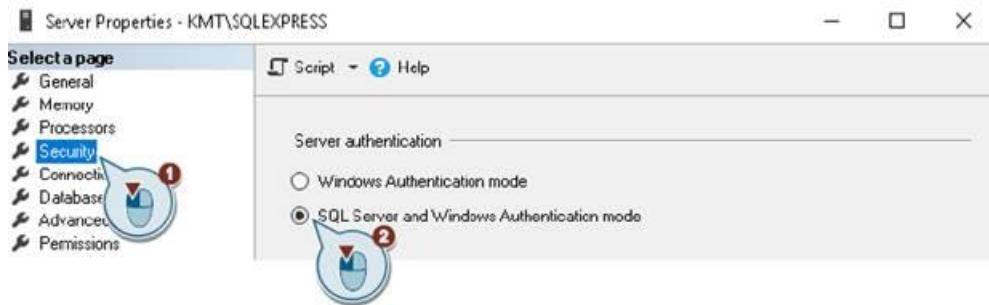
<https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver15>

## Logarea in SQL server

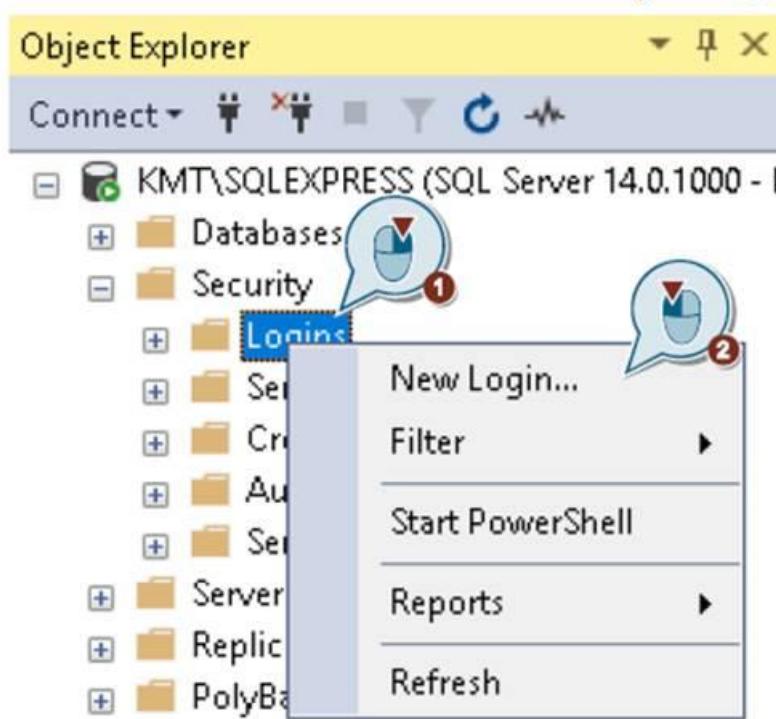
1. Porniti Microsoft SQL Server Management Studio
2. Deschideti meniu "Properties" a SQLEXPRESS



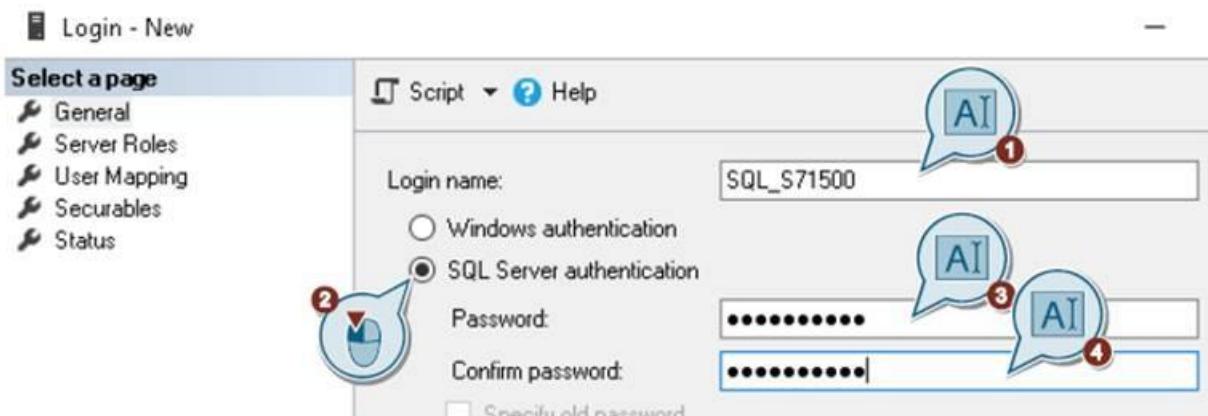
3. In meniul Security activati "SQL Server Authentication Mode" si confirmati modificarile



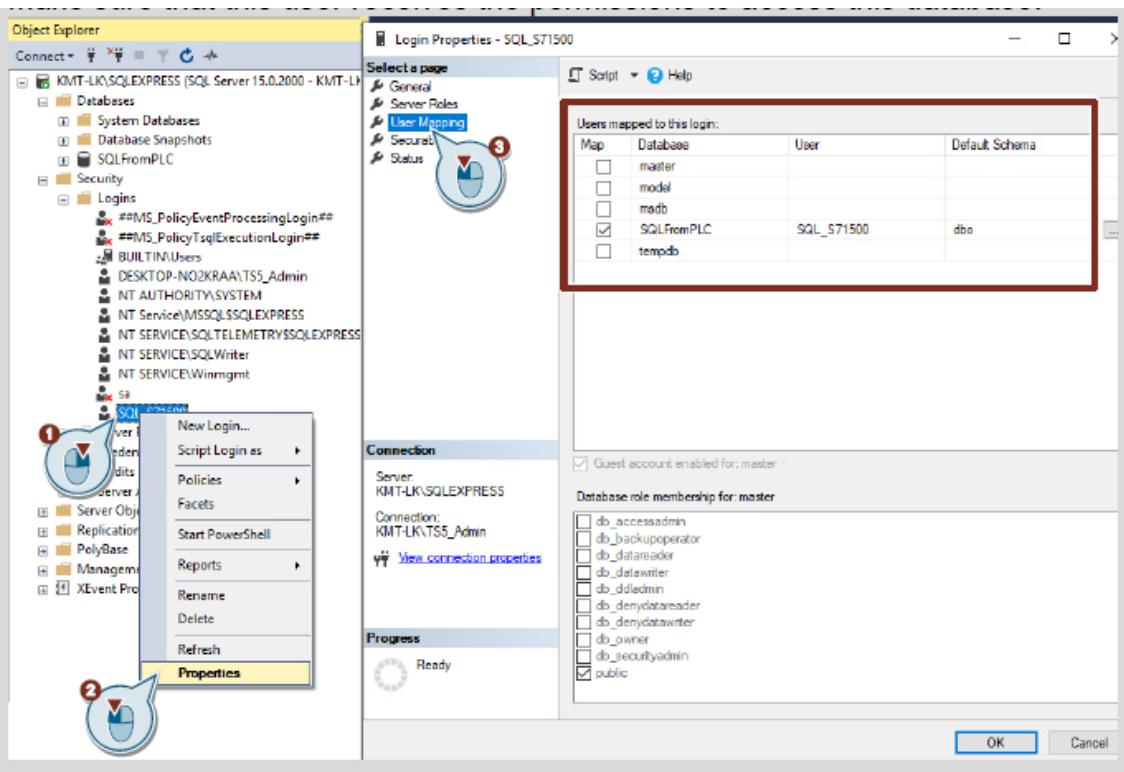
4. Creati un nou user: Security > Logins > New Login



5. Selectati SQL Server authentication si adaugati un user name si o parola

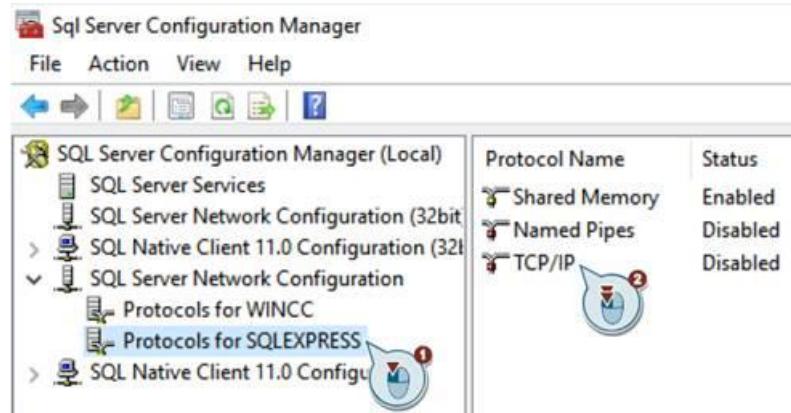


Asigurati-vă ca user-ul creat are dreptul de a accesa baza de date



## Setarea portului pentru SQL server

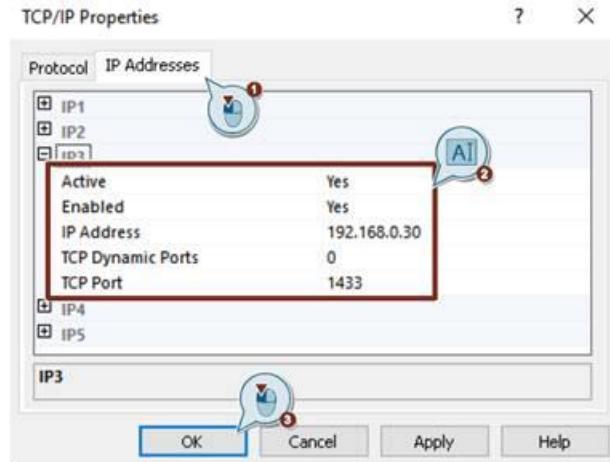
1. Se deschide "Microsoft SQL Server Configuration Manager" si se cauta "Protocols for SQLEXPRESS". Se deschide TCP/IP



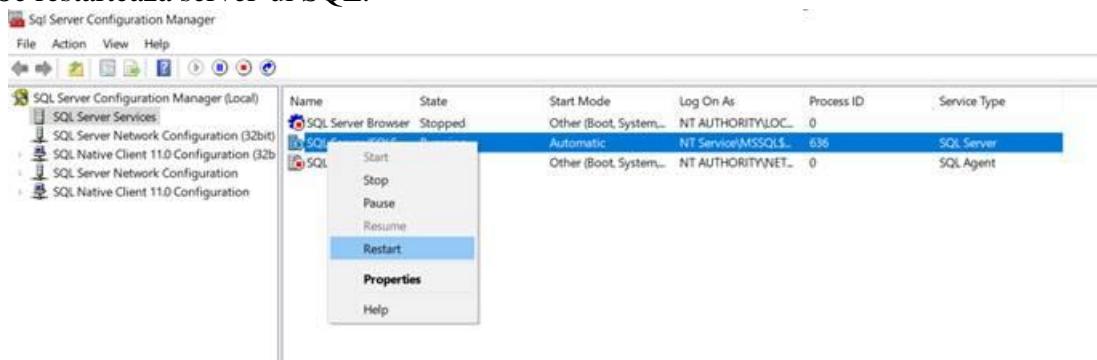
2. Se activeaza TCP/IP



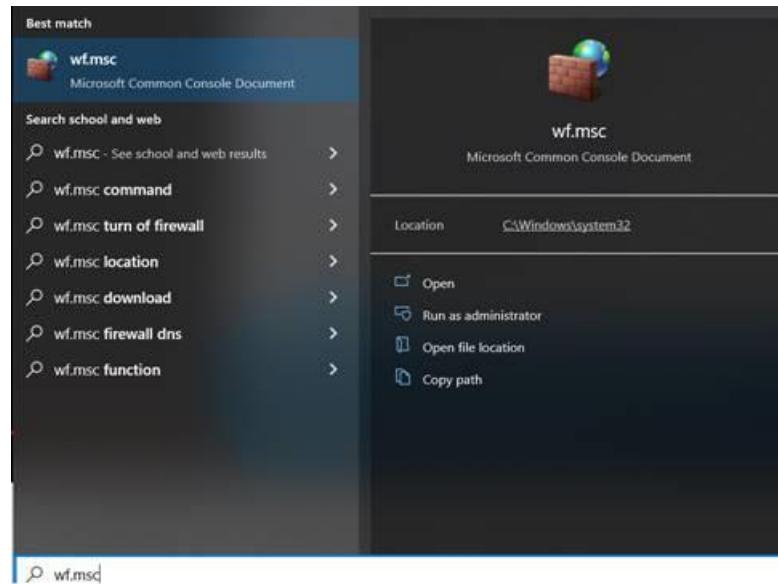
3. Se deschide IP3, se configureaza adresa IP si se seteaza portul 1433. Se activeaza si apasa OK.



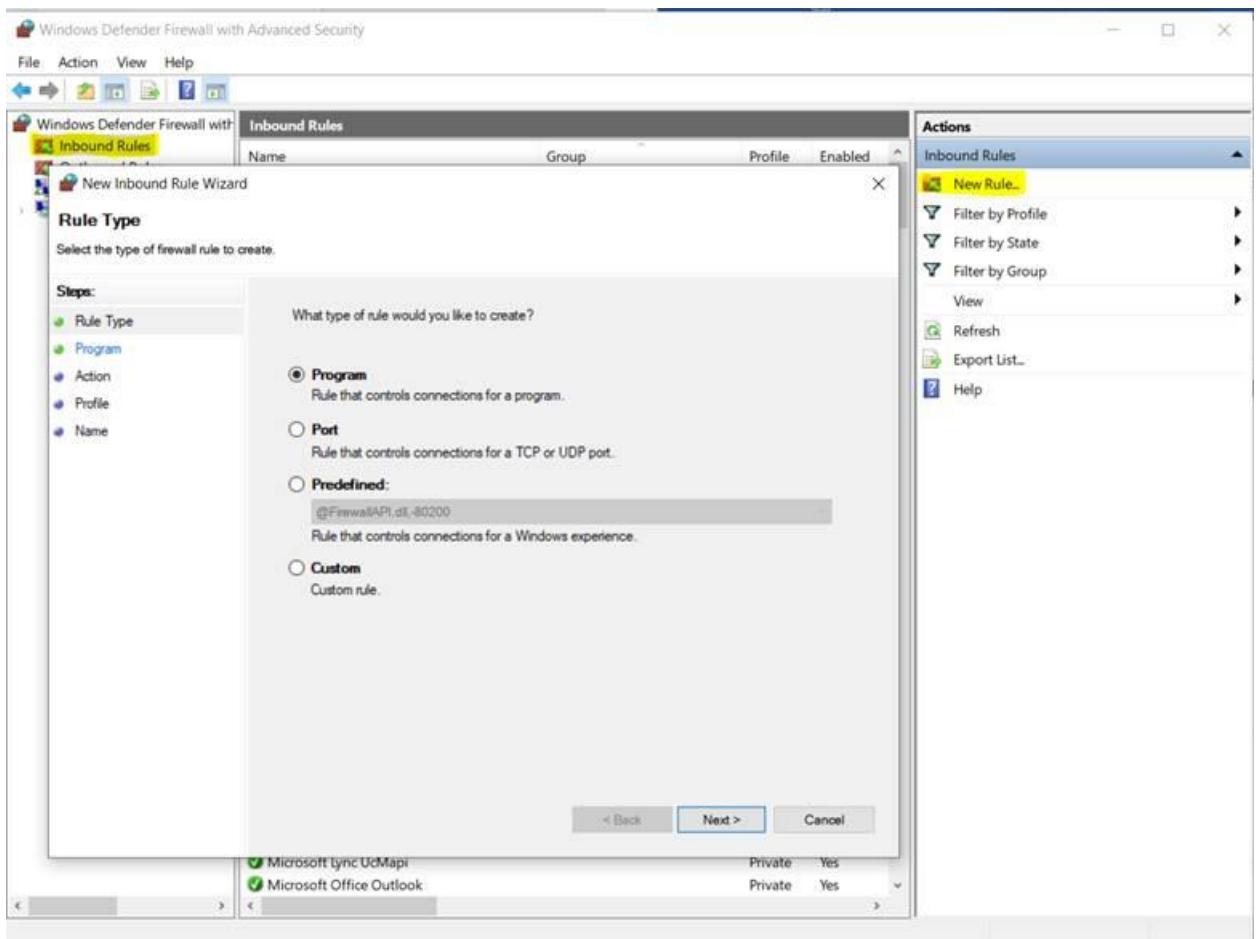
4. Se restarteaza server-ul SQL.



5. Se deschide wf.msc



6. Se adauga o noua regula in sectiunea “Inbound Rules” > “New Rule”



7. Se selecteaza “Port” > “TCP” > “Specific local ports: 1433” > “Allow the connection” > “Domain, Private, Public” > Finish

**Step 1: Rule Type**

- Program
- Port
- Predefined:
- Custom

**Step 2: Protocol and Ports**

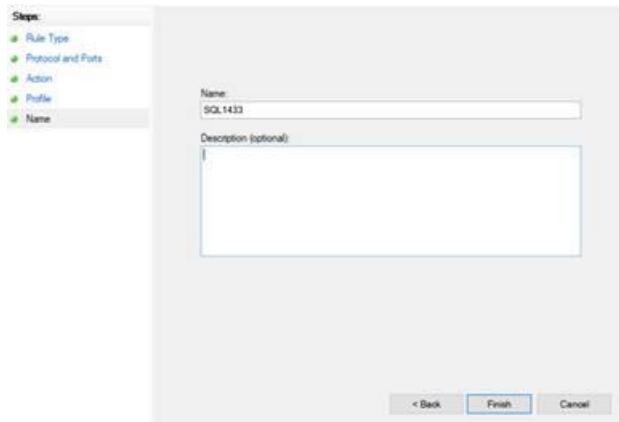
- TCP
- UDP

Does this rule apply to all local ports or specific local ports?

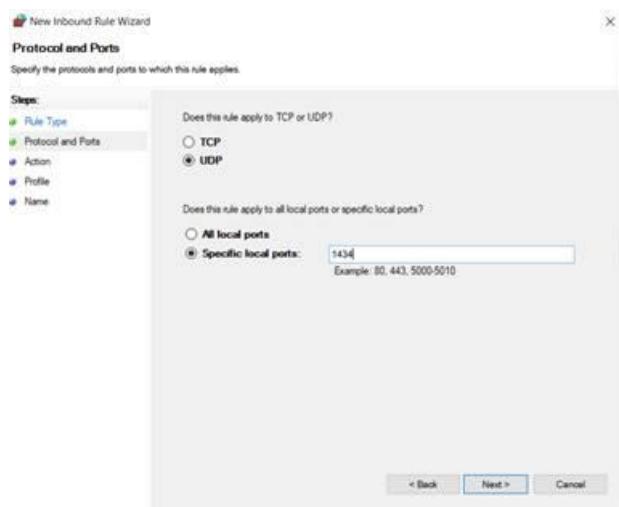
- All local ports
- Specific local ports: 1433

**Step 3: When does this rule apply?**

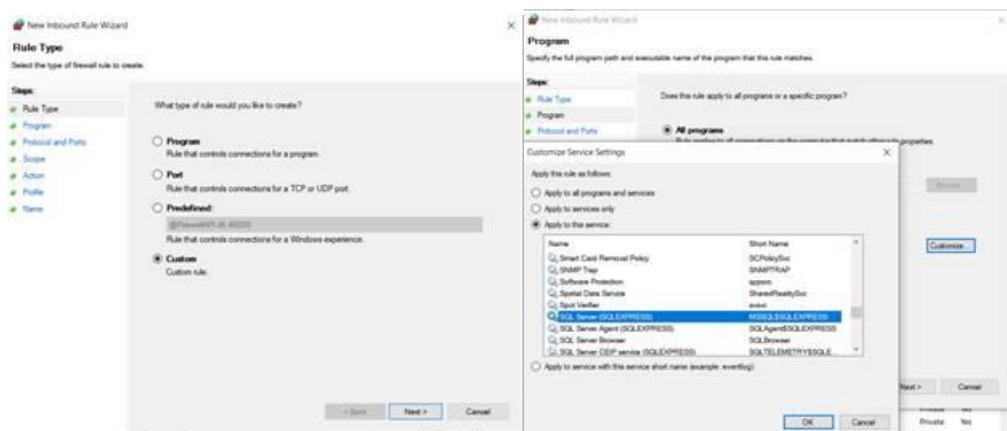
- Domain
- Private
- Public



8. Se repeta pasul 7, dar de data aceasta se creeaza o noua regula pentru portul 1434, UDP:



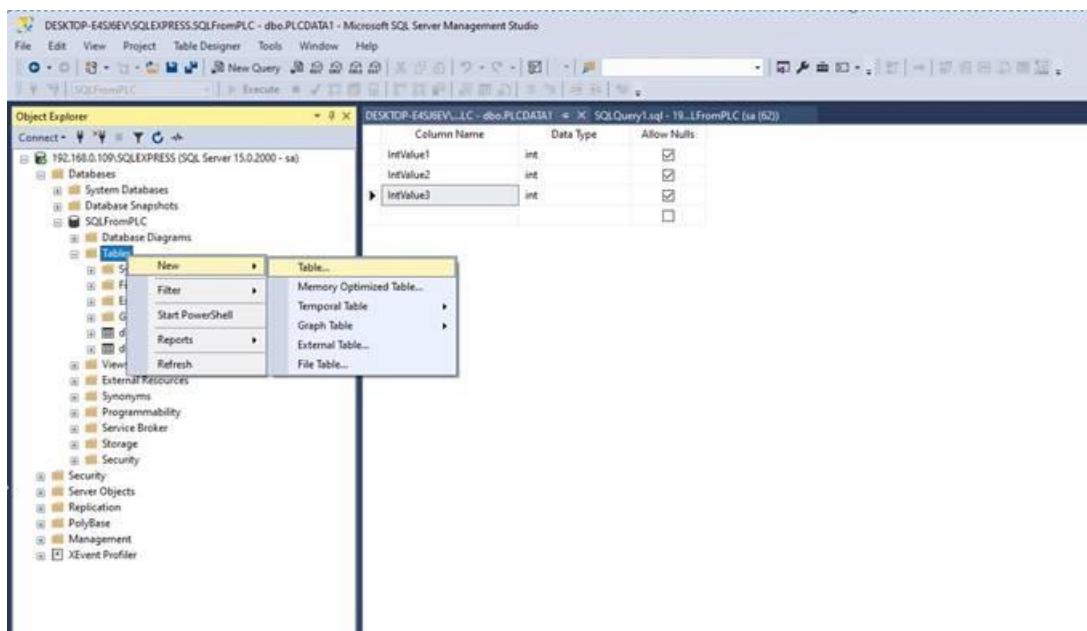
9. Urmatoare regula care trebuie creata este “Custom Rule”, astfel selectam “New Rule” >”Custom rule” > “Customize...” > “Apply to this service” . Se apasa OK si Next pana la final.



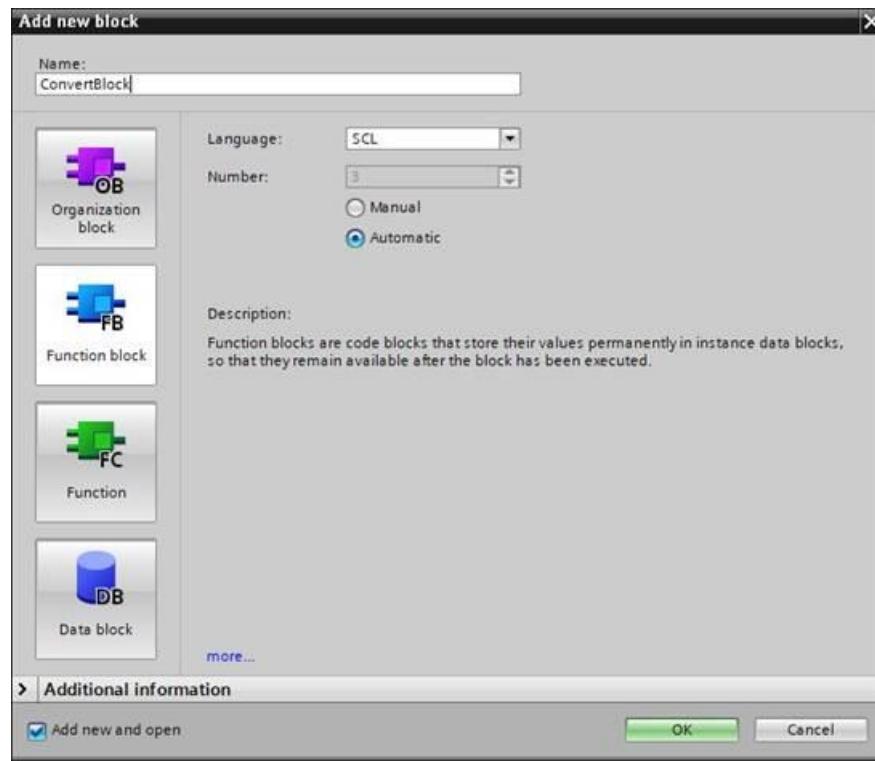
**Exemplu aplicatie de scriere in baza de date**

Dupa ce s-au realizat pasii descrisi anterior, se deschide Microsoft SQL Server Management Studio si se acceseaza baza de date creata

1. In interiorul bazei de date se creaza un nou tabel de forma celui prezentat in figura urmatoare, cu numele “PLCDATA1”



- 2.
3. In proiectul creat se adauga un bloc de tip “Function Block” iar ca limbaj se selecteaza SCL



4. Functia se completeaza cu urmatoarele date si se apeleaza in Main

```
#sep := ',';
#close := ')';
#data := 'insert into PLCDATA1 values(';
#data1str := INT_TO_STRING(IN := #data1);
#data2str := INT_TO_STRING(IN := #data2);
#data3str := INT_TO_STRING(IN := #data3);

#data := CONCAT_STRING(IN1 := #data, IN2 := #data1str);
#data := CONCAT_STRING(IN1 := #data, IN2 := #sep);
#data := CONCAT_STRING(IN1 := #data, IN2 := #data2str);
#data := CONCAT_STRING(IN1 := #data, IN2 := #sep);
#data := CONCAT_STRING(IN1 := #data, IN2 := #data3str);
#data := CONCAT_STRING(IN1 := #data, IN2 := #close);
```

**DataBase\_Test > PLC\_1 [CPU 1215C DO/DO/DC] > Program blocks > ConvertBlock [FB1]**

Name	Data type	Default value	Retain	Accessible f...	Write	Visible in ...	Setpoint	Comment
Input								
data1	int	0	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
data2	int	0	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
data3	int	0	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Output								
data	String	-	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
InOut								
-Add new...								
Static								
-Add new...								
Temp								
data1str	String							
data2str	String							
data3str	String							
sep	String							
close	String							
Constant								

```

CODE PAGE: 1252
OP: 1000.00 F: 1-1_NORM
1. keep := '';
2. EC009 := "1";
3. Data := "Insert into PLCDATA Values(";
4. #data1 := INT_TO_STRING(IE := #data1);
5. #data2 := INT_TO_STRING(IE := #data2);
6. #data3 := INT_TO_STRING(IE := #data3);

7. #data := CONCAT_STRING(IE := #data, IE2 := #data1);
8. #data := CONCAT_STRING(IE := #data, IE2 := #sep);
9. #data := CONCAT_STRING(IE := #data, IE2 := #data2);
10. #data := CONCAT_STRING(IE := #data, IE2 := #sep);
11. #data := CONCAT_STRING(IE := #data, IE2 := #data3);
12. #data := CONCAT_STRING(IE := #data, IE2 := #close);
13. Data := Data + #data;

```

**DataBase\_Test > PLC\_1 [CPU 1215C DO/DO/DC] > Program blocks > Main [OB1]**

Name	Data type	Default value	Comment
Input			
Initial_Call	Bool		Initial call of this OB

**Network 1:**

```

    *OB5
    "ConvertBlock_
    OB"
    "W1
    "Convertblock"
    EN
    "ConvertBlock_
    OB".data1 --- data1
    "ConvertBlock_
    OB".data2 --- data2
    "ConvertBlock_
    OB".data3 --- data3
    ENO
    "SqlCommands"
    data --- sqlCommands[0]

```

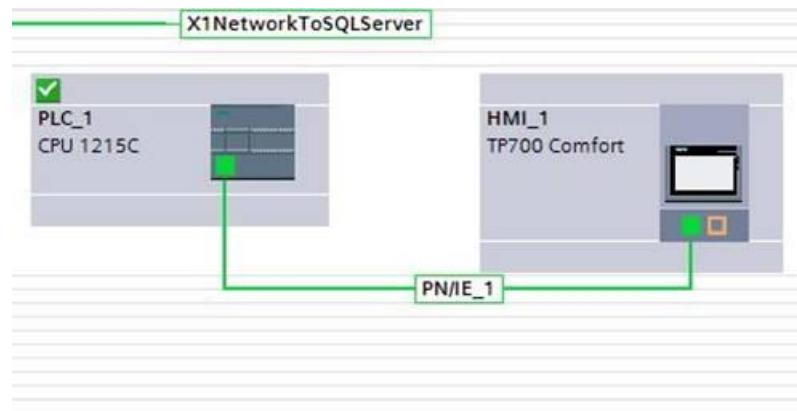
**Network 2:**

```

    *OB4
    "Lsql_Microsoft_
    OB"
    "W2
    "Lsql_Microsoft"
    EN
    "SqlConfig"
    enableLsql --- enable
    "SqlConfig".
    connectionSetting --- connectionSetting
    ENO
    void --- void
    busy --- busy
    error --- error
    status --- status

```

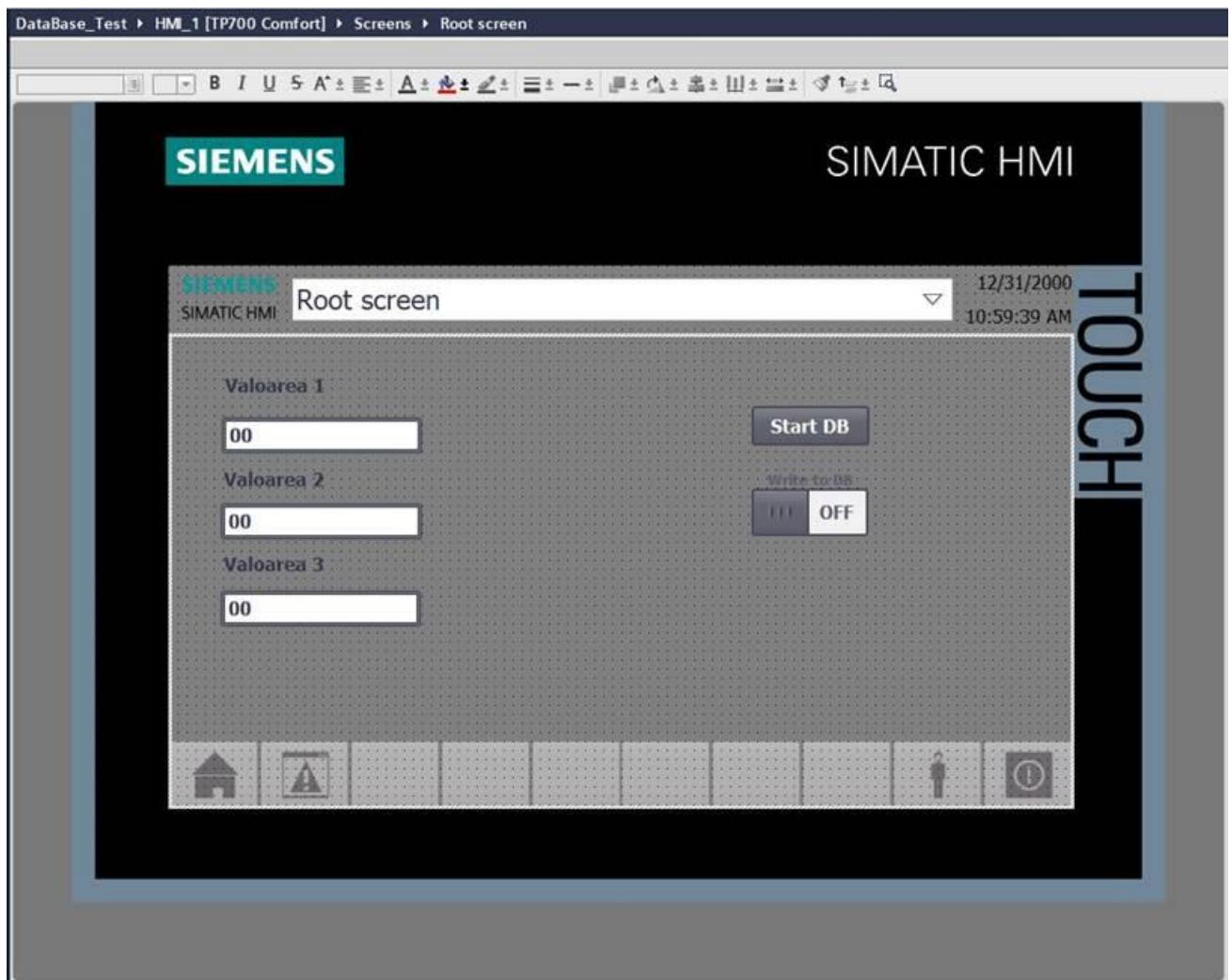
## 5. Se adauga HMI-ul in proiect si se conecteaza la PLC



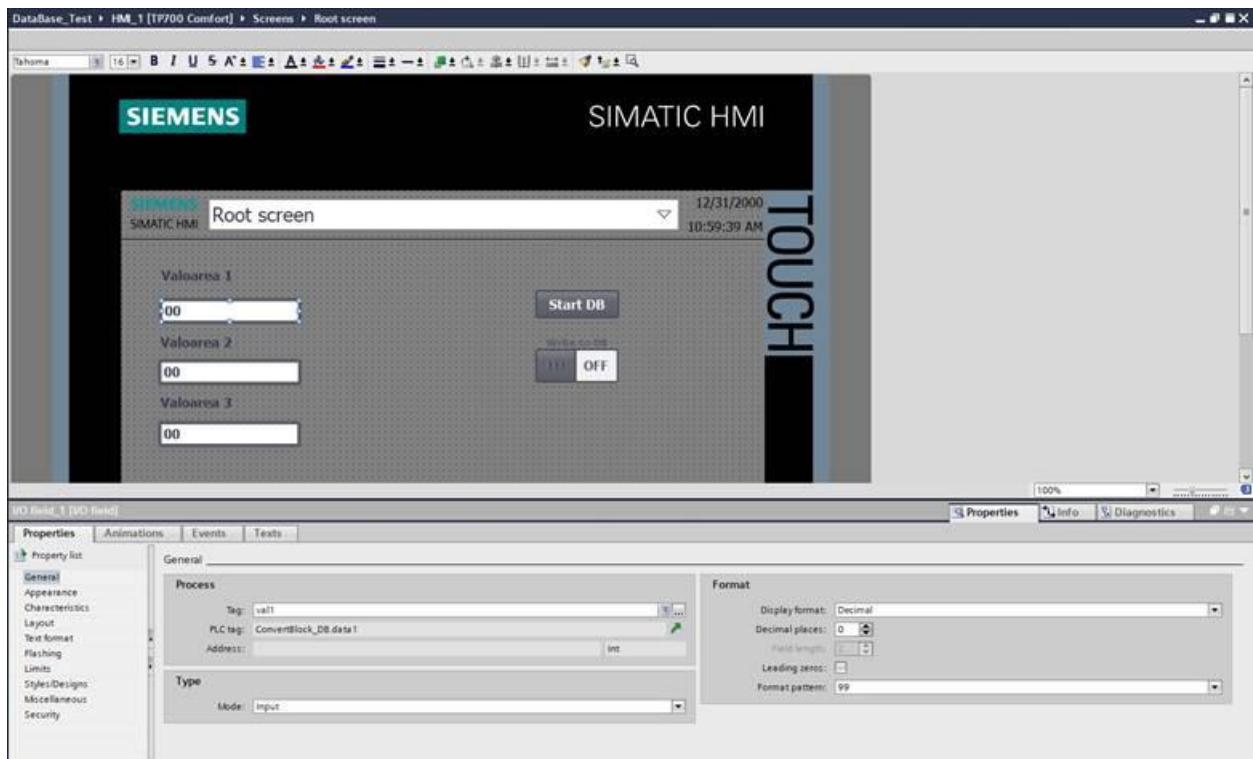
6. Se adauga urmatoarele HMI tags

HM tags										
Name	Tag table	Data type	Connection	PLC name	PLC tag	Address	Access mode	Acquisition cycle	Logged	Source comment
enable_db	Default tag table	Bool	HMI_Conne... PLC_1	PLC_1	SqlConfig enableSql	<symbolic access>	1:1	1:1	<input type="checkbox"/>	enables Ltg block (starts connection and log)
Tag_ScreenNumber	Default tag table	UInt	-internal tag-						<input type="checkbox"/>	
val1	Default tag table	Int	HMI_Connectio... PLC_1	PLC_1	ConvertBlock_DB_data1	<symbolic access>	1:1	1:1	<input type="checkbox"/>	
val2	Default tag table	Int	HMI_Connectio... PLC_1	PLC_1	ConvertBlock_DB_data2	<symbolic access>	1:1	1:1	<input type="checkbox"/>	
val3	Default tag table	Int	HMI_Connectio... PLC_1	PLC_1	ConvertBlock_DB_data3	<symbolic access>	1:1	1:1	<input type="checkbox"/>	
write_to_db	Default tag table	Bool	HMI_Connectio... PLC_1	PLC_1	SqlConfig executeSqlCommand	<symbolic access>	1:1	1:1	<input type="checkbox"/>	trigger to execute a single sql command. Ena
<Add more>										

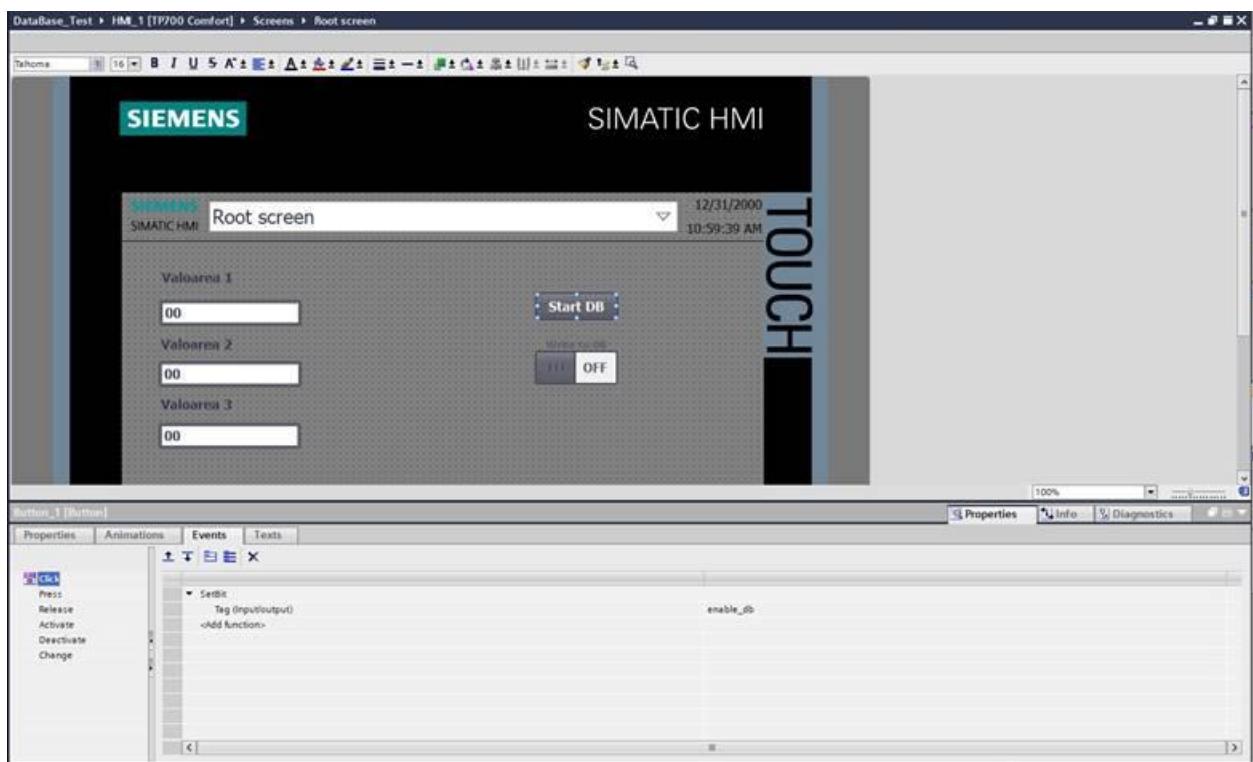
7. In Root Screen de adauga urmatoarele elemente



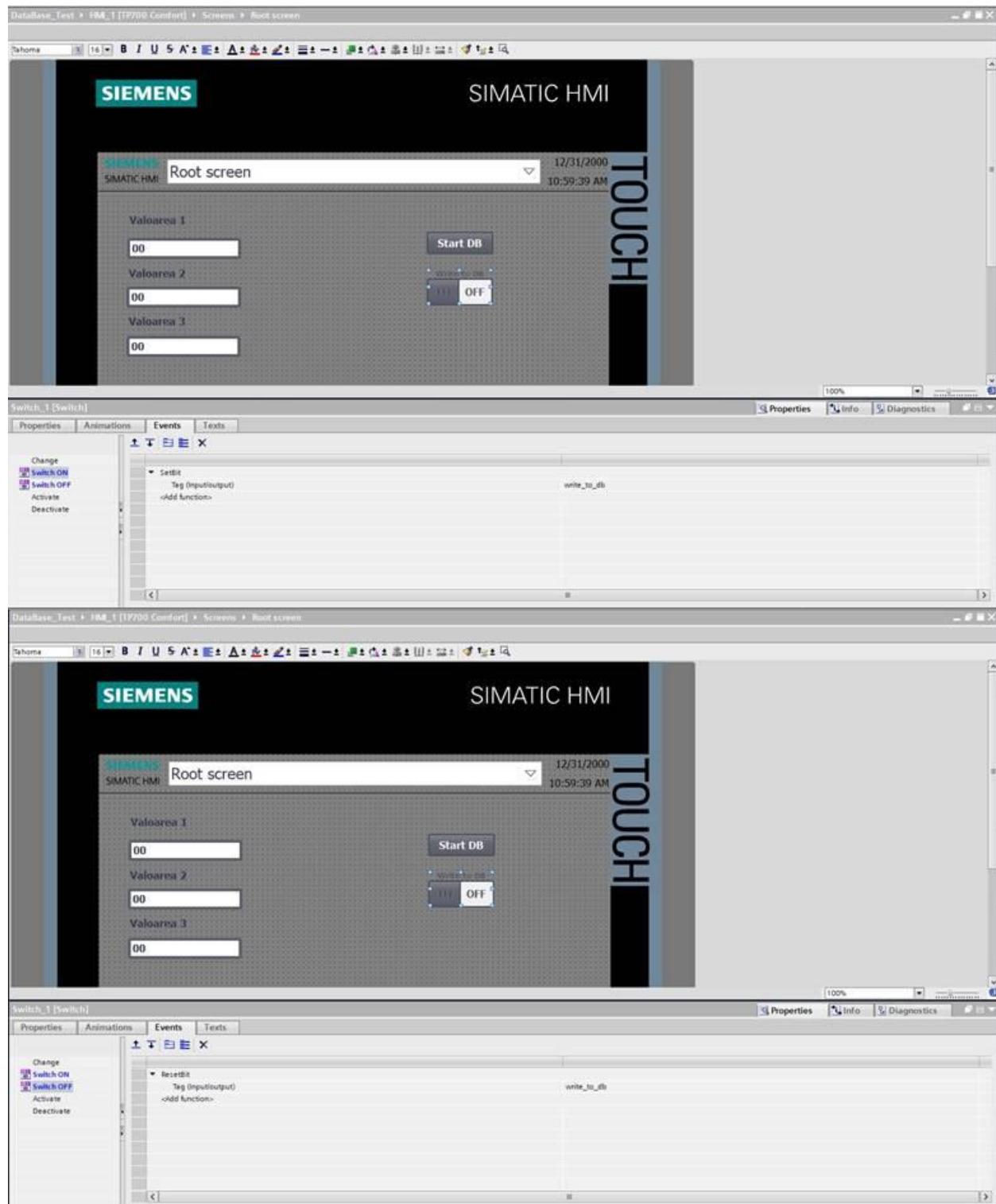
8. Elementele de tip "I/O Field" se conecteaza cu datele de intrare ale functiei create si se modifica tipul elementului > "Input"



9. Pentru butonul de “Start” se adauga un eveniment de tip “Click”, pe care se adauga functia “SetBit” la care ii se atribuie tag-ul “enable\_db”. Acest tag se conecteaza cu tag-ul din PLC conform figurii.



10. Pentru switch se adauga pe evenimentul "Switch ON" functia SetBit si ii se atribuie tag-ul write\_to\_db. Pe evenimentul "Switch OFF" se adauga functia "ResetBit" careia ii se atribuie tag-ul "write\_to\_db"



11. Se incarca programul creat in PLC si in HMI si se testeaza astfel: se apasa butonul “Start DB” > se inscriu valorile dorite > se comuta switch-ul in pozitia ON > se cauta inregistrarile in baza de date.

## Rezumat

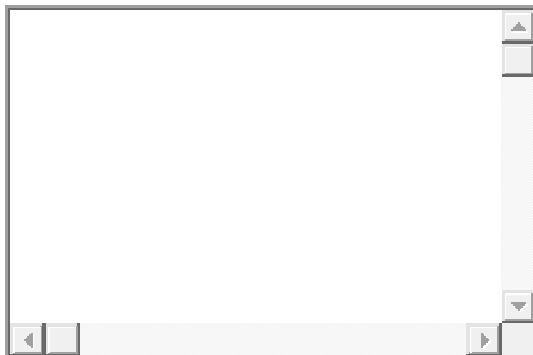
- **Web server**

Formatul comenzii AWP pentru declararea unei variabile:

```
<!-- AWP_In_Variable Name='''Nume_var''' -->
```

Toate variabilele citite din CPU trebuie sa aiba forma:

```
:='''Nume_var'''
```



- **Rezultate asteptate**

Dupa studierea acestui modul, ar trebui sa cunoasteti:

- Cum sa configurati Web\_server-ul
- Cum sa realizati o pagina web HTML in care sa includeti comenzii AWP
- Sa realizati aplicatii WEB-SCADA

- **Termeni esentiali**

Termen	Descriere
SCADA	Supervisory Control And Data Aquisition

OPC	Open Platform Communications
SQL	Structured Query Language
SCL (Structured Control Language)	Limbaj pentru programarea PLC-urilor de tipul SIMATIC S7-1200/1500

### • Recomandari bibliografice

- [1] T. Turc - Sisteme SCADA, Ed. Univ. "Petru Maior", ISBN: 978-606-581-110-2 , 2013
- [2] T. Turc - Aplicatii SCADA, Ed. Univ. " Petru Maior", ISBN: 978-606-581-109-6 , 2013
- [3] T. Turc - Programarea microprocesoarelor din familia X86:, Ed. Univ. "Petru Maior", ISBN: 978-606-581-026-6, 2011
- [4] T. Turc - Tehnologii WEB:, Ed. Univ. "Petru Maior", ISBN: 978-973-755-576-2, 2010
- [5] T. Turc - Informatica aplicata in ingineria electrica, ISBN: 978-973-169-700-0, Ed. univ. UMFST, Tg. Mures, 2021.
- [6] T. Turc - Programare avansata C++ pentru ingineria electrica, ISBN: 978-973-755-588-5, Ed. MatrixRom, 2009.
- [7] T. Turc - Elemente de programare C++ utile in ingineria electrica, ISBN: 978-973-755-576-2, Ed. MatrixRom, 2009
- [8] B. Barbat - Informatica industriala - Programarea în timp real – Institutul Central pentru Conducere si informatica 1984
- [9] I. Babuita – Conducerea automata a proceselor – Ed. Facla 1985

### • Link-uri utile

- <https://support.industry.siemens.com/cs/ww/en/view/109755216> - SIMATIC WinCC V15.1 - Programming reference - 2021 -
- <https://support.industry.siemens.com/cs/us/en/view/109755202> - STEP 7 and SIMATIC WinCC V15.1 System Manual - 2021 -
- <https://support.industry.siemens.com/cs/ww/en/view/81318674> - Programming for SIMATIC S7-1200 and S7-1500 - 2021 -
- <https://support.industry.siemens.com/cs/document/39710145> - SIMATIC S7-1200 Easy Book - 2021 -
- <https://support.industry.siemens.com/cs/ww/en/view/68011496> - Creating and using user-defined web pages on S7-1200 / S7-1500 -
- [S7-1200\\_1500\\_Webserver\\_DOC\\_v4\\_en.pdf](#) - Creating user-defined web pages for S7-1200 / S7-1500 - 2021 -
- [SQL\\_S7\\_1500\\_DOC\\_en\\_V21.pdf](#) - Connecting a S7-1500 to a SQL Database -