

Aplicatii in energetica

- **Obiective**

- Realizarea de aplicatii SCADA in energetica.
- Definirea simbolurilor grafice necesare dezvoltarii de aplicatii SCADA destinate sectorului energetic.
- Definirea elementelor unei linii monofilare.
- Aplicatii SCADA pentru simularea unei linii monofilare.
- Aplicatii SCADA pentru simularea unei linii monofilare si implementarea algoritmilor de conditionalitate intre elementele liniei monofilare.
- Aplicatii SCADA pentru simularea conectarii secventiale a liniilor monofilare.
- Aplicatii SCADA pentru managementul liniilor monofilare.

- **Organizarea sarcinilor de lucru**

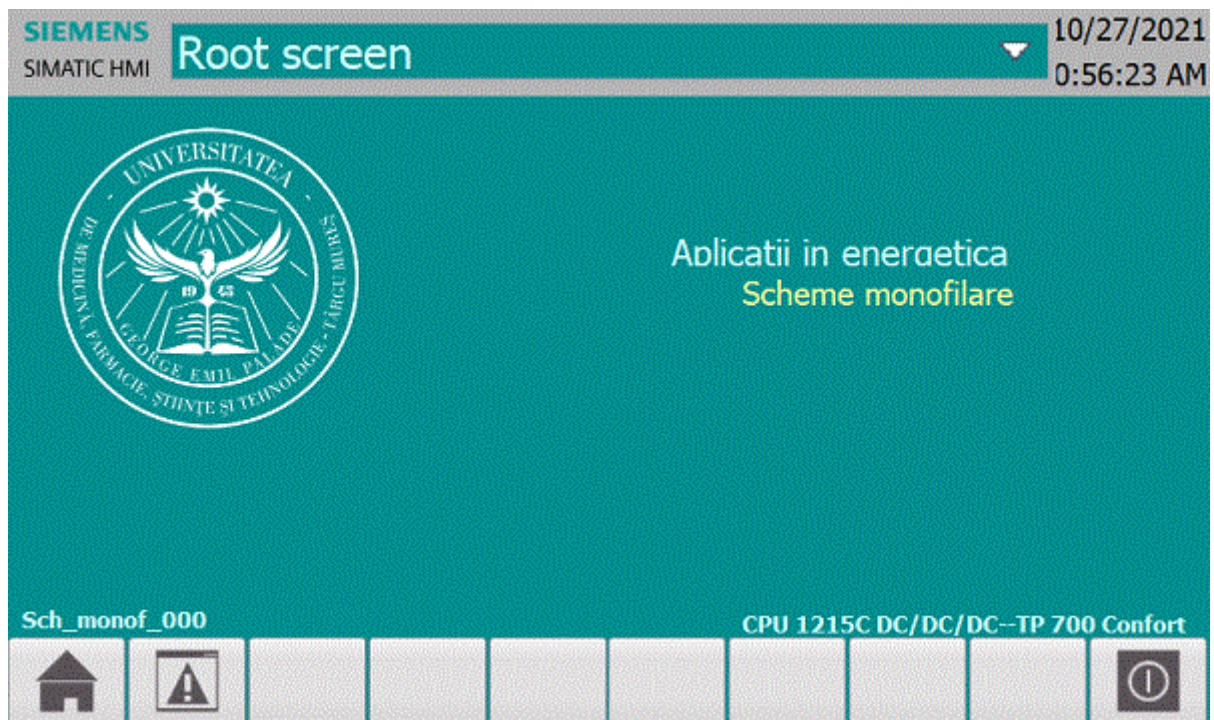
- Parcurgeti cele trei capitole ale cursului.
- In cadrul fiecarui capitol urmariti exemplele ilustrative si incercati sa le realizati in medul de dezvoltare SCADA TIA Portal.
- Fixati principalele idei ale cursului, prezentate în rezumat.
- Completati testul de autoevaluare.
- Timpul de lucru pentru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

1. Scheme monofilare - elemente introductive

Sectorul energetic este unul din sectoarele care necesita control si monitorizare la diverse nivele. Vom realiza in continuare o serie de aplicatii SCADA in domeniul energetic.

- **Crearea proiectului: "Scheme monofilare"**

Vom dezvolta in continuare cateva aplicatii din domeniul energetic in cadrul unui proiect numit Sch_monof_000



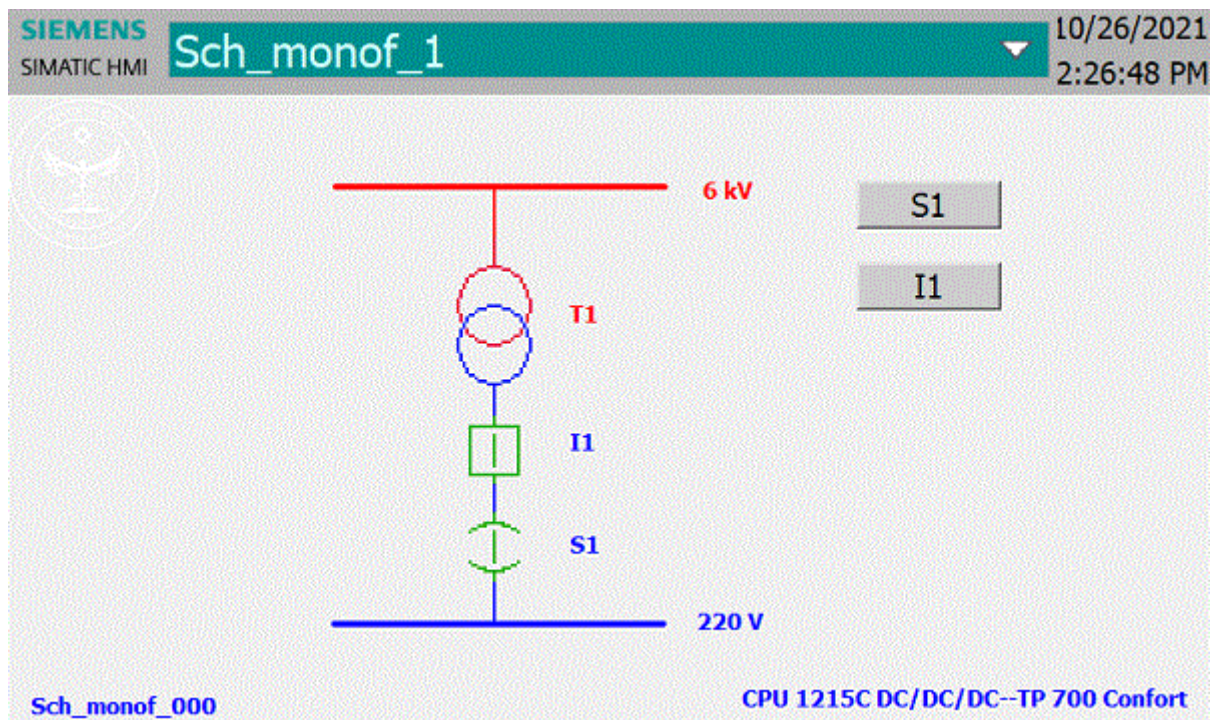
In cadrul acestui proiect vom realiza diverse scheme monofilare carora le vom da functionalitate. Pentru a realiza scheme monofilare avem nevoie de o serie de simboluri cum ar fi simboluri pentru: separatoare, intreruptoare, transformatoare etc.

Vom folosi urmatoarele simboluri:

intr_on	intr_on_v	intr_off	intr_off_r	sep_on	sep_on_v
sep_off	sep_off_r	contact_on	contact_off	trafo_on	trafo_off

- **Elementele unei linii monofilare**

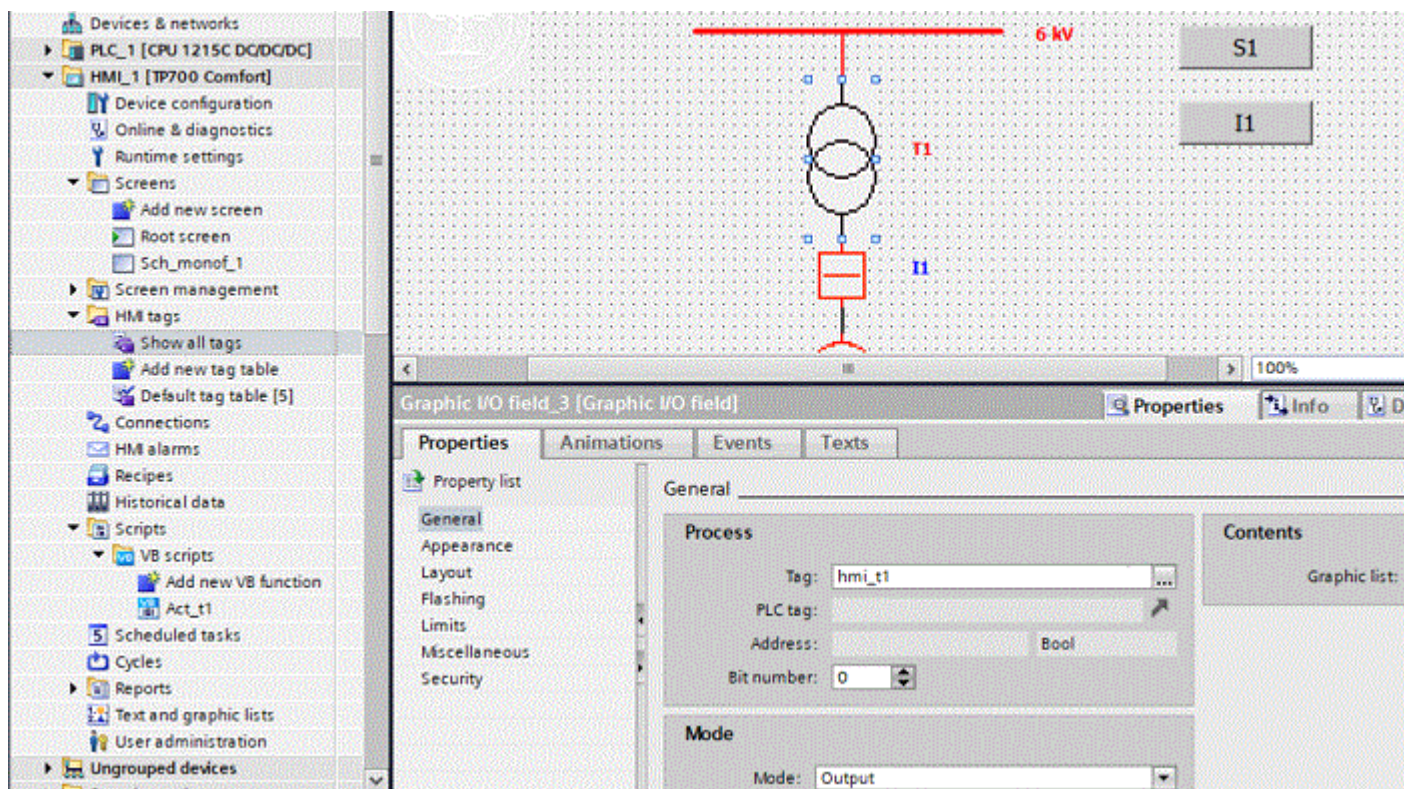
Vom defini in cadrul proiectului "Sch_monof_000" un nou screen numit "Sch_monof_1" in care vom plasa un separator, un intrerupator si un transformator, realizand schema monofilara de mai jos.



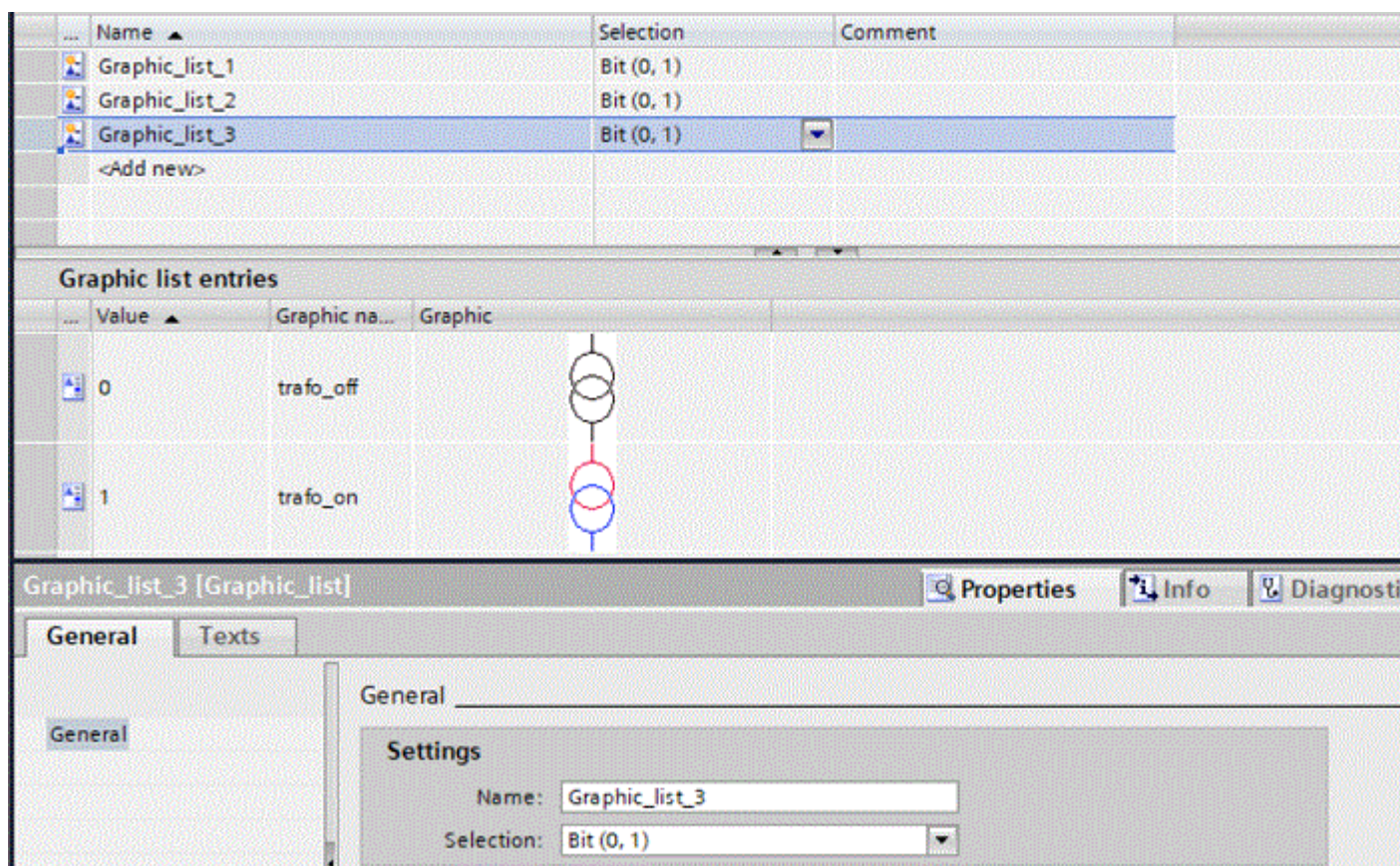
Pentru realizarea aplicatiei avem nevoie de urmatoarele HMI tags

HMI tags				
Nume	Tip	Domeniu	Um	Comentariu
hmi_i1	Bool	-	-	Interrupator I1
hmi_s1	Bool	-	-	Separator S1
hmi_t1	Bool	-	-	Transformator T1

Pentru transformator s-a utilizat un obiect "Graphic I/O field" caruia i s-a atribuit tag-ul "hmi_t1" si modul "Output".



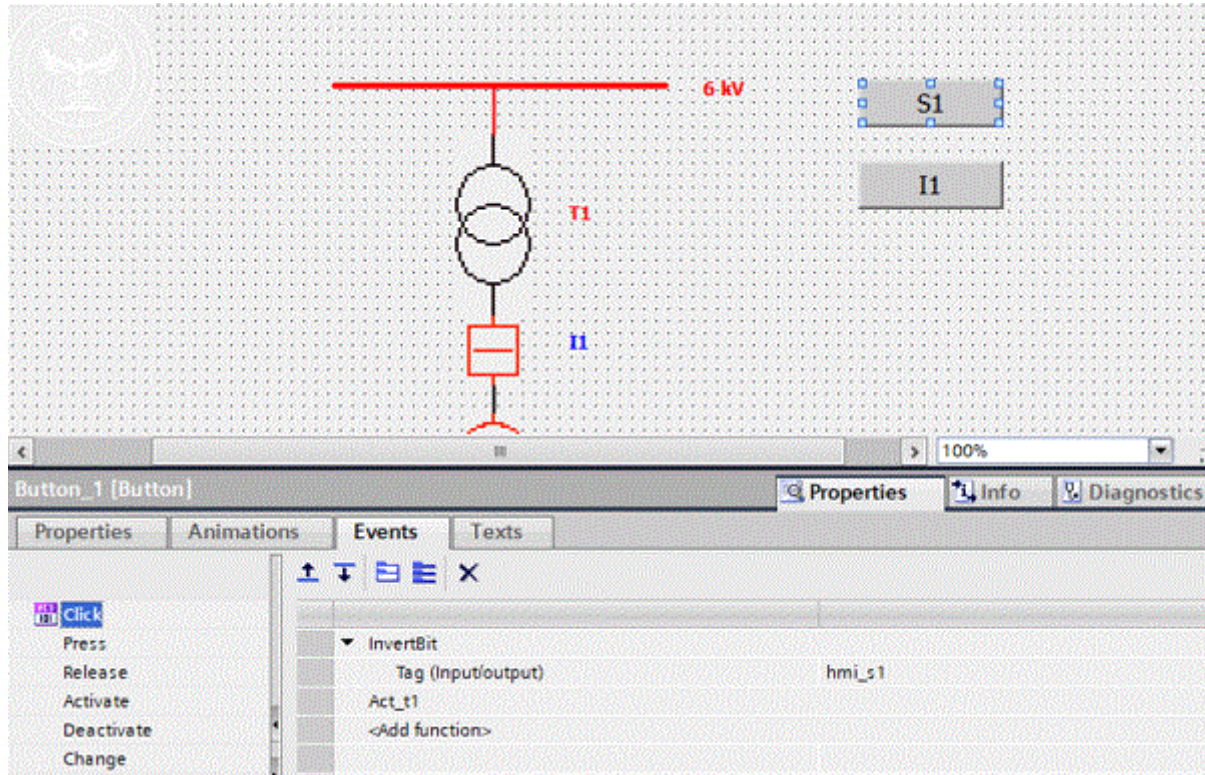
Lista grafica "Graphics_list_3" s-a setat prin apasarea sagetii incercuite cu rosu din imaginea de sus.



In lista grafica "Graphics_list_3" s-au adaugat imaginile coresunzatoare respectiv "trafo_off" si "trafo_on".

Similar, pentru intreruptor s-au utilizat imaginile "intr_off" si "intr_on" iar pentru separator s-au utilizat imaginile "sep_off" si "sep_on".

Pe evenimentul "Click" al butonului S1, pe langa operatia "InvertBit" asupra tag-ului "hmi_s1" se lanseaza si script-ul "Act_t1" care actualizeaza starea tag-ului "hmi_t1"



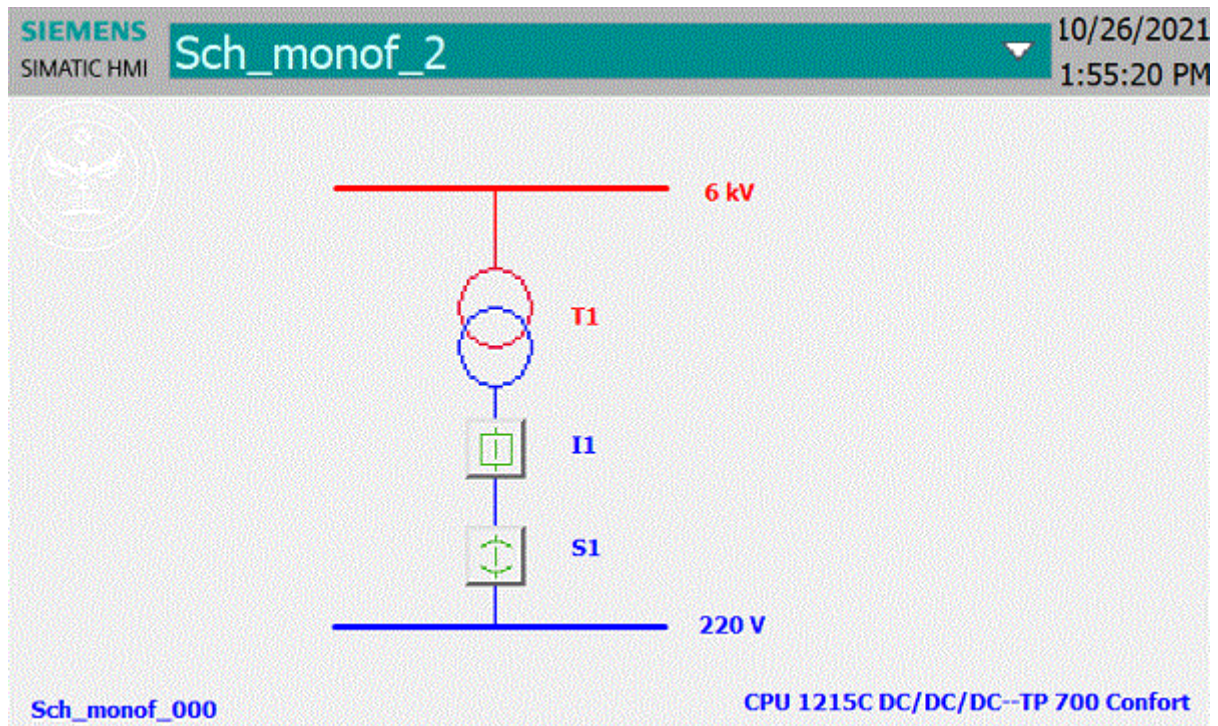
```
Sub Act_t1()  
    If hmi_s1 And hmi_i1 Then  
        hmi_t1=True  
    Else  
        hmi_t1=False  
    End If  
End Sub
```

1. Functionarea unei lini monofilare

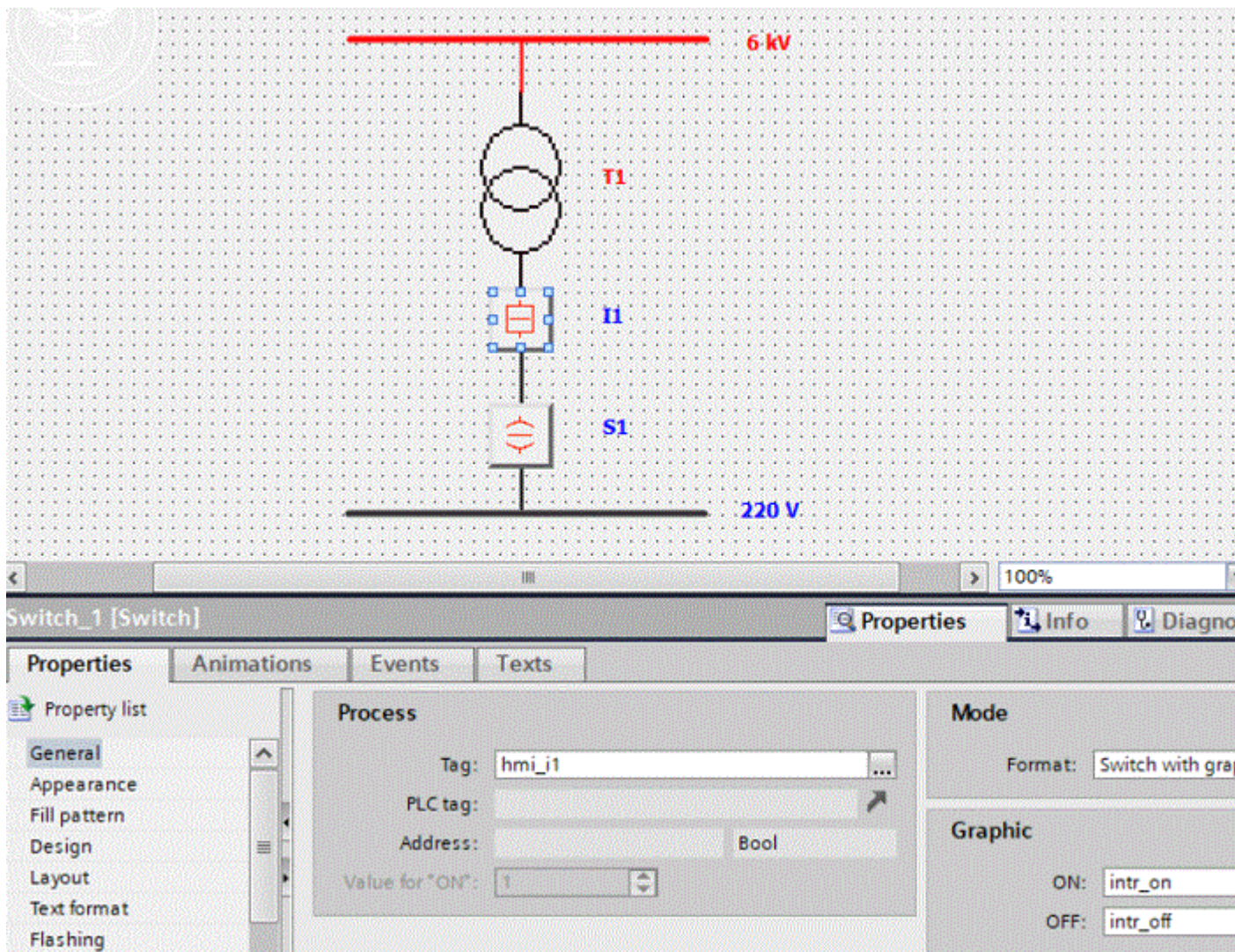
Vom introduce in continuare o serie de elemente care sa faca functionala linia monofilara schitata anterior.

Sa introducem deci noi facilitati si sa adaugam comenzi directe pentru inchiderea respectiv deschiderea separatorului si intreruptorului, realizand astfel un nou screen **sch_monof_2** . Putem imbogati functionalitatea schemei precedente cu facilitati de comanda pe langa

facilitatile anterioare de monitorizare. Vom da deci o dubla functionalitate celor doua simboluri (S1, I1) atat pentru afisarea starii cat si pentru preluarea comenzii de inchidere respectiv deschidere.



Pentru (S1, I1) s-au folosit obiecte "Switch" setate astfel:



Cu alte cuvinte, se inglobeaza functionalitatea butonul S1 si I1 in simbolurile S1 si I1.

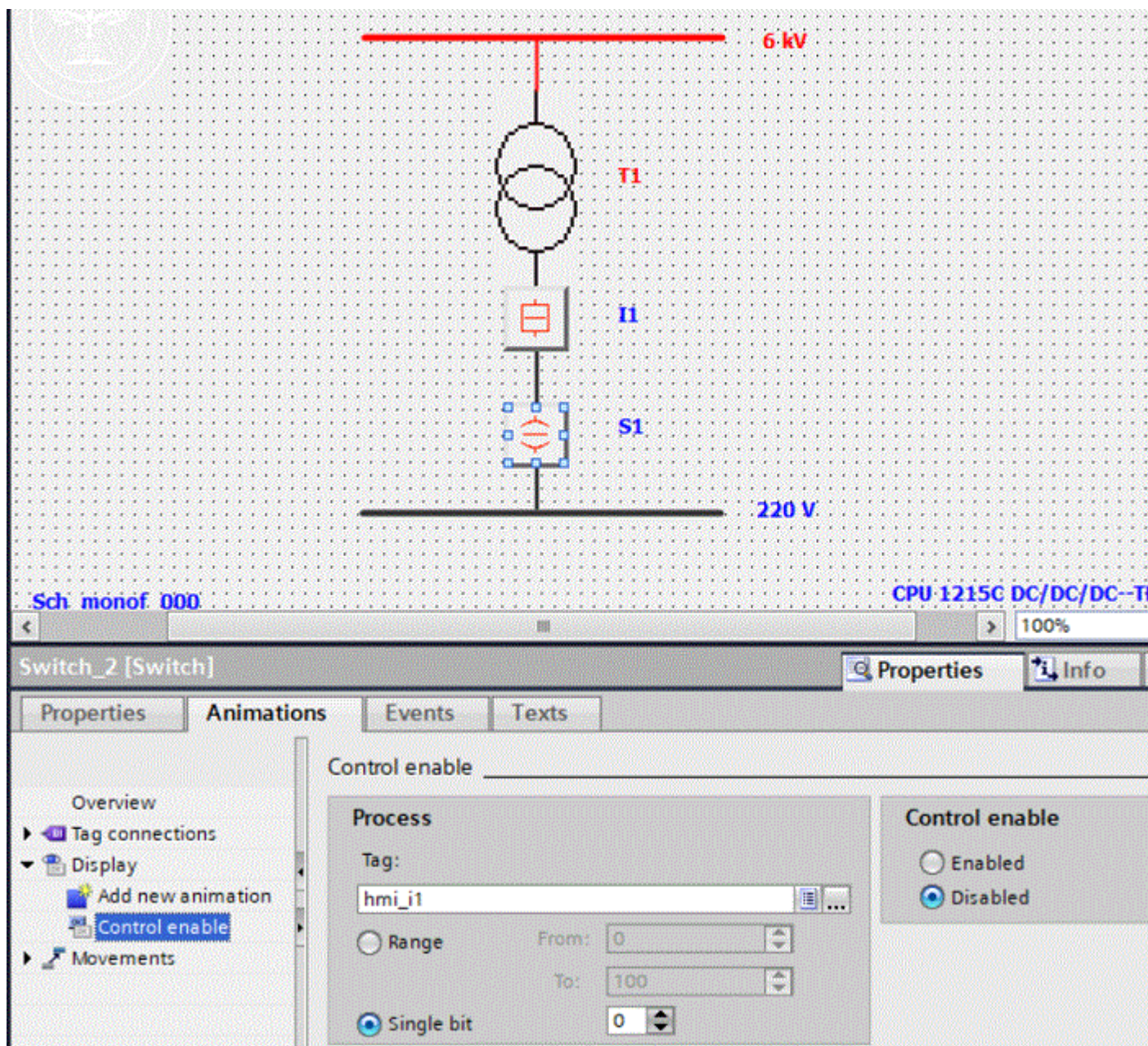
Am eliminat cele doua butoane ce simuleaza starea celor doua elemente, comanda lor, facandu-se direct de la acestea.

Cand cele doua elemente sunt inchise, simbolul transformatorului se modifica indicand functionarea acestuia, prin atribuirea script-ului "Act_t1" (care actualizeaza starea tag-ului "hmi_t1") pe evenimentul "Change"

In aplicatiile reale exista o ordine in care se actioneaza cele doua elemente, in sensul ca separatorul nu poate fi niciodata actionat in sarcina. Acesata schema nu asigura aceasta constrangere, nefiind conditionalitate intre cele doua comenzi.

- **Conditionarea separatorului de intreruptor**

Pentru a realiza conditionalitatea intre cele doua comenzi, in sensul ca separatorul nu poate fi niciodata actionat in sarcina, in urmatorul screen **sch_monof_3** se va adauga la S1 Animation --> Control enable --> Tag : "hmi_i1" pentru invalidarea butonului S1 cand e actionat I1.



• Confirmarea executiei comenzilor

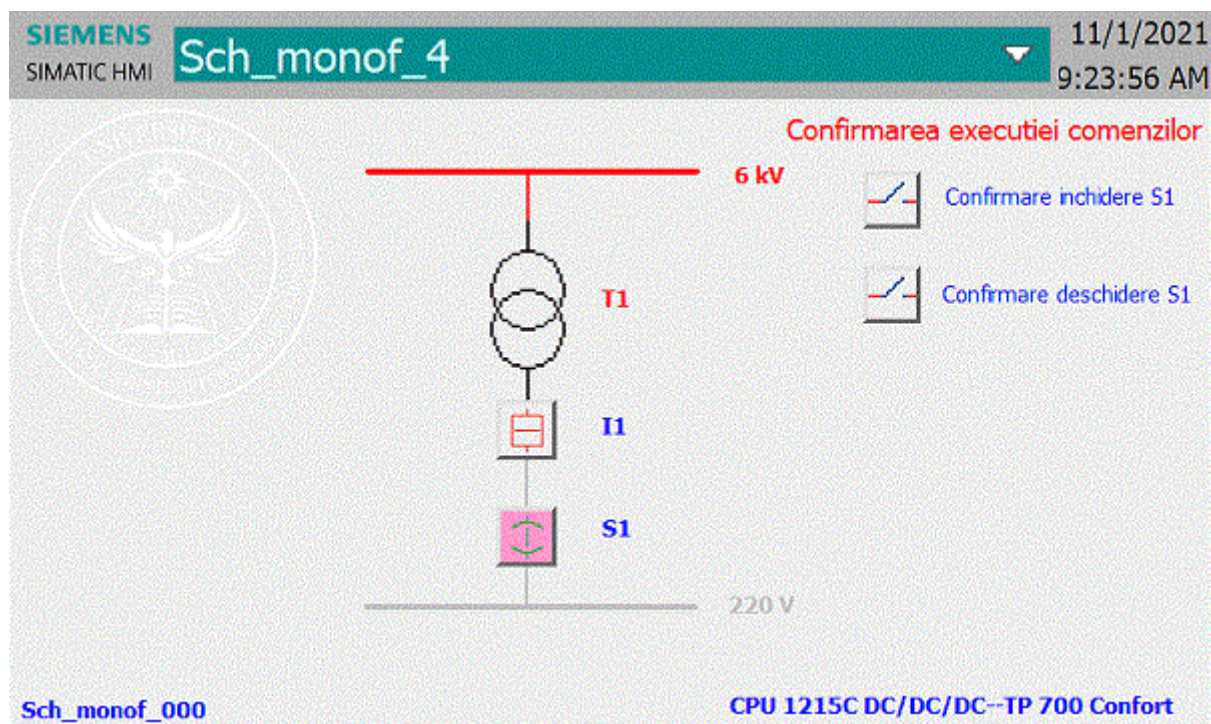
În sistemele energetice, închiderea respectiv deschiderea unui separator sau a unui întrerupător nu este instantanee. Vom ține cont la următorul screen **sch_monof_4** de acest fenomen. După comanda acționării unui element se scurge un interval de timp până la efectuarea completă a comenzii. În aplicația curentă, după acționarea separatorului S1, vom avea nevoie de un semnal care să confirme efectuarea comenzii.

Vom introduce pe lângă tag-urile existente, noi tag-uri pentru a implementa confirmarea efectuării comenzii.

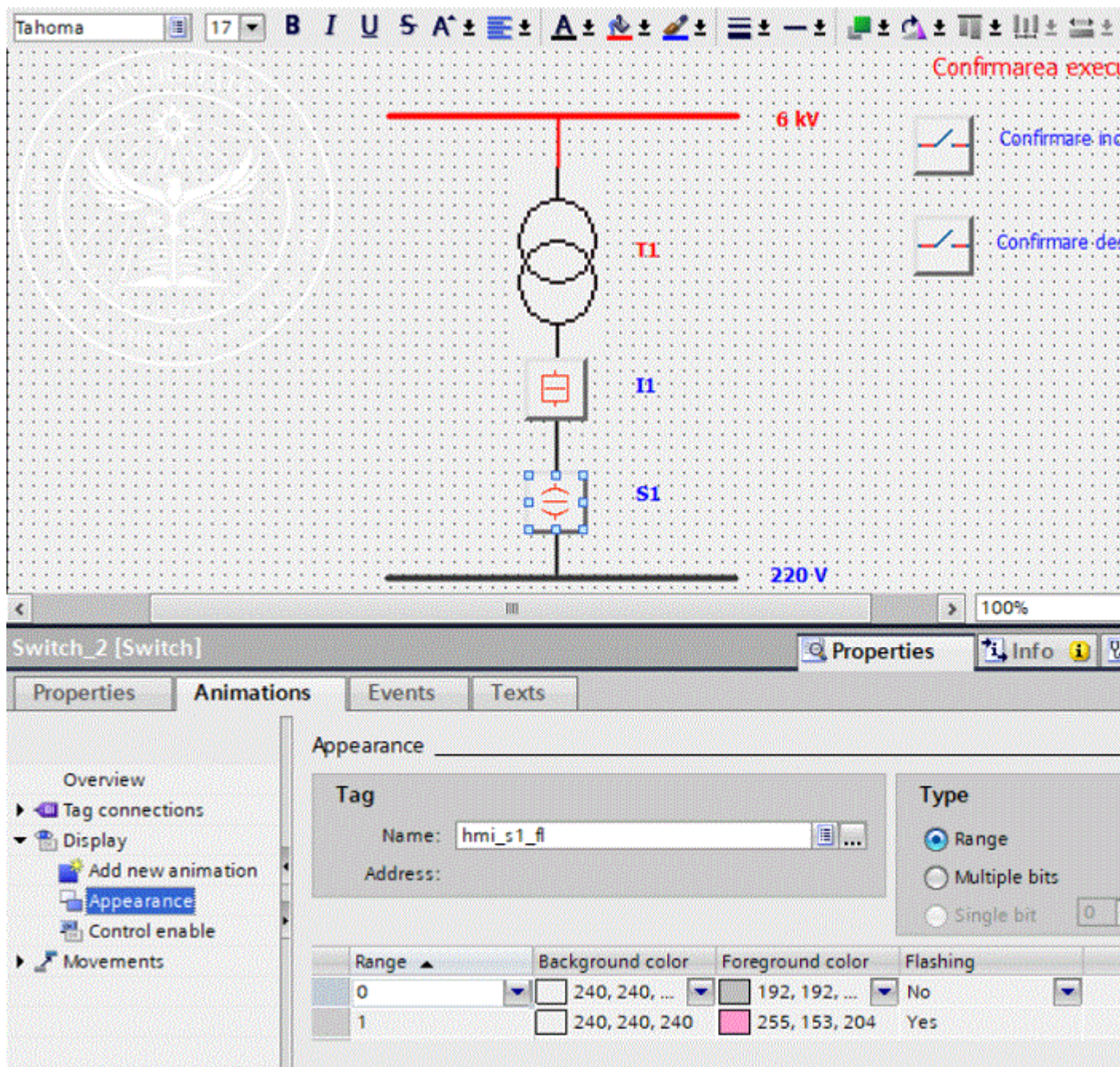
Tag-uri aferente

Nume	Tip	Domeniu	Um	Comentariu
hmi_s1_i	Bool	-	-	Confirmare separator S1 inchis
hmi_s1_d	Bool	-	-	Confirmare separator S1 deschis
hmi_s1_fl	Bool	-	-	Flesh S1

Confirmările de realizare ale comenzilor sunt simulate de butoanele din partea dreapta a screen-ului.



Dupa actionarea separatorului s1, se activeaza flesh pana la aparitia confirmarii. Pentru activarea flesh-ului s-a definit tag-ul hmi_s1_fl.



Activarea tag-ului hmi_s1_fl se face prin intermediul script-ului "Act_sch4()". Acest script se lanseaza pe evenimentele "Click" ale tuturor butoanelor cat si pe evenimentul "Load" al screen-ului "Sch_monof_4".

```
Sub Act_sch4()
    'Actualizare Sch_monof_4
    'Se tine seama de confirmarea inchiderii sau deschiderii S1

    'S1 inchis si nu are confirmare
    If hmi_s1 And (Not hmi_s1_i) Then
        hmi_s1_fl=True
    End If
End Sub
```



```

End If

'S1 deschis si nu are confirmare
If (Not hmi_s1) And (Not hmi_s1_d) Then
    hmi_s1_fl=True
End If
'S1 inchis si are confirmare sau S1 deschis si are confirmare
If (hmi_s1 And hmi_s1_i) Or ((Not hmi_s1) And hmi_s1_d) Then
    hmi_s1_fl=False
End If
'Confirmare simultana S1 deschis si inchis
If hmi_s1_i And hmi_s1_d Then
    hmi_s1_fl=True
End If
'I1 inchis, S1 inchis si are confirmare deci trafo T1 este activ
If hmi_s1_i And hmi_i1 Then
    hmi_t1=True
Else
    hmi_t1=False
End If
End Sub

```

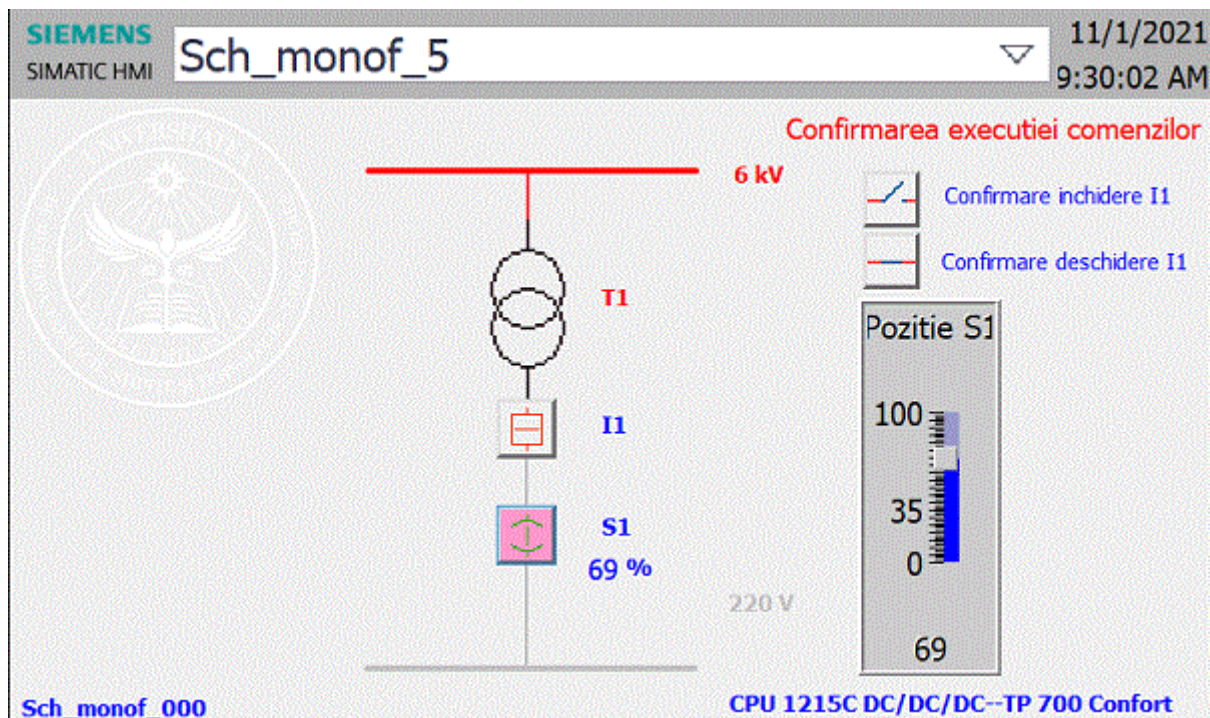
• Simularea executiei comenzilor

Inchiderea respectiv deschiderea separatorului se face in intervale de timp de ordinul secundelor, deci aceste operatii ar putea fi simulate printr-o marime analogica (avand valori între 0 si 100%) prin utilizarea unui obiect de tip "Slider" care ar fi in legatura cu un tag de tip analogic. Inchiderea respectiv deschiderea intreruptorului se face in intervale de timp de ordinul milisecundelor deci simularea acestei operatii se va face numai pe baza a doua semnale digitale care sa confirme inchiderea respectiv deschiderea intreruptorului. Avem nevoie deci de urmatoaree tag-uri:

Tag-uri aferente				
Nume	Tip	Domeniu	Um	Comentariu
hmi_s1	Bool	-	-	Separator S1
hmi_s1_i	Bool	-	-	Confirmare separator S1 inchis
hmi_s1_d	Bool	-	-	Confirmare separator S1 deschis
hmi_s1_fl	Bool	-	-	Flesh S1
poz_s1	INT	100	%	Pozitie separator S1
hmi_i1	Bool	-	-	Intreruptor I1
hmi_i1_i	Bool	-	-	Confirmare intreruptor I1 inchis
hmi_i1_d	Bool	-	-	Confirmare intreruptor I1 deschis
hmi_i1_fl	Bool	-	-	Flesh I1
hmi_t1	Bool	-	-	Transformator T1

Vom realiza deci urmatoarea pagina grafica **sch_monof_5** in care simularea inchiderii

respectiv deschiderii separatorului se va realiza manual de la un "Slider", iar inchiderea respectiv deschiderea intreruptorului va fi confirmata de la butoane.



Dupa actionarea intreruptorului I1, se activeaza flesh pana la aparitia confirmarii. Pentru activarea flesh-ului s-a definit tag-ul hmi_i1_fl.

Dupa actionarea separatorului s1, se activeaza flesh pana la inchiderea completa a separatorului, respectiv pan procentul de inchidere ajunge la 100%. Pentru activarea flesh-ului s-a definit tag-ul hmi_s1_fl.

Activarea tag-urilor hmi_i1_fl, hmi_s1_fl se face prin intermediul script-ului "Act_sch5()". Acest script se lanseaza pe evenimentele "Click" ale tuturor butoanelor, pe evenimentul "Change" al slider-ului cat si pe evenimentul "Load" al screen-ului "Sch_monof_5".

```
Sub Act_sch5()
    'Actualizare Sch_monof_5
    'Se tine seama de confirmarea inchiderii sau deschiderii I1 cat si
    de pozitia separatorului

    'Setarea confirmarilor pentru S1 in functie de poz S1
    If hmi_s1_poz =100 Then
        hmi_s1_i=True
        hmi_s1_d=False
    End If
    If hmi_s1_poz =0 Then
        hmi_s1_i=False
        hmi_s1_d=True
    End If

    If (hmi_s1_poz > 0) And (hmi_s1_poz < 100) Then
```

```

        hmi_s1_i=False
        hmi_s1_d=False
    End If

    'Validare S1

    'S1 inchis si nu are confirmare
    If hmi_s1 And (Not hmi_s1_i) Then
        hmi_s1_fl=True
    End If

    'S1 deschis si nu are confirmare
    If (Not hmi_s1) And (Not hmi_s1_d) Then
        hmi_s1_fl=True
    End If

    ' S1 inchis si are confirmare sau S1 deschis si are confirmare
    If (hmi_s1 And hmi_s1_i) Or ((Not hmi_s1) And hmi_s1_d) Then
        hmi_s1_fl=False
    End If

    'Confirmare simultana S1 deschis si inchis
    If hmi_s1_i And hmi_s1_d Then
        hmi_s1_fl=True
    End If

    ' I1 inchis, S1 inchis si are confirmare deci trafo T1 este activ
    If hmi_s1_i And hmi_i1 Then
        hmi_t1=True
    Else
        hmi_t1=False
    End If

    'Validare I1

    'I1 inchis si nu are confirmare
    If hmi_i1 And (Not hmi_i1_i) Then
        hmi_i1_fl=True
    End If

    'I1 deschis si nu are confirmare
    If (Not hmi_i1) And (Not hmi_i1_d) Then
        hmi_i1_fl=True
    End If

    ' I1 inchis si are confirmare sau I1 deschis si are confirmare
    If (hmi_i1 And hmi_i1_i) Or ((Not hmi_i1) And hmi_i1_d) Then
        hmi_i1_fl=False
    End If

    'Confirmare simultana I1 deschis si inchis
    If hmi_i1_i And hmi_i1_d Then
        hmi_i1_fl=True
    End If

    'Validare Trafo

    ' I1 inchis si are confirmare, S1 inchis si are confirmare deci
    trafo T1 este activ
    If hmi_s1_i And hmi_i1_i Then
        hmi_t1=True
    Else
        hmi_t1=False
    End If

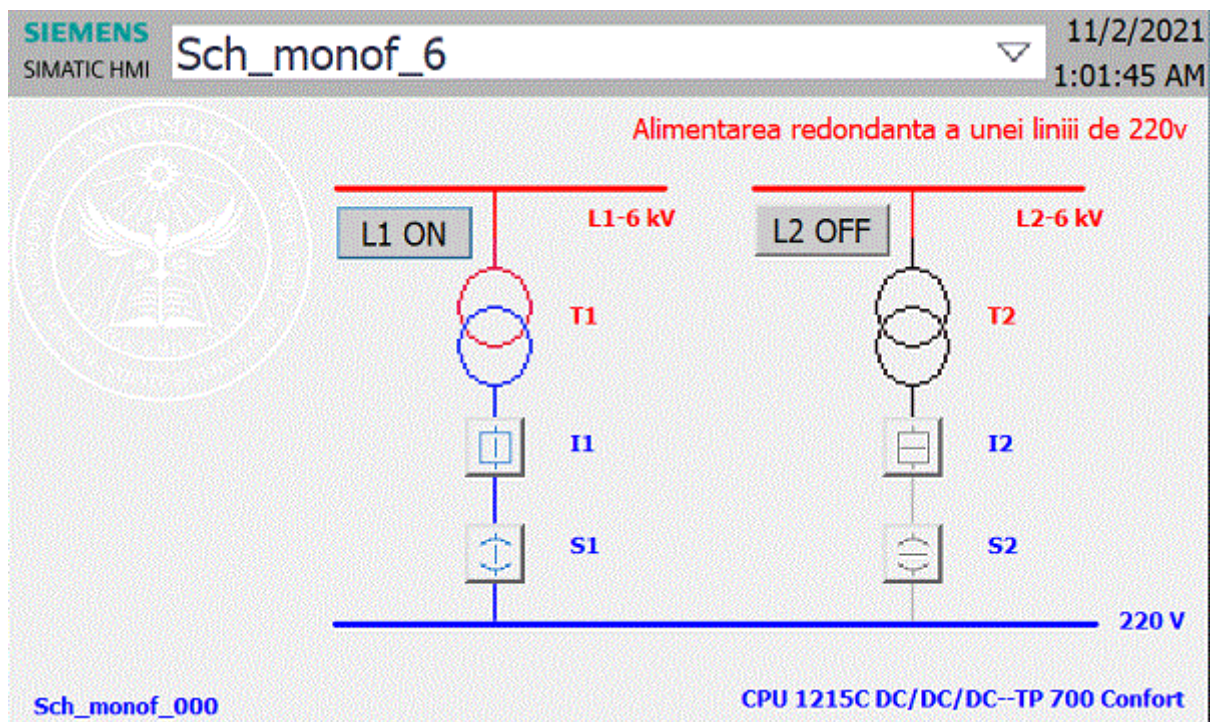
```


End Sub

- **Aplicatia de proces: executia comenzilor**

- **Schema redondanta de alimentare cu energie a unei linii electrice**

Urmatoarea pagina grafica **sch_monof_6** o reprezinta o schema monofilara pentru alimentarea redondanta a unei linii de 220v de la doua linii diferite de 6kv. Aplicatia SCADA trebuie sa permita conectarea uneia dintre linii si sa nu permita conectarea simultana a celor doua linii. Manevrele de comutare se pot face manual sau automat de la cele doua butoane.



Pentru realizarea aplicatiei avem nevoie de urmatoarele HMI tags

HMI tags				
Nume	Tip	Domeniu	Um	Comentariu
hmi_i1	Bool	-	-	Intrerupator I1
hmi_i2	Bool	-	-	Intrerupator I2
hmi_s1	Bool	-	-	Separator S1
hmi_s2	Bool	-	-	Separator S2
hmi_t1	Bool	-	-	Transformator T1
hmi_t2	Bool	-	-	Transformator T2

hmi_l1	Bool	-	-	Linia L1
hmi_l2	Bool	-	-	Linia L2
hmi_l220	Bool	-	-	Linia L220

Acest script se lanseaza pe evenimentele "Click" ale butoanelor S1, I1, S2, I2 cat si pe evenimentul "Load" al screen-ului "Sch_monof_6".

```
Sub Act_sch_6()
    If hmi_i1 And hmi_s1 Then
        hmi_t1=True
        hmi_l1=True
        hmi_l2=False
        hmi_i2=False
        hmi_l220=True
    Else
        hmi_t1=False
        hmi_l1=False
        hmi_l220=False
    End If
    If hmi_i2 And hmi_s2 Then
        hmi_t2=True
        hmi_l2=True
        hmi_l1=False
        hmi_i1=False
        hmi_l220=True
    Else
        hmi_t2=False
        hmi_l2=False
        hmi_l220=False
    End If
    If Not (hmi_t1 Or hmi_t2) Then
        hmi_l220=False
    End If
End Sub
```

Pe evenimentul "Change al butonului L1 se lanseaza script-ul:

```
Sub Act_sch_6_sw1()
    If hmi_l1 Then
        hmi_l2=False
        hmi_i2=False
        hmi_s2=False
        hmi_t2=False
        hmi_s1=True
        hmi_i1=True
        hmi_t1=True
        hmi_l220=True
    Else
        hmi_s1=False
        hmi_i1=False
        hmi_t1=False
    End If
End Sub
```



```
        hmi_1220=False
    End If
End Sub
```

Pe evenimentul "Change al butonului L2 se lanseaza script-ul:

```
Sub Act_sch_6_sw2()
    If hmi_12 Then
        hmi_11=False
        hmi_i1=False
        hmi_s1=False
        hmi_t1=False
        hmi_s2=True
        hmi_i2=True
        hmi_t2=True
        hmi_1220=True
    Else
        hmi_s2=False
        hmi_i2=False
        hmi_t2=False
        hmi_1220=False
    End If
End Sub
```

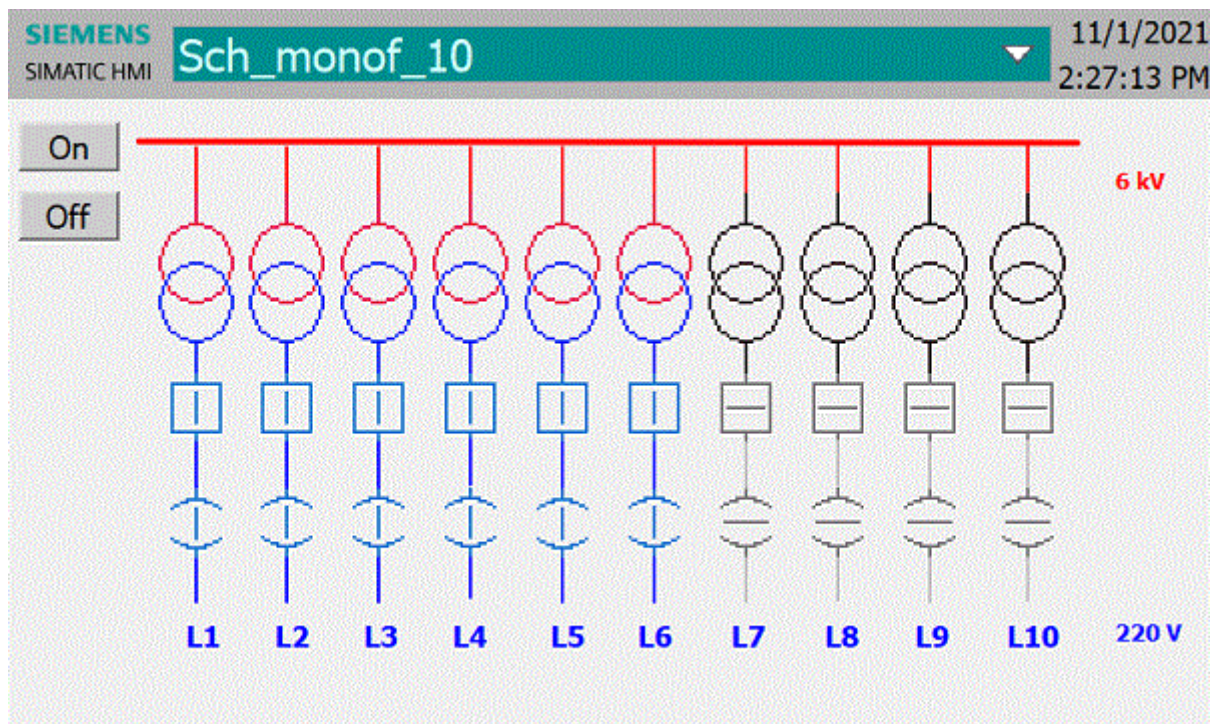
3. Managementul liniilor monofilare

In multe cazuri in cadrul schemelor monofilare se intalnesc mai multe elemente de acelasi tip. In acest caz se recomanda definirea tag-urilor de tip Array si utilizarea instructiunilor repetitive.

Vom realiza in continuare diverse scheme monofilare in care avem mai multe linii de alimentare dotate fiecare cu transformator, intreruptor, separator etc.

Pentru inceput, sa luam o schema monofilara cu 10 linii separate de alimentare.

Sa realizam un nou screen **sch_monof_10** care sa comande inchiderea respectiv deschiderea elementelor de comutatie, intr-o anumita ordine.



Pentru inchiderea respectiv deschiderea elementelor de comutatie, vom introduce 3 tag-uri de tip Array:

Tag-uri aferente					
Nume	Tip	Domeniu	Um	Array Size	Comentariu
hmi_sep	Bool	-	-	11	Separatoarele sep[1]-sep[10]
hmi_intr	Bool	-	-	11	Intreruptoarele intr[1]-intr[10]
hmi_trafo	Bool	-	-	11	Transformatoarele trafo[1]-trafo[10]

Pe evenimentelor click ale butoanelor "On" respectiv "Off" le atribuim scriept-urile "Act_sch_10_on()" respectiv Act_sch_10_off().

```

Sub Act_sch_10_on()
    Dim i,tm
    tm=Timer()
    i=1
    Do While i <= 10
        hmi_sep(i)=True
        End If
        If Timer > tm+0.6 Then
            hmi_intr(i)=True
            hmi_trafo(i)=True
            tm=Timer()
            i=i+1
        End If
    Loop

```



```

End Sub

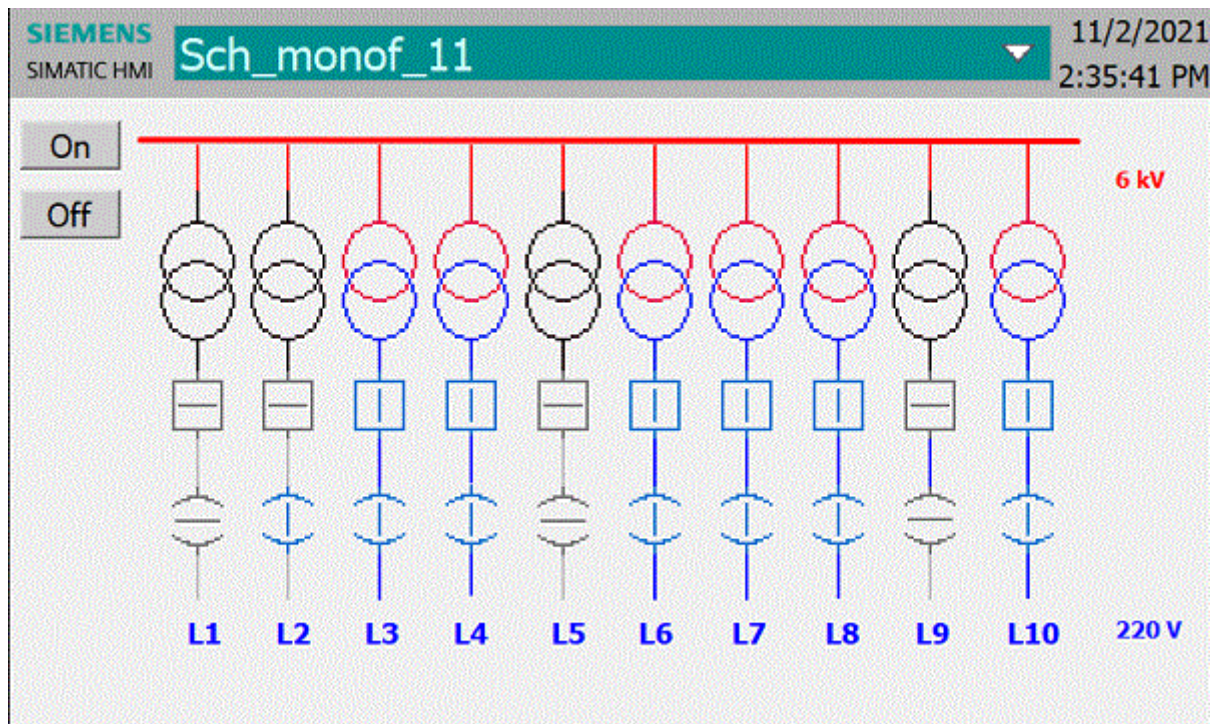
Sub Act_sch_10_off()
    Dim i, tm
    tm=Timer()
    i=10
    Do While i>0
        If Timer > tm+0.3 Then
            hmi_intr(i)=False
            hmi_trafo(i)=False
        End If
        If Timer > tm+0.6 Then
            hmi_sep(i)=False
            tm=Timer()
            i=i-1
        End If
    Loop
End Sub

```

- **Ordinea de comutare sau deconectare a liniilor**

Sa presupunem acum ca dorim activarea elementelor de comutatie dintr-o schema monofilara, intr-o anumita ordine.

Urmatoarea pagina grafica **sch_monof_11** o reprezinta o schema monofilara in care elementele de comutatie comuta intr-o anumita ordine stabilita intr-un vector de prioritati.



Scriptul "Act_sch_11_on()" se activeaza pe evenimentul "Click" al butonului "on"

```

Sub Act_sch_11_on()
    Dim i,tm, prior(10)
    prior(1)=10
    prior(2)=6
    prior(3)=7
    prior(4)=4
    prior(5)=3
    prior(6)=8
    prior(7)=2
    prior(8)=5
    prior(9)=9
    prior(10)=1
    tm=Timer()
    i=1
    Do While i <= tm+0.3 Then
        hmi_sep(prior(i))=True
    End If
    If Timer > tm+0.6 Then
        hmi_intr(prior(i))=True
        hmi_trafo(prior(i))=True
        tm=Timer()
        i=i+1
    End If
    Loop
End Sub

```

Scriptul "Act_sch_11_off()" se activeaza pe evenimentul "Click" al butonului "off"

```

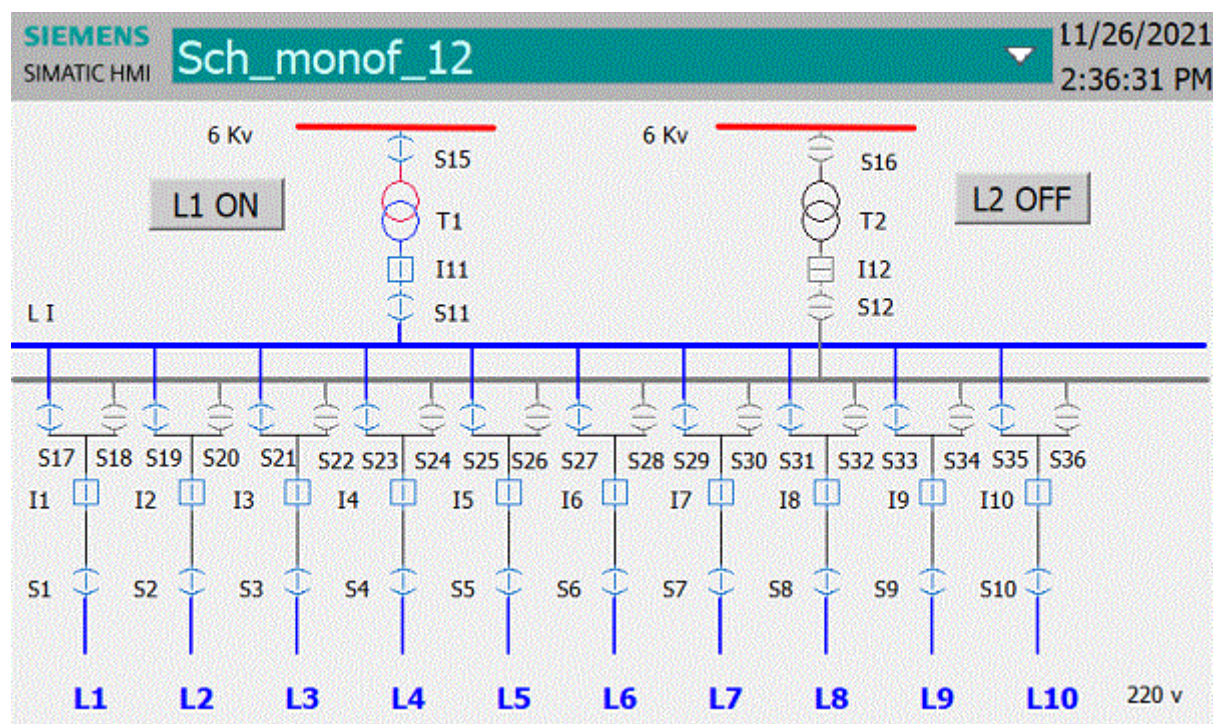
Sub Act_sch_11_off()
    Dim i,tm, prior(10)
    prior(1)=10
    prior(2)=6
    prior(3)=8
    prior(4)=4
    prior(5)=3
    prior(6)=7
    prior(7)=2
    prior(8)=5
    prior(9)=9
    prior(10)=1
    tm=Timer()
    i=10
    Do While i>0
        If Timer > tm+0.3 Then
            hmi_intr(prior(i))=False
            hmi_trafo(prior(i))=False
        End If
        If Timer > tm+0.6 Then
            hmi_sep(prior(i))=False
            tm=Timer()
            i=i-1
        End If
    Loop
End Sub

```


- **Schema redondanta de alimentare cu energie a mai multor linii**

Pentru asigurarea continuitatii alimentarii cu energie electrica se realizeaza diverse scheme redondante de alimentare cu energie pe mai multe linii.

Urmatoarea pagina grafica **sch_monof_12** o reprezinta o schema monofilara redondanta pentru alimentarea cu energie pe mai multe linii



Scriptul "Act_sch_12_sw1()" si "Act_sch_12_l1()" se activeaza pe evenimentul "Click" al butonului "L1"

```
Sub Act_sch_12_sw1()
    If hmi_l1 Then
        hmi_l2=False
    End If
End Sub

Sub Act_sch_12_l1()
    Dim j
    If hmi_l1 Then
        hmi_sep(12)=False
        hmi_intr(12)=False
        hmi_sep(16)=False
        hmi_trafo(2)=False

        hmi_sep(11)=True
        hmi_intr(11)=True
    End If
End Sub
```

```

        hmi_sep(15)=True
        hmi_trafo(1)=True
        For j=1 To 10
            hmi_sep(j)=True
            hmi_intr(j)=True
            hmi_l(j)=True
        Next
        For j=18 To 36
            hmi_sep(j)=False
            j=j+1
        Next
        For j=17 To 35
            hmi_sep(j)=True
            j=j+1
        Next

    Else
        hmi_sep(11)=False
        hmi_intr(11)=False
        hmi_sep(15)=False
        hmi_trafo(1)=False
        For j=1 To 10
            hmi_sep(j)=False
            hmi_intr(j)=False
            hmi_l(j)=False
        Next
        For j=17 To 35
            hmi_sep(j)=False
            j=j+1
        Next
    End If
End Sub

```

Scriptul "Act_sch_12_sw2()" si "Act_sch_12_l2()" se activeaza pe evenimentul "Click" al butonului "L2"

```

Sub Act_sch_12_sw2()
    If hmi_l2 Then
        hmi_l1=False
    End If
End Sub

Sub Act_sch_12_l2()
    Dim j
    If hmi_l2 Then
        hmi_sep(11)=False
        hmi_intr(11)=False
        hmi_sep(15)=False
        hmi_trafo(1)=False

        hmi_sep(12)=True
        hmi_intr(12)=True
        hmi_sep(16)=True
        hmi_trafo(2)=True
        For j=1 To 10

```



```

        hmi_sep(j)=True
        hmi_intr(j)=True
        hmi_l(j)=True
    Next
    For j=17 To 35
        hmi_sep(j)=False
        j=j+1
    Next
    For j=18 To 36
        hmi_sep(j)=True
        j=j+1
    Next
Else
    hmi_sep(12)=False
    hmi_intr(12)=False
    hmi_sep(16)=False
    hmi_trafo(2)=False
    For j=1 To 10
        hmi_sep(j)=False
        hmi_intr(j)=False
        hmi_l(j)=False
    Next
    For j=18 To 36
        hmi_sep(j)=False
        j=j+1
    Next
End If
End Sub

```

Rezumat

Aplicatii in energetica - linii monofilare

Pentru a realiza scheme monofilare avem nevoie de o serie de simboluri cum ar fi simboluri pentru: separatoare, intreruptoare, transformatoare etc.

O schema monofilara contine o serie de elemente de comutatie cum ar fi separatoare, intreruptoare, etc. Exista o ordine in care se actioneaza cele doua elemente, in sensul ca separatorul nu poate fi niciodata actionat in sarcina.

In sistemele energetice, inchiderea respectiv deschiderea unui separator sau a unui intreruptor nu este instantanee. In aplicatiile SCADA se tine cont de acest fenomen. Dupa comanda actionarii unui element se scurge un interval de timp pana la efectuarea completa a comenzii. Dupa actionarea unui element e nevoie de un semnal care sa confirme efectuarea comenzii.

Pentru testarea aplicatiilor SCADA fara conectare la sistemul real de achizitie si comanda, inchiderea respectiv deschiderea elementelor de comutatie, ar putea fi simulata printr-o marime analogica, avand valori intre 0 si 100%. Pe schema aceasta valoare ar putea fi simulata prin utilizarea unui control de tip "Slider" care ar fi in legatura cu un tag de tip analogic.

Starea fiecarui element plasat pe HMI este controlata de tag-ul corespunzator.

In multe cazuri in cadrul schemelor monofilare se intalnesc mai multe elemente de acelasi

tip. In acest caz se recomanda definirea tag-urilor de tip Array si utilizarea instructiunilor repetitive.

In cazul sistemelor de alimentare cu energie, avand mai multe linii, pentru a evita socurile de sarcina, conectarea respectiv deconectarea acestora se face secvential intr-o ordine prestabilita, in functie de importanta utilizatorului. Pentru a memora prioritatea consumatorilor, se recomanda utilizarea unui vector de prioritati.

Tag-urile sunt actualizate de sistemul de achizitie si comanda. Acesta este prevazut atat cu elemente pentru citirea strilor cat si cu elemente de comanda.

• Rezultate asteptate

Dupa studierea acestui modul, ar trebui sa stiti:

- Sa realizati pagini grafice reprezentand scheme monofilare
- Sa dati functionalitate schemelor monofilare
- Sa implementati algoritmul de functionare al elementelor de comutatie incluse in aplicatii SCADA
- Sa realizati aplicatii SCADA care sa cuprinda scheme monofilare complexe.

• Termeni esentiali

Termen	Descriere
SCADA	Supervisory Control And Data Acquisition
Tag	Nume generic pentru elementele din procesul monitorizat codificate prin intermediul variabilelor
HMI	Human Machine Interface -Interfata dintre aplicatie si utilizator
VB Script	Limbaj de programare inclus in mediul de dezvoltare HMI SCADA - TIA portal
Sistem de achiziti date	Sistem fizic care interfațeaza sistemele energetice cu aplicatiile SCADA. Acesta este prevazut atat cu elemente pentru citirea strilor elementelor componente ale sistemului energetic cat si cu elemente de comanda
Elemente de comutatie	Elemente componente ale retelelor de alimentare cu energie care permit inchiderea respectiv deschiderea unui circuit

• 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: Elemente de programare C++ utile in ingineria electrica, ISBN: 978-973-755-576-2, Ed. MatrixRom, 2009
- [6] T. Turc: Programare avansata C++ pentru ingineria electrica, ISBN: 978-973-755-588-5, Ed. MatrixRom, 2009.

- [7] Boldur Barbat - Informatica industrială - Programarea în timp real – Institutul Central pentru Conducere și informatica 1984
- [8] Ioan Babuita – Conducerea automată a proceselor – Ed. Facla 1985
- [9] Ghercioiu-National Instruments - Orizonturi în instrumentație 1995

• 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 -

Test de evaluare

- -Marcați răspunsurile corecte la întrebările următoare.
- -ATENȚIE: pot exista unul, niciunul sau mai multe răspunsuri corecte la aceeași întrebare.
- -Timp de lucru: 10 minute

1. Cine controlează starea elementelor plasate pe o schemă monofilă ?

- ☐ a. Simbolurile plasate
- ☐ b. Tag-urile asociate
- ☐ c. Variabilele locale utilizate
- ☐ d. Funcțiile definite de utilizator

Răspuns:

2. Pentru a introduce funcționalități de comandă pentru elementele de pe schemă, se setează proprietatea:

- ☐ a. Access
- ☐ b. Input
- ☐ c. Slider

- ☐ d. Movement

Raspuns:

3. Care este conditionalitatea intre actionarea separatorului si intreruptorului ?

- ☐ a. Primul se actioneaza intotdeauna intreruptorul
☐ b. Primul se actioneaza intotdeauna separatorul
☐ c. Separatorul nu se actioneaza in sarcina
☐ d. Intreruptorul nu se actioneaza in sarcina

Raspuns:

4. Ce reprezinta confirmarea executiei comenzilor ?

- ☐ a. Aparitia unui eveniment
☐ b. Aparitia unui semnal de confirmare
☐ c. Modificarea valorii unui tag
☐ d. Scurgerea unui interval de timp

Raspuns:

5. De ce e nevoie de conectare secventiala a liniilor ?

- ☐ a. Pentru a evita socurile de sarcina
☐ b. Pentru a diferentia consumatorii in functie de importanta
☐ c. Pentru implementarea nivelelor de prioritate
☐ d. Pentru a putea urmari conectarea fiecarei linii in parte

Raspuns:

6. Un tag care gestioneaza un simbol reprezentand un separator are tipul:

- ☐ a. INT
☐ b. DIGITAL
☐ c. BYTE
☐ d. BCD

Raspuns:

7. Prescrierea curentului nominal pentru o linie se face utilizand:

- ☐ a. Variabile locale
- ☐ b. Vectori de valori
- ☐ c. Tag-uri
- ☐ d. Functii definite de utilizator

Raspuns:

8. Dupa decuplarea unei linii care a depasit curentul nominal, recuplarea acesteia se face:

- ☐ a. Automat dupa ce valoarea curentului scade dupa curentul nominal
- ☐ b. Dupa actionarea manuala
- ☐ c. Dupa un algoritm prestabilit in program
- ☐ d. Dupa un interval de timp prestabilit

Raspuns:

9. Calculul consumului total se face:

- ☐ a. Utilzand un vector
- ☐ b. Utilizand un tag
- ☐ c. Utilizand un algoritm de calcul
- ☐ d. Utilizand un obiect de tip Trending

Raspuns:

10. Un vector care pastreaza curentii nominali se defineste:

- ☐ a. Din program
- ☐ b. Prin introducerea unui tag de tip Array
- ☐ c. Prin utilizarea unui obiect de tip multitrending
- ☐ d. Prin utilizarea mai multor variabile

Raspuns: