

Dezvoltarea HMI-urilor in SCADA - TIA Portal

- **Obiective**

- Prezentarea limbajului de programare, mod de utilizare, particularitati
- Prezentarea modului de realizarea a functiilor
- Prezentarea instructiunilor decizionale, repetitive si realizarea de aplicatii cu aceste instructiuni
- Prezentarea si exemplificarea facilitatilor de programarea in realizarea aplicatiilor SCADA

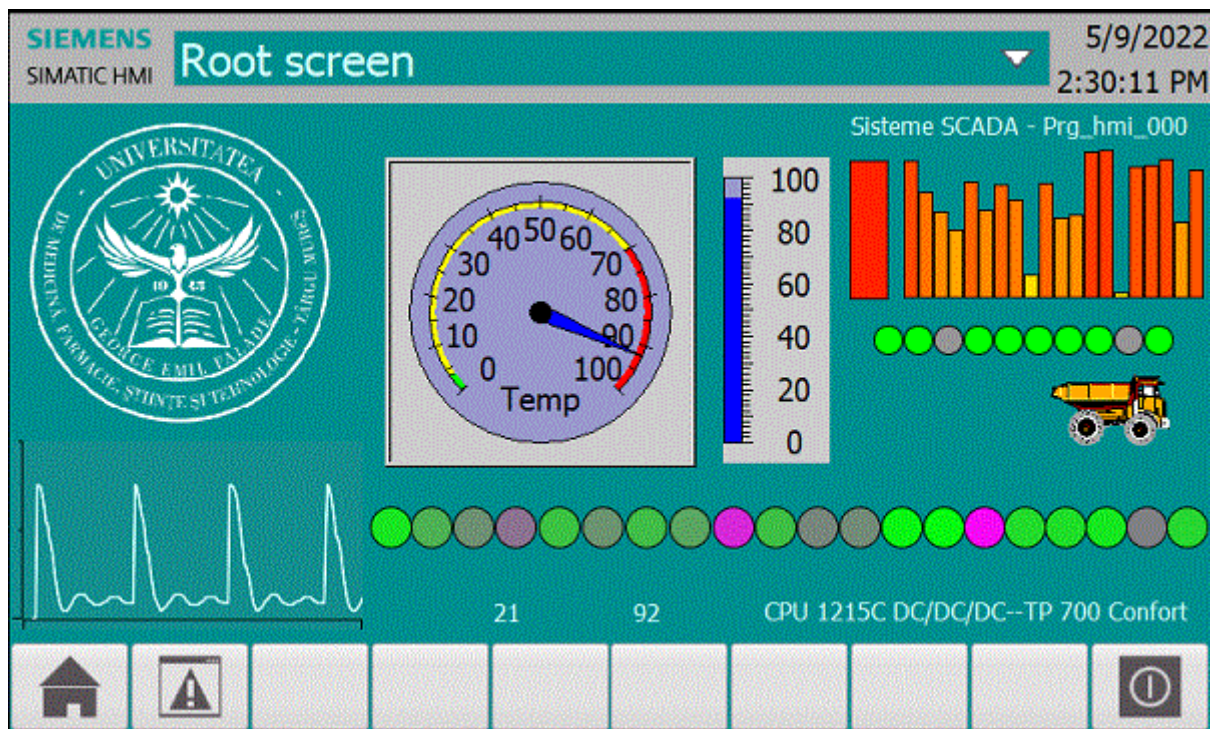
- **Organizarea sarcinilor de lucru**

- 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.

Aplicatiile SCADA simple se pot crea in principiu fara a utiliza notiuni de programare. Aplicatiile mai complexe nu pot fi realizate decat utilizand facilitatile oferite de un limbaj de programare.

Mediul de dezvoltare SCADA TIA-Portal, include limbajul de programare Visual Basic, permitand astfel controlul software a elementelor plasate pe HMI-uri (Human Machine Interface).

Vom dezvolta in continuare proiectul numit [Prg hmi_000](#) in cadrul caruia vom realiza diverse screen-uri in care vom folosi facilitatile de programare Visual Basic.



1.Elemente de programare VBasic

Definirea variabilelor

In VB Script pot fi definite si utilizate variabile. Definirea unui variabile se face prin simpla decalarare a acesteia si anume:

Dim nume_variabila

Exemplu:

Dim contor

In VB Script variabilele sunt slab tipizate deci nu trebuie definit tipul.

O alta metoda pentru definirea unuei variabile se realizeaza prin definirea unui "Tag"

Definirea functiilor

Pot fi definite diverse functii Visual Basic numite "VB script", avand urmatorul format:

Sub nume_script()

declaratii

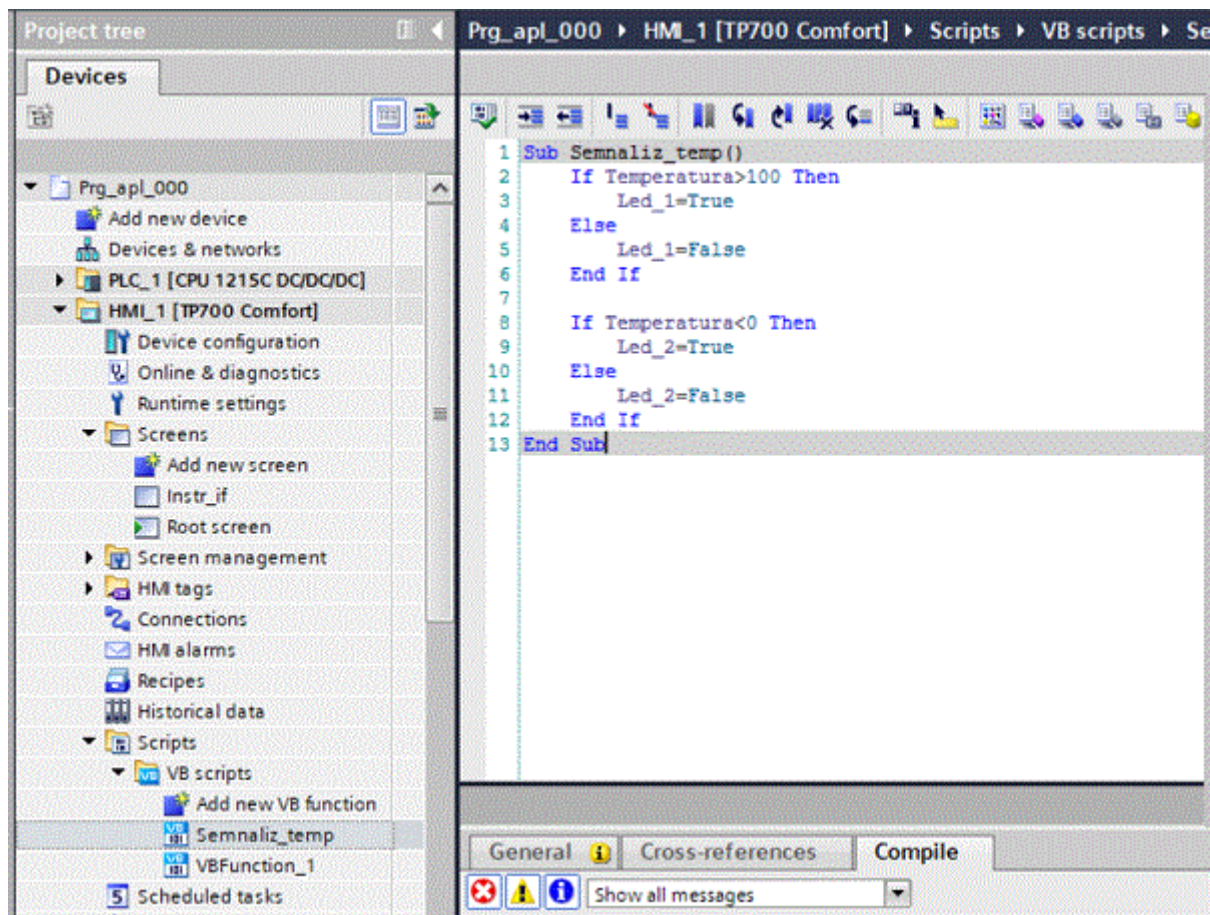
.

.

.

declaratii

End Sub



Obiectele plasate pe HMI-uri dispun de diverse evenimente, care permit lansarea script-urilor "VB script".

Definirea tablourilor

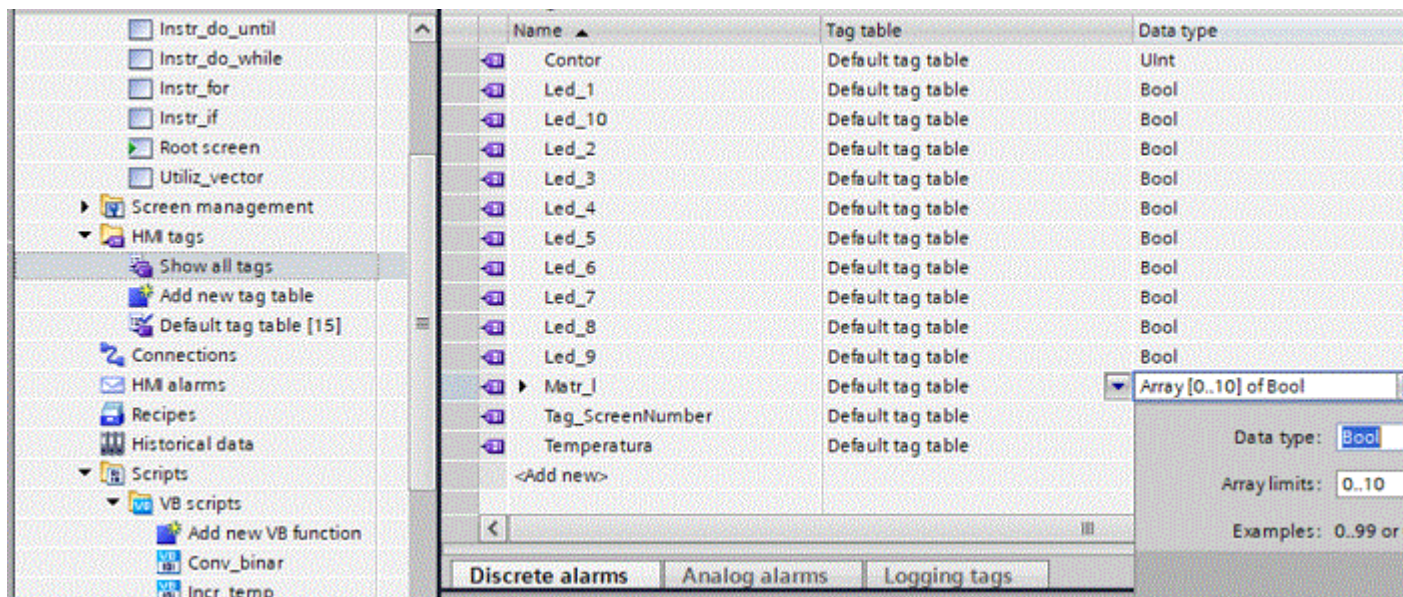
În VB Script pot fi definite și utilizate tablouri. Definirea unui tablou se face prin simpla decalarare a acestuia și anume:

Dim nume_tablou(nr_elemente)

Exemplu:

Dim valori(100)

O altă metodă pentru definirea unui tablou se realizează prin definirea unui "Tag" de tip "Array"



S-a definit astfel Tabloul "Matr_l" cu 11 elemente de tip Bool-ean

2. Instructiuni

Instructiunea if :

Instructiunea **if** se foloseste pentru a selecta executia unei instructiuni (sau a unui grup de instructiuni) functie de valoarea logica a unei expresii relationale

Formatul instructiunii:

Instructiunea **if** are urmatoarele formate:

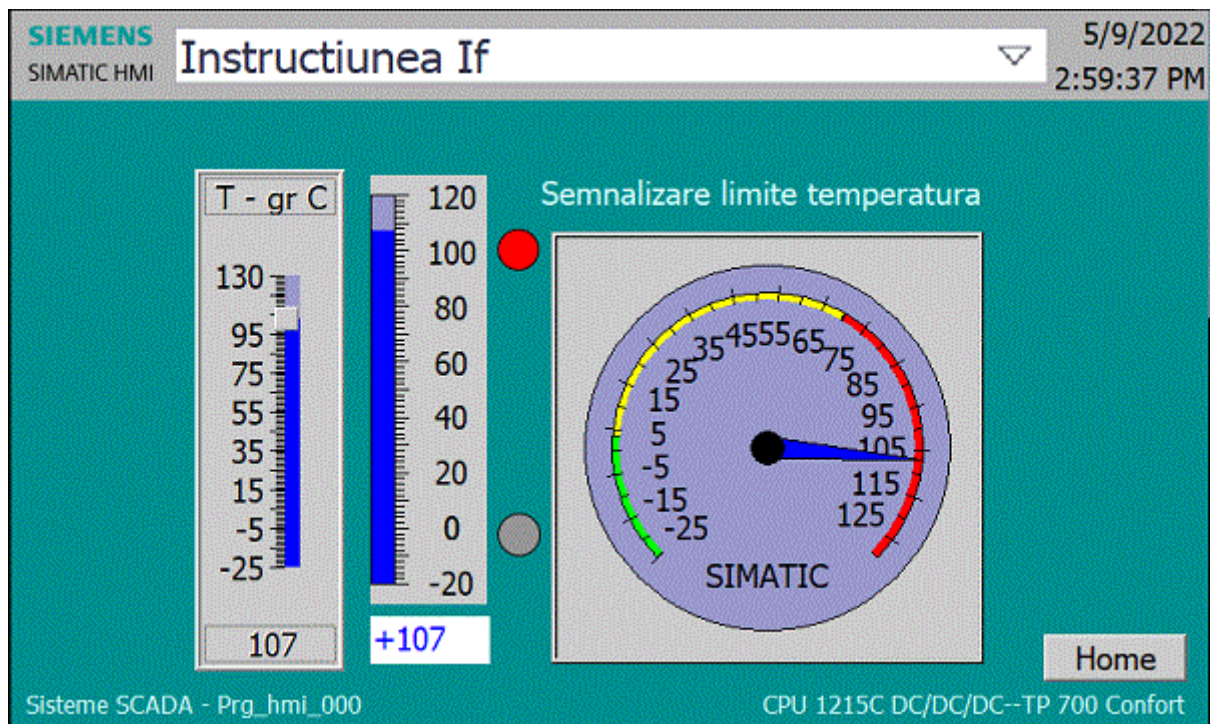
If expresie relationala **Then**
 instructiune(instructiuni)
End

sau

If expresie relationala **Then**
 instructiune(instructiuni)
Else
 instructiune(instructiuni)
End

Aplicatii care utilizeaza instructiuni decizionale

Vom crea in cadrul proiectului "Prg_hmi_000" un nou screen numit "Instr_if" in care vom semnaliza atingerea unui prag inferior respectiv a unui prag superior al unui parametru folosind doua LED-uri.



Avem nevoie de tag-urile:

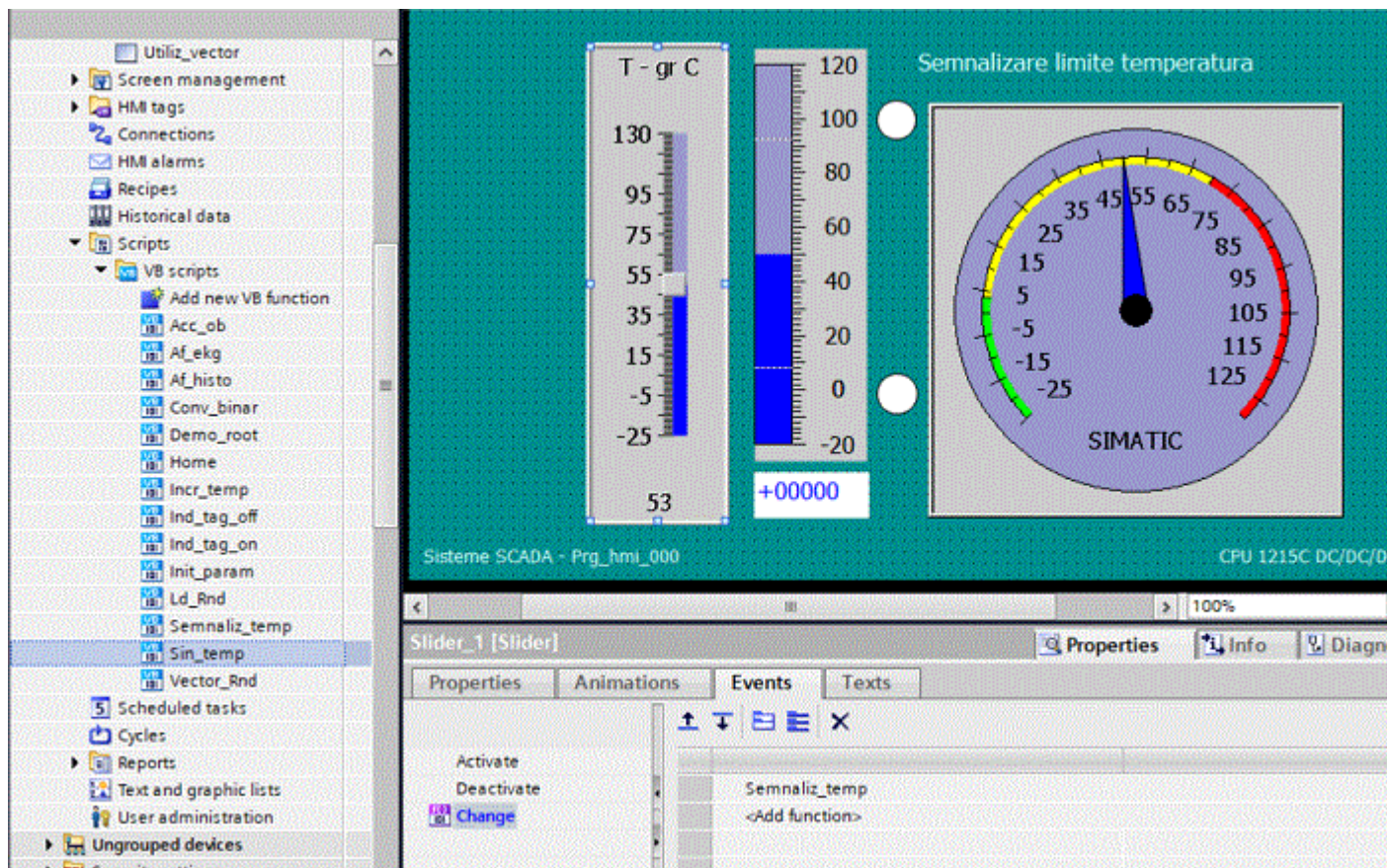
- "Temperatura" - pentru simularea temperaturii
- "Led_1"- pentru a semnaliza depasirea temperaturii peste 100 de grade
- "Led_2"- pentru a semnaliza scaderea temperaturii sub 0 de grade

Vom realiza script-ul "Semnaliz_temp()":

```
Sub Semnaliz temp()
    If Temperatura>100 Then
        Led_1=True
    Else
        Led_1=False
    End If

    If Temperatura
```

Vom atribui acest script evenimentului "Change" al obiectului "Slider_1" care seteaza valoarea temperaturii.



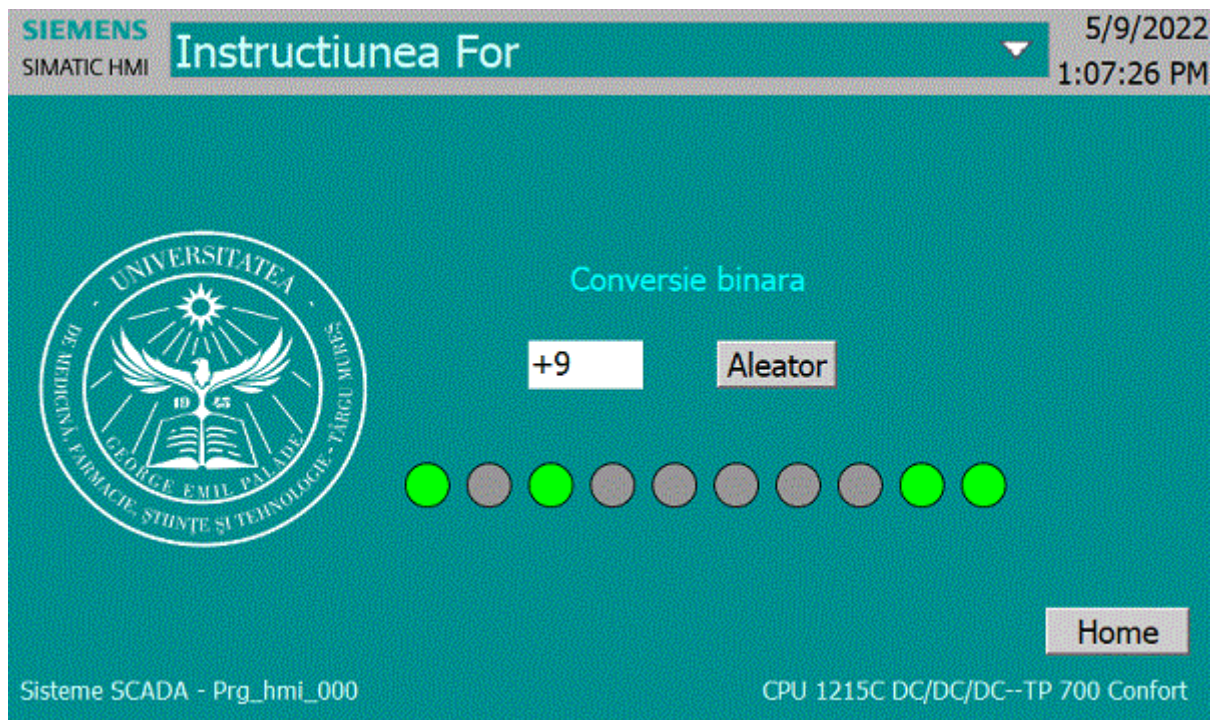
Instructiunea for :

```

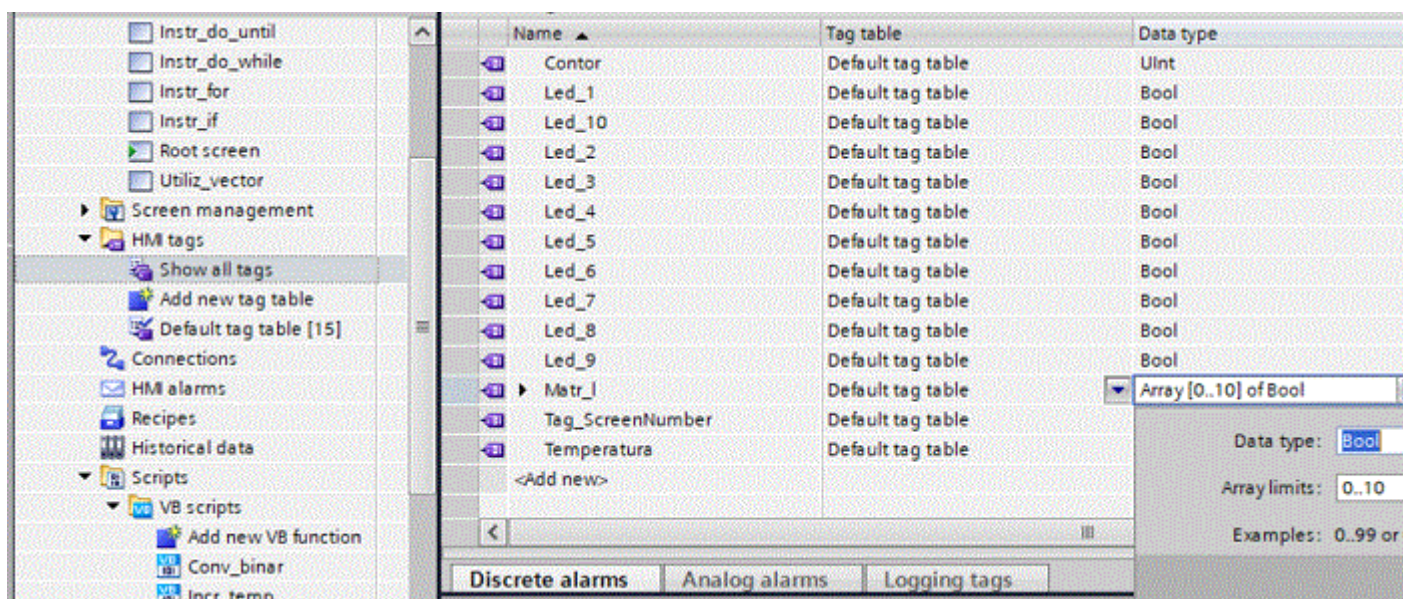
For contor=contor_start To contor_stop
    declaratii
    .
    .
    .
    declaratii
Next

```

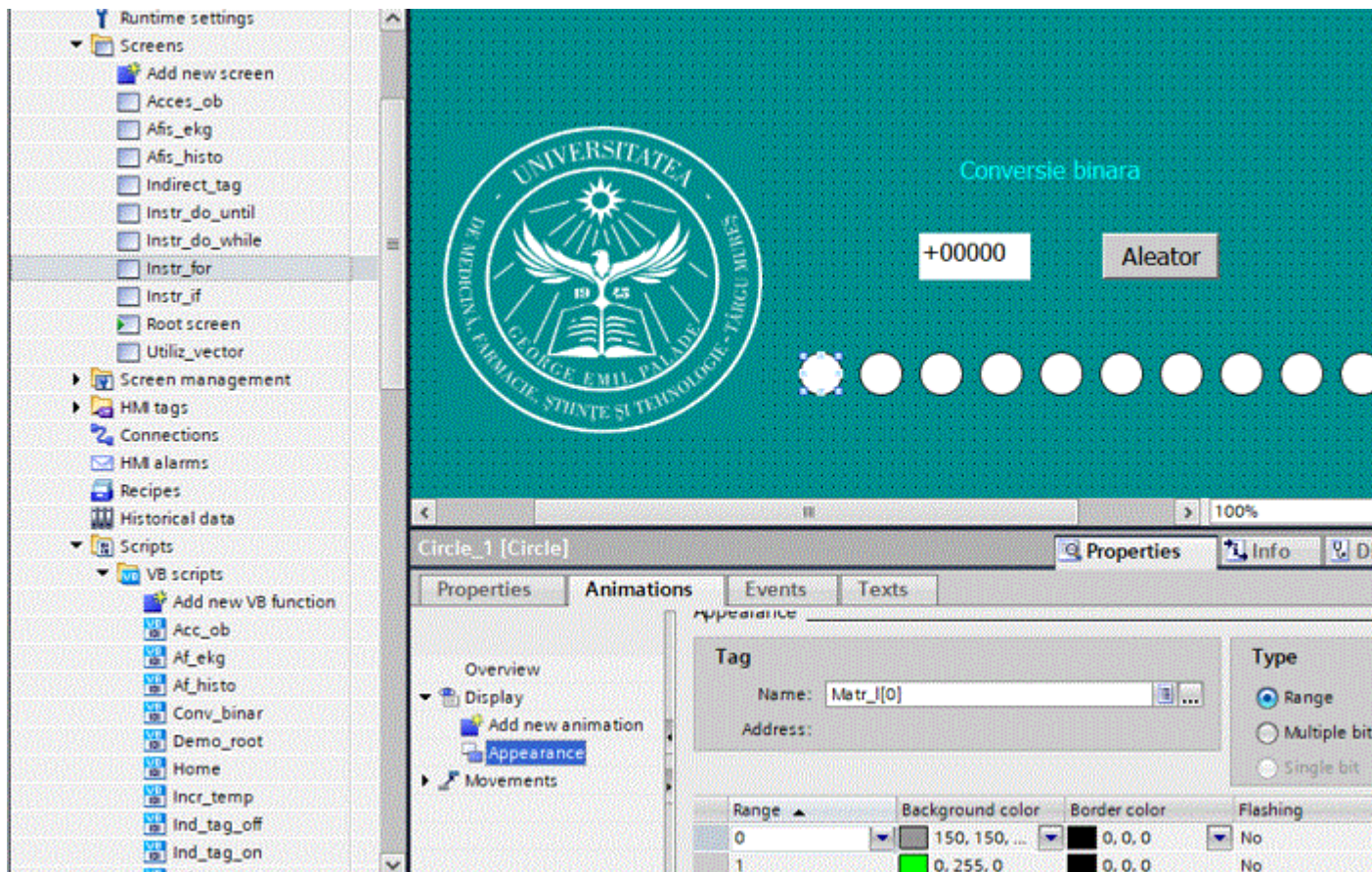
Vom adauga screen-ul "Instr_for" in care vom folosi instructiunea For pentru a converti un numar intreg in binar pe 10 biti si vom afisa rezultatul pe 10 LED-uri.



Informatiile despre cele 10 LED-uri vor fi pastrate in Tag-ul "Matr_1".



Vom seta proprietatea Animations --> Apperance pentru fiecare LED astfel:



Contorul de tip I/O field are setat tag-ul "Contor", iar pe evenimentul "Input Finished" s-a atribuit functia "Conv_binar"

```
Sub Conv_binar()
    Dim j,nr
    nr=Contor
    For j=0 To 9
        If (nr Mod 2)= 0 Then
            Matr_1(9-j)= False
        Else
            Matr_1(9-j)= True
        End If
        nr=Int(nr/2)
    Next
End Sub
```

Pe evenimentul "Click" al obiectului de tip Button s-a atribuit functia "Ld_Rnd"
Daca se apasa pe butonul "Aleator" se vor aprinde aleatorcele 10 LED-uri.

```
Sub Ld_Rnd()
    Dim i,r
    For i=0 To 10
```

```

        r=Rnd(2)
        If r > 0.5 Then
            Matr_1(i)= True
        Else
            Matr_1(i)= False
        End If
    Next
End Sub

```

Instructiunea do while :

Do While expresie relationala
declaratii

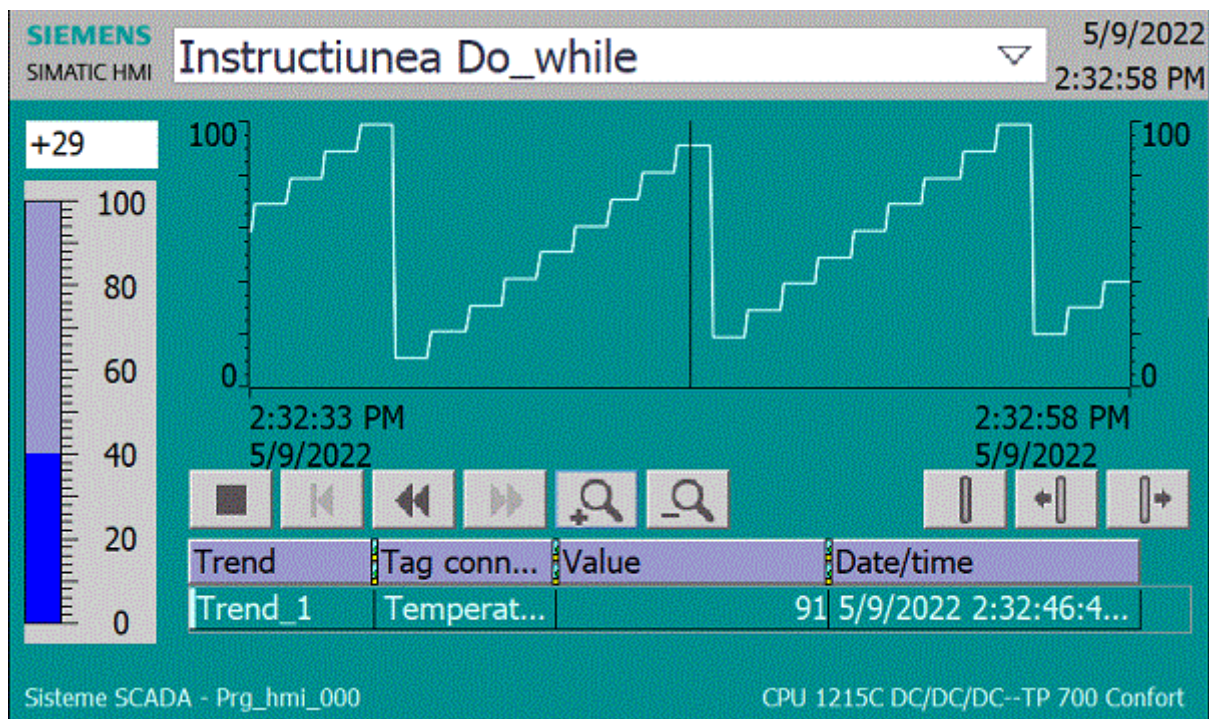
.

declaratii

Loop

Vom adauga screen-ul "Instr_do_while" in care vom folosi instructiunea Do While pentru a incrementa de zece ori temperatura cu cate 10 grade la interval de o secunda. Daca se trece de temperatura maxima adica 100 de grade, temperatura va fi initializata cu o valoare Random de maxim 50 de grade. Pentru a incrementa temperatura de 10 ori, avem nevoie de tag-ul Contor. Vom afisa evolutia in timp a temperaturii folosind un obiect "Trend_view".

Pentru declansarea incrementarii temperaturii, vom plasa un buton "Start", pe a carui eveniment "Click", plasam functia "Incr_temp()".



```

Sub Incr_temp()
    Dim v_s
    Contor=0
    v_s=Second(Now())
    Do While Contor < 10
        If v_s <> Second(Now()) Then
            Contor=Contor+1
            Temperatura=Temperatura+10
            If Temperatura>100 Then
                Temperatura=50*Rnd()
            End If
            v_s=Second(Now())
        End If
    Loop
End Sub

```

Dupa cum se observa in functia Incr_temp(), pentru a incrementa temperatura la fiecare secunda, s-a utilizat functia Second(Now()) care ne furnizeaza valoarea numerica a secundelor din ora sistem data de functia Now().

Instructiunea do until :

Do Until expresie relationala
declaratii

.
.
 .

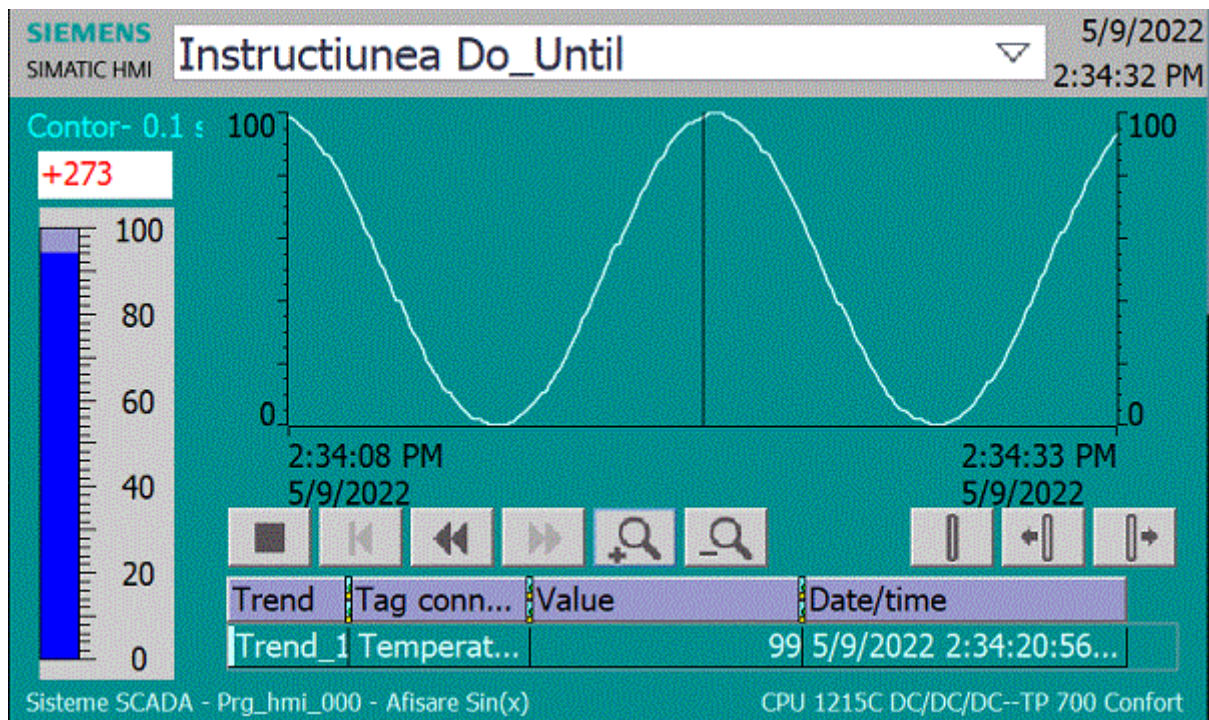
declaratii

Loop

Vom adauga screen-ul "Instr_do_until" in care vom folosi instructiunea Do Until pentru a da temperaturii o variatie sinusoidala. Vor fi generate 250 de valori la un interval de 0.1 secunde. Tag-ul Contor va fi incrementat la fiecare 0.1 secune.

Vom afisa evolutia in timp a temperaturii folosind un obiect "Trend_view".

Pentru declansarea modificarea temperaturii, vom plasa un buton, pe a carui eveniment "Click", plasam functia "Sin_temp()".



```

Sub Sin_temp()
    Dim v_s,tm
    Contor=0
    v_s=Second(Now())
    tm=Timer()
    Do Until Contor >= 250
        If Timer > tm+0.10 Then
            Contor=Contor+1
            Temperatura=50*(1+Sin(Contor/20))
            tm=Timer()
        End If
    Loop
End Sub

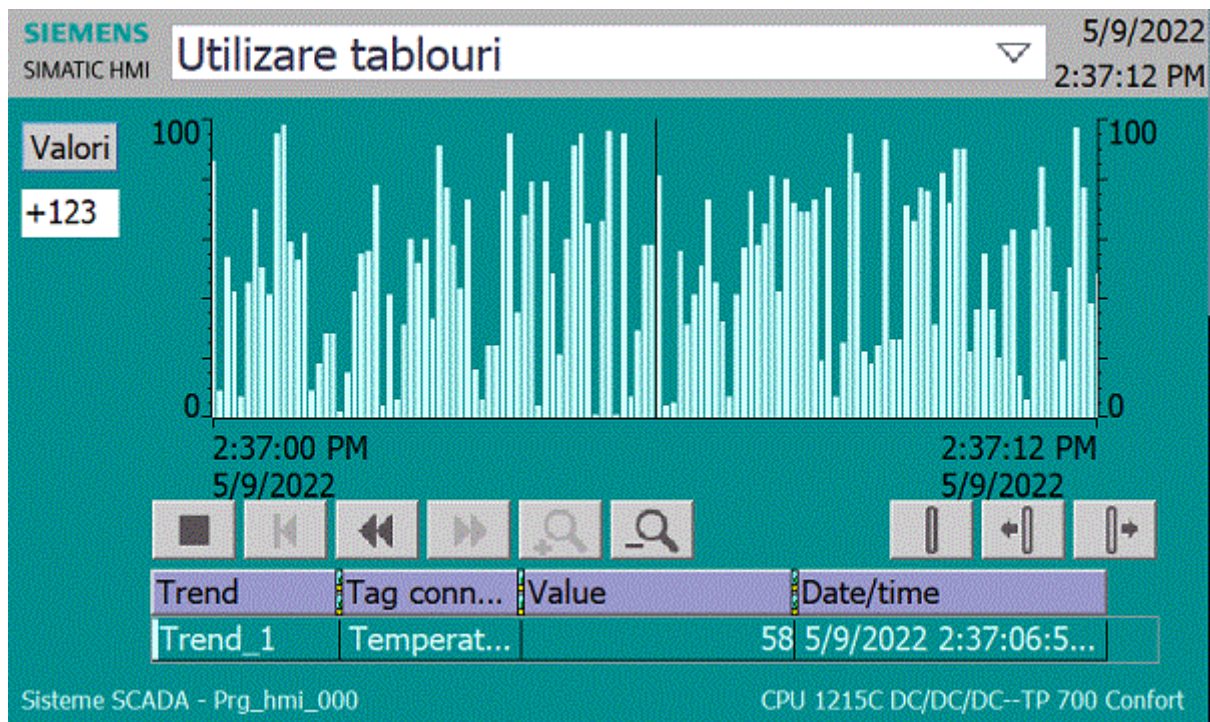
```

Dupa cum se observa in functia Sin_temp(), pentru a modifica temperatura la fiecare zecime de secunda, s-a utilizat functia Timer() care ne furnizeaza valoarea numerica a timpului trecut incepind de la ora 12.00 - am.

3. Diverse aplicatii

Utilizarea tablourilor

Urmatoarul screen "Utiliz_vetor" defineste si utilizeaza un tablou unidimensional(vector) care inscrie 125 valori Rnd() intr-un vector dupa care afiseaza valorile din vector folosind un obiect "Trent view".



La apasarea butonului "Valori" se lanseaza functia "Vector_Rnd()":

```
Sub Vector_Rnd()
    Dim Valori(125),tm,i
    For i=0 To 125
        Valori(i) = 100*Rnd()
    Next
    Contor=0
    tm=Timer()
    Do Until Contor >= 124
        If Timer > tm+0.1 Then
            Contor=Contor+1
            Temperatura=Valori(Contor)
            tm=Timer()
        End If
    Loop
End Sub
```

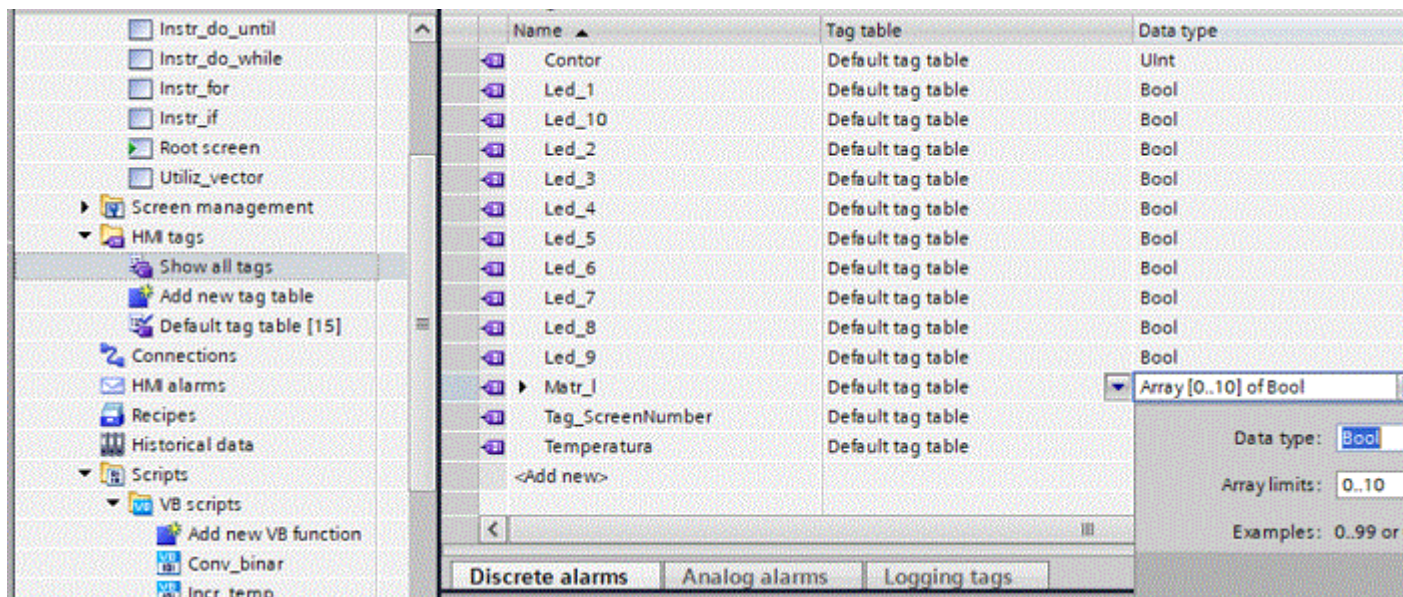
Obiectul "Trent view" are setat "Style" de tip "Bar", "Source setting: "Temperatura" si "Cyclic" la 0.1s motiv pentru care atribuirea valorilor pentru Tag-ul "Temperatura" se face la intervale de 0.1s

Adresarea indirecta a tag-urilor

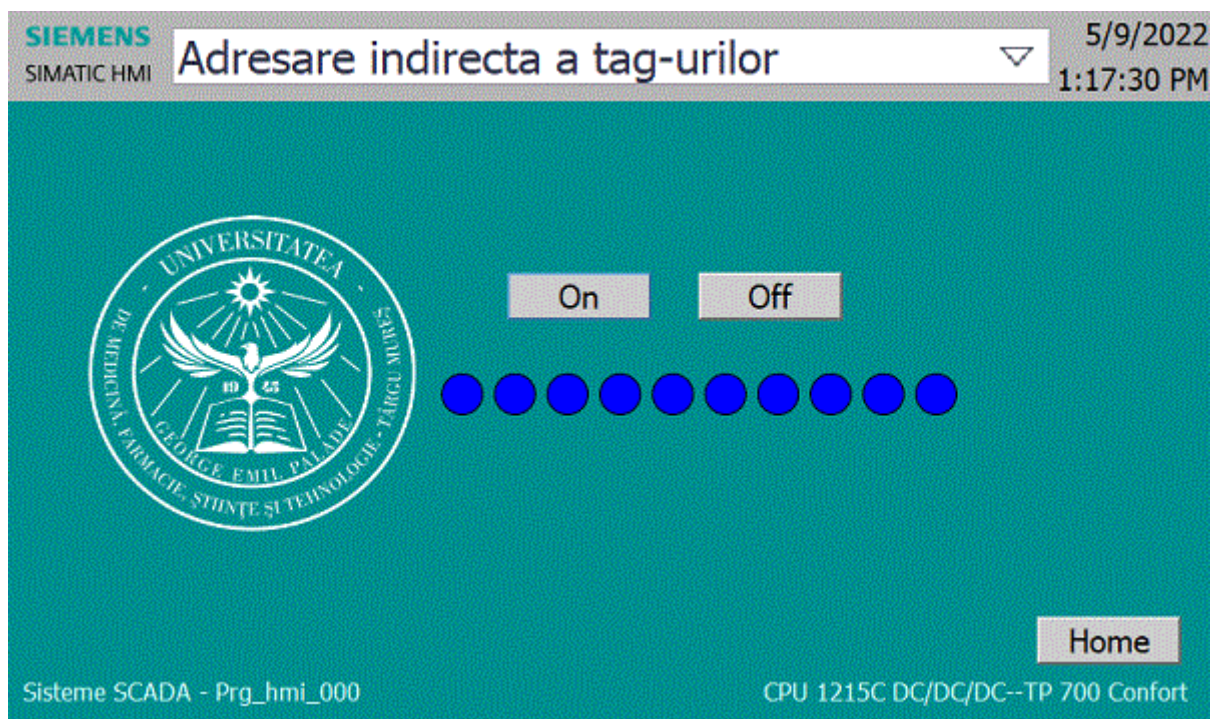
Urmatorul screen "Indirect_tag" defineste o serie de Tag-uri care vor fi apelate iterativ dintr-un VB Script.

Se prezinta un mod de adresare indirecta a Tag-urilor pentru cazurile in care avem mai multe Tag-uri de acelasi fel neputamd defini un Tag de tip "Array". In acest caz numele Tag-

urilor va trebui sa fi dat astfel incat sa putem genera iterativ numele acestor Tag-uri in vederea accesarii acestora. De exemplu vom defini Tag-urile Led_1...Led_10 de tip Bool-ean.



Plasam pe screen-ul "Indirect_tag" 10 LED-uri si setam proprietatea Animations --> Apperance --> Led_1 respectiv Led_10 pentru cele 10 LED-uri.
Plasam pe screen-ul "Indirect_tag" doua butoane.



La apasarea butonului "On" se lanseaza functia "Ind_tag_on()":

```
Sub Ind_tag_on()
```

```

    Dim i
    For i=1 To 10
        SmartTags("Led_"+CStr(i)) = True
    Next
End Sub

```

La apasarea butonului "Off" se lanseaza functia "Ind_tag_off()":

```

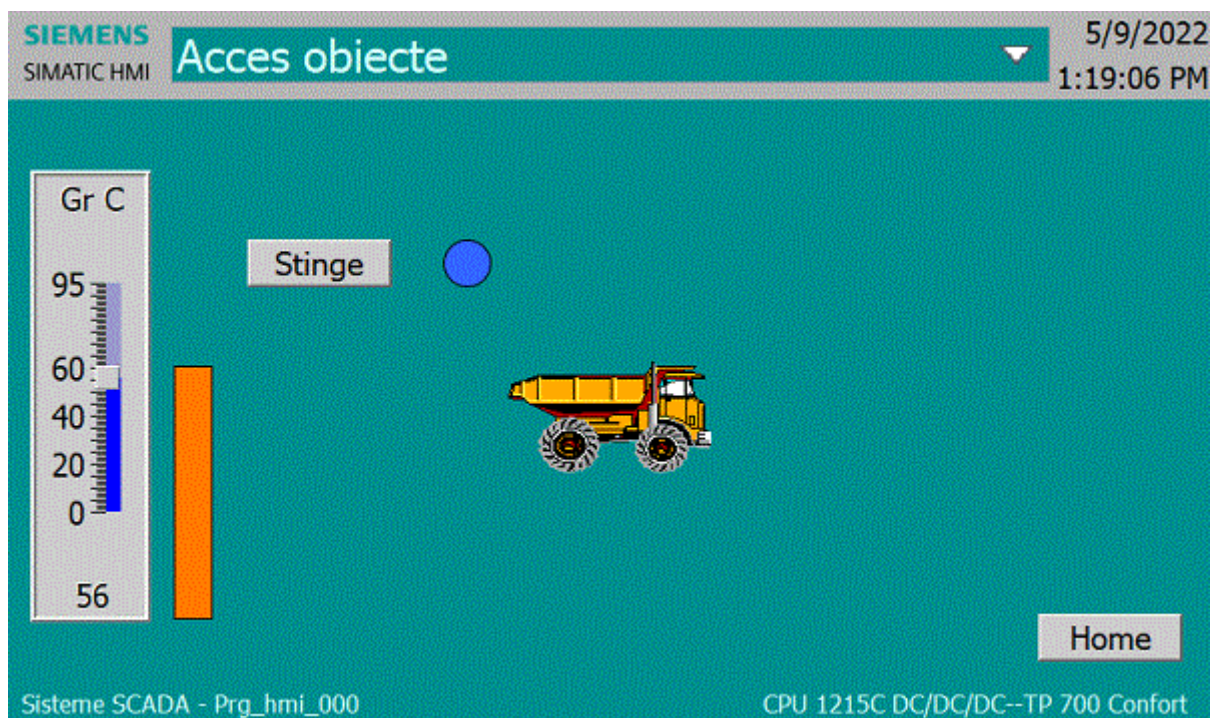
Sub Ind_tag_off()

    Dim i
    For i=1 To 10
        SmartTags("Led_"+CStr(i)) = False
    Next
End Sub

```

Accesarea din VB Scripts a obiectelor plasate pe screen

In urmatorul screen "Acces_elem", se acceseaza din diverse Script-uri obiectele plasate pe acest Screen.



Atribuim evenimentului "Change" al obiectului "Slider_1", script-ul "Acc_elem".

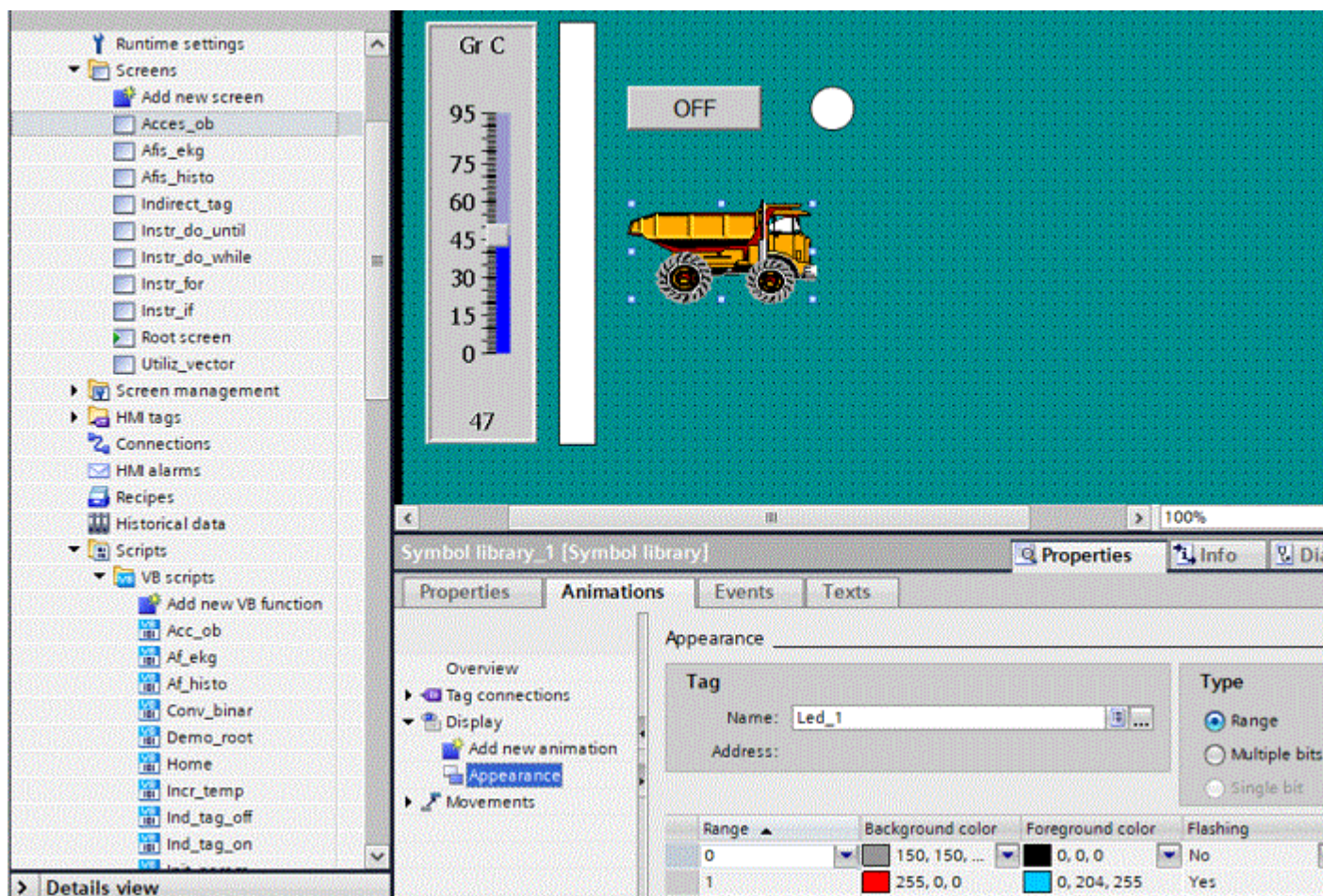
```

Sub Acc_elem()
    Dim dr, btn, sw, im
    Set dr =
HmiRuntime.Screens("Acces_elem").ScreenItems("Rectangle_1")
    Set btn = HmiRuntime.Screens("Acces_elem").ScreenItems("Circle_1")
    Set sw = HmiRuntime.Screens("Acces_elem").ScreenItems("Switch_1")
    Set im = HmiRuntime.Screens("Acces_elem").ScreenItems("Symbol
Library_1")
    dr.BackColor = RGB(255, 255 - 2.4 * Temperatura, 0)
    dr.Height = 1 + 3 * Temperatura
    dr.Top = (300 - 3 * Temperatura) + 45
    sw.TextOff = "Stins"
    sw.TextOn = "Aprins"
    im.Left = 265 + 3 * Temperatura
    Set dr = Nothing
    Set btn = Nothing
    Set sw = Nothing
End Sub

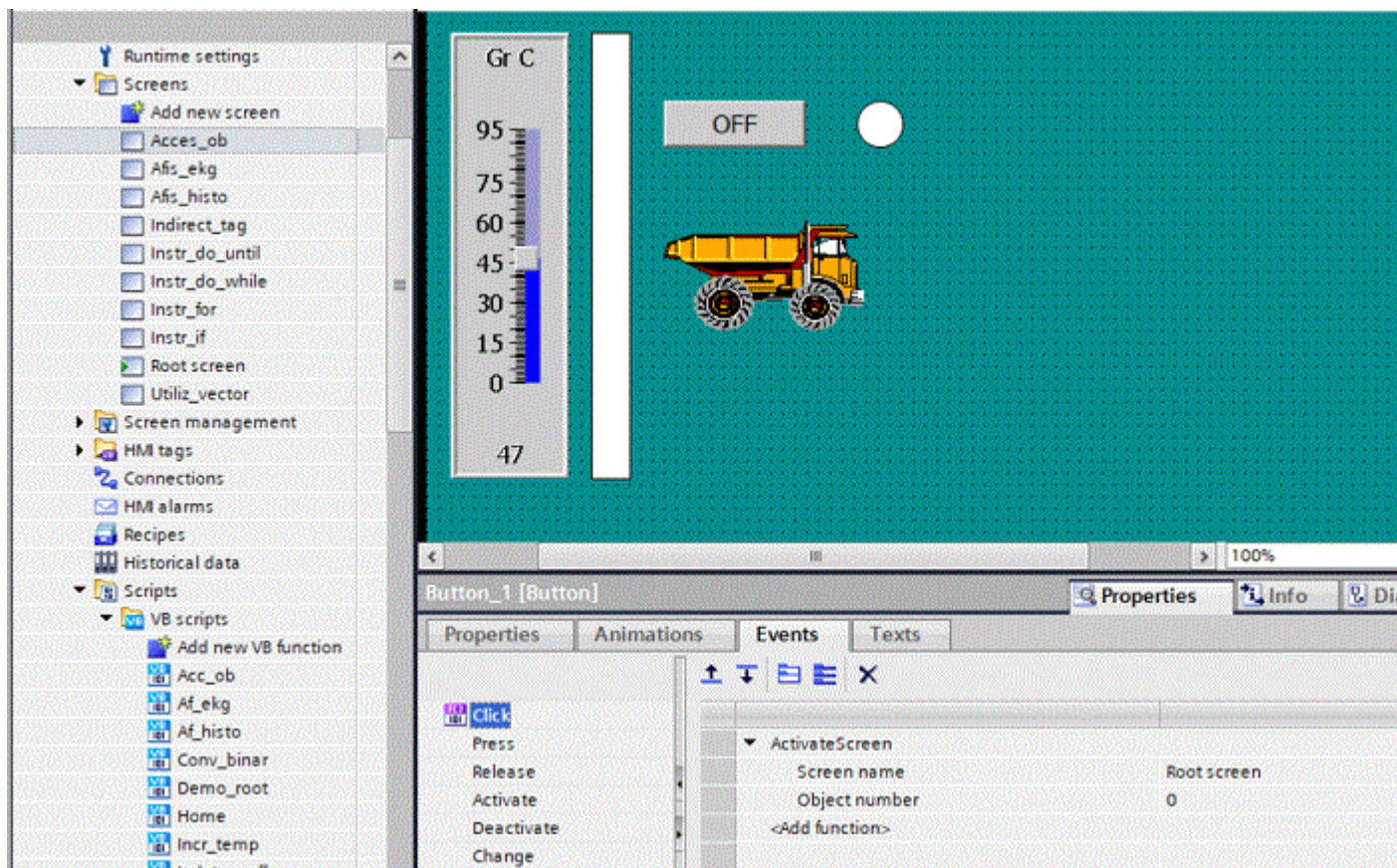
```

Dupa cum se observa, se atribuie variabilelor dr, btn, sw, im obiectele: "Rectangle_1", "Circle_1", "Switch_1", "Symbol Library_1" dupa care se acceseaza din script metodele sau proprietatile acestor obiecte.

Pentru a realiza "Flesh" LED si imaginea plasata, se adauga Animatia si se seteaza optiunea "Flashing"



La apasarea butonului "Home" activeaza Screen-ul "Root screen" deoarece s-a atribuit evenimentului al acestui buton functia predefinita "Activate screen"



Putem realiza propria noastra functie numita "Activ_screen" sub forma:
Plasam butonul "Instr_for" si atribuim functia "Activ_screen" evenimentului "Click"

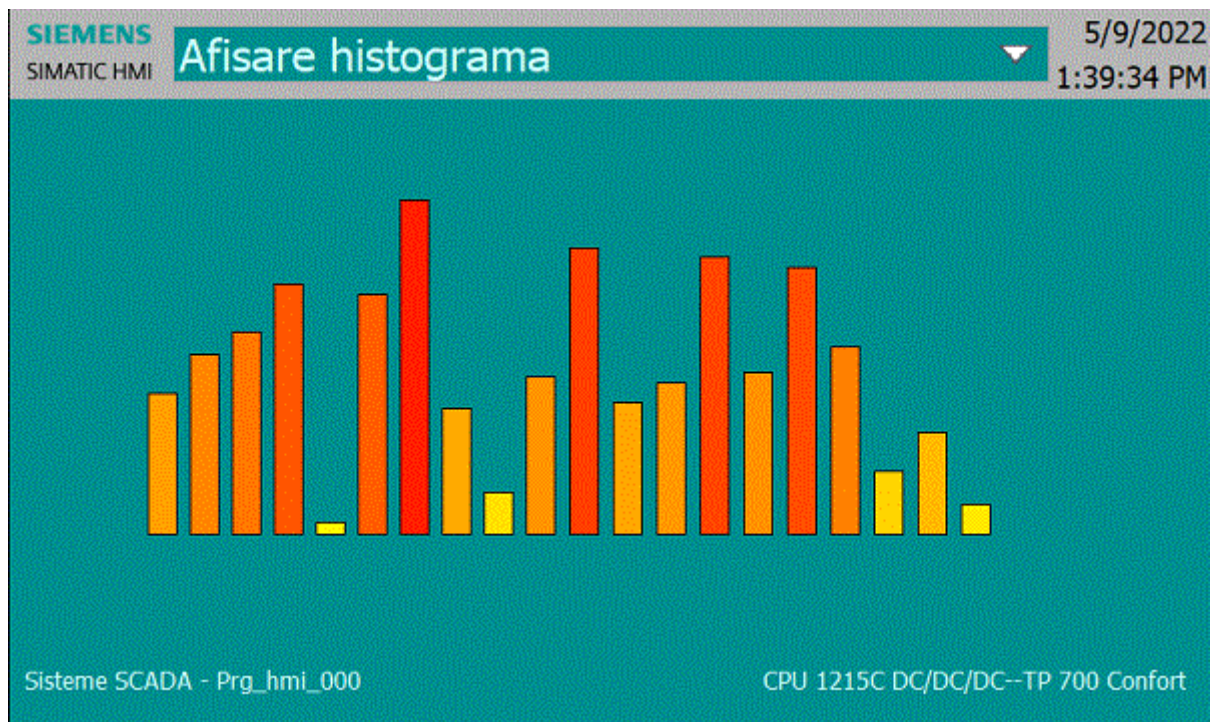
```
Sub Activ_screen()  
    ActivateScreen "Instr_for",0  
End Sub
```

Afisare histograma

In urmatorul screen "Afis_histo" vrem sa afisam 20 de valori aleatoare pe o histograma (grafic sub forma de bare). Valorile vor fi pastrate in tag-ul "Valori" de tip Array ce contine 25 elemente de tip int.

Plasam pe screen-ul "Afis_histo" 20 de dreptunghiuri: "Rectangle_1", "Rectangle_2...", "Rectangle_20" de inaltime 250 care vor reprezenta barele histogramei. Vom modifica inaltimea fiecarui dreptungi din script-ul "Af_histo()" in functie de valorile stocate in tag-ul "Valori"

Plasam pe screen-ul "Afis_histo" butonul "Start".



La apasarea butonului "On" se lanseaza functia "Af_histo()":

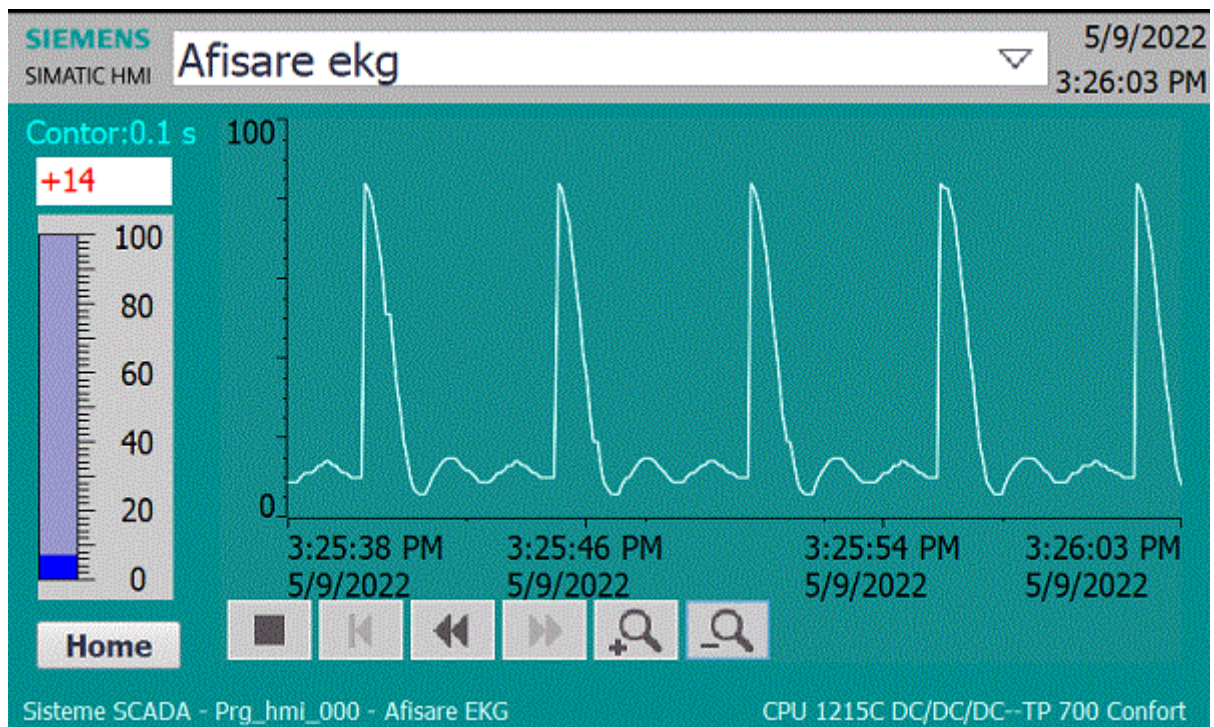
```
Sub Af_histo()
    Dim dr,i
    i=0
    For i=1 To 20
        Set dr =
HmiRuntime.Screens("Afis_histo").ScreenItems("Rectangle_"+CStr(i))
        Valori(i)=Int(250*Rnd())
        dr.Height=0
        dr.Top=(251-Valori(i))+40
        dr.Height=Valori(i)
        dr.BackColor = RGB(255,255-Valori(i),0)
        Set dr = Nothing
    Next
End Sub
```

Dupa cum se observa, accesarea dreptunghiurilor se face iterativ prin adresare indirecta a obiectelor de tip "Rectangle". Numele obiectului se construiește la fiecare iteratie. Inainte de setarea proprietatii "Top" a dreptunghiului curent, se seteaza proprietatea "Height" la 0 in caz contrar s-ar putea genera o eroare (pentru cazul in care vechea valoare "Height"+ noul "Top" iese din ecran), dupa care putem seta valoarea "Top", urmeaza "Height" si "BackColor"

Afisare ekg

In urmatorul screen "Afis_ekg" simulam intr-un control de tip "trend view" un semnal ekg.

Definim Tag-ul "Trend_ekg" tag care va fi utilizat pentru afisarea in controlul de tip "trend view".



Apelam pe evenimentul "Loaded" al screen-ului "Afis_ekg", functia "Af_ekg()":

```

Sub Af_ekg()
    Dim i,x,n,tm,l,a,b
    Dim harm,qrs1,qrs2,qrswav,gr
    Set gr = HmiRuntime.Screens("Afis_ekg").ScreenItems("Trend
view_1")
    Contor=0
    tm=Timer()
    Do While SmartTags("Tag_ScreenNumber")=10
        If Timer > tm+0.10 Then
            Contor=Contor+1
            If Contor > 50 Then
                Contor=0
            End If

            '-----
            x=Contor*0.0045
            l = 4                                ' li=4
            a = 1.6                                ' a_qrswav=1.6
            b = (2 * l) /0.09                        ' d_qrswav=0.09
            n = 100
            qrs1 = (a / (2 * b)) * (2 - b)
            qrs2 = 0.0
            For i = 1 To n
                harm = (((2 * b * a) / (i * i * 3.14 * 3.14)) * (1 -
Cos((i * 3.14) / b))) * Cos((i * 3.14 * x) / l)
                qrs2 = qrs2 + harm
            Next
            qrswav = qrs1 + qrs2
            '-----
            Trend_ekg = 55*(1+qrswav)
            tm=Timer()
        End If
    End Do
End Sub

```

```

Loop
Contor=0
End Sub

```

Dupa cum se observa, s-a folosit de data aceasta instructiunea **Do While SmartTags("Tag_ScreenNumber")=10**

Aplicatia ruleaza atata timp cat suntem pe screen-ul 10.

Plasarea unui buton "Home" este eficient numai daca vom folosi atat evenimentul "Click" cart si evenimentul "Release" astfel:

- Pe evenimentul "Click" vom Seta tag-ul Tag_ScreenNumber la 1, fortand astfel iesirea din instructiunea repetitiva
Do While SmartTags("Tag_ScreenNumber")=10
- Pe evenimentul "Release" vom lansa functia "Home"

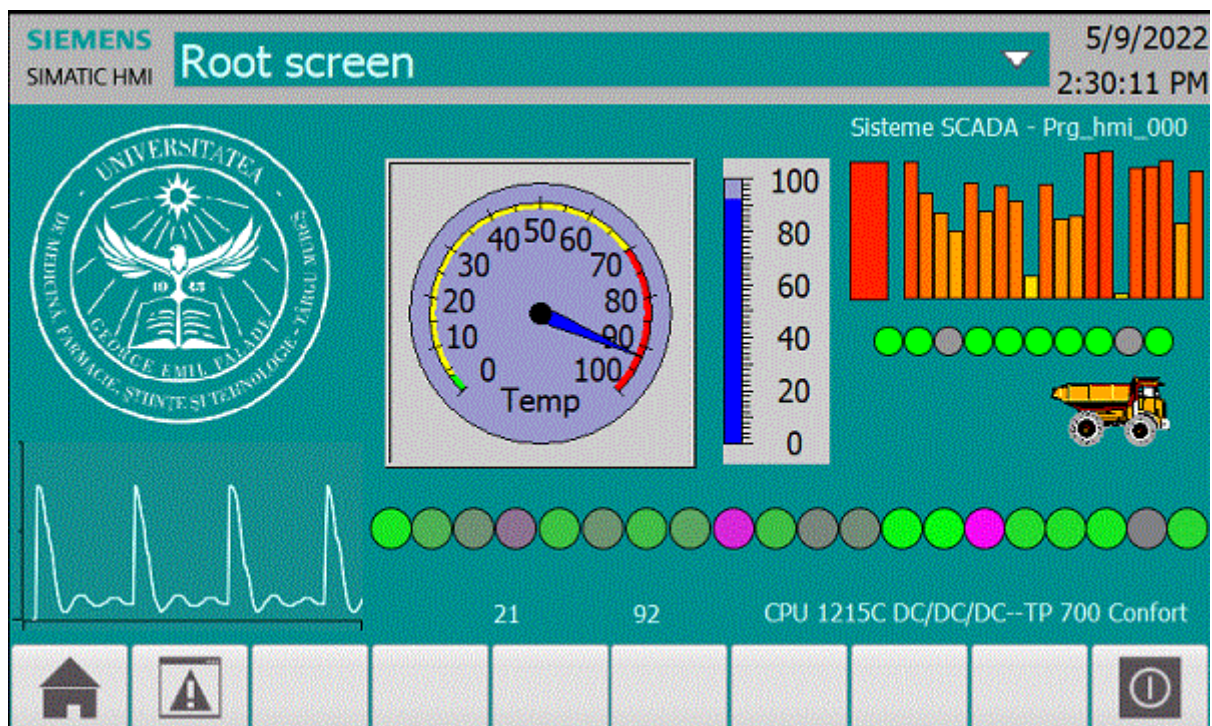
```

Sub Home()
    SmartTags("Tag_ScreenNumber")=1
    ActivateScreen "Root screen",1
End Sub

```

Afisare Root screen

Vom cumula aplicatiile realizate pana acum si le vom plasa pe screen "Root screen".



Pe evenimentul "Loaded" al screen-ului "Init_param()" si functia "Demo_root()".

```
Sub Init_param()  
    Temperatura=20  
    Contor=0  
End Sub
```

```
Sub Demo_root()  
    Dim tm, dr, dr21,i,j,im,crc  
    Set im = HmiRuntime.Screens("Root screen").ScreenItems("Symbol  
library_1")  
    Set dr21 = HmiRuntime.Screens("Root  
screen").ScreenItems("Rectangle_21")  
    tm=Timer()  
    Do While SmartTags("Tag_ScreenNumber")=1  
        If Timer > tm+0.5 Then  
            For i=1 To 20  
                Set dr = HmiRuntime.Screens("Root  
screen").ScreenItems("Rectangle_"+CStr(i))  
                Valori(i)=Int(100*Rnd())  
                dr.Height=0  
                dr.Top=(120-Valori(i))+10  
                dr.Height=Valori(i)  
                dr.BackColor = RGB(255,255-2*Valori(i),0)  
                Set dr = Nothing  
                Set crc = HmiRuntime.Screens("Root  
screen").ScreenItems("Circle_"+CStr(i+10))  
                'crc.Top=(330-Valori(i))  
                crc.BackColor = RGB(255-  
2.5*Valori(i),2.5*Valori(i),255-2.4*Valori(i))  
                Set crc = Nothing  
            Next  
  
            'EKG -----  
  
            Dim x,n,l,a,b,harm,qrs1,qrs2,qrswav  
            x=Contor*0.0045  
            l = 4 ' li=4  
            a = 1.6 ' a_qrswav=1.6  
            b = (2 * l) /0.09 ' d_qrswav=0.09  
            n = 100  
            qrs1 = (a / (2 * b)) * (2 - b)  
            qrs2 = 0.0  
            For i = 1 To n  
                harm = (((2 * b * a) / (i * i * 3.14 * 3.14)) * (1 -  
Cos((i * 3.14) / b))) * Cos((i * 3.14 * x) / l)  
                qrs2 = qrs2 + harm  
            Next  
            qrswav = qrs1 + qrs2  
            Trend_ekg = 50*(1+qrswav)
```

```

' -----

Temperatura=Valori(1)
Contor=Contor+1
If Contor > 50 Then
    Contor=0
End If
im.Left=555+1.5*Temperatura
dr21.BackColor = RGB(255,255-2.4*Temperatura,0)
dr21.Height=1+1*Temperatura
dr21.Top=(120-1*Temperatura)+10
tm=Timer()
Ld_Rnd()
End If
Loop
End Sub

```

Rezumat

- **Limbajul de programare VB (Visual Basic) Script**

Definirea functiilor

```

Sub nume_script()
    declaratii
    .
    .
    .
    declaratii
End Sub

```

Definirea tablourilor

```

Dim nume_tablou(nr_elemente)

```

Instructiunea if :

```

If expresie relationala Then
    instructiune(instructiuni)
End

```

sau

If expresie relationala **Then**
 instructiune(instructiuni)
Else
 instructiune(instructiuni)
End

Instructiunea for :

For contor=contor_start **To** contor_stop
 declaratii
 .
 .
 .
 declaratii
Next

Instructiunea do while :

Do While expresie relationala
 declaratii
 .
 .
 .
 declaratii
Loop

Instructiunea do until :

Do Until expresie relationala
 declaratii
 .
 .
 .
 declaratii
Loop

• Rezultate asteptate

Dupa studierea acestui modul, ar trebui sa cunoasteti:

- Cum sa scrieti VB scripts
- Sa utilizati instructiuni decizionale si repetitive
- Sa definiti functii proprii
- Sa realizati aplicatii SCADA in care sa folositi elemente de programare VB Script

• Termeni esentiali

Termen	Descriere
SCADA	Supervisory Control And Data Aquisition
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 Visual Basic cu care se realizeaza VB Scripts
Instructiuni decizionale	Instructiuni care permit alegerea setului de instructiuni care urmeaza a fi executate in functie de o expresie relationala
Instructiuni repetitive	Instructiuni care permit rularea repetitiva functie de o expresie relationala, a unui setului de instructiuni
Expresie relationala	Expresie a carui rezultat este o valoare logica

• 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: Elemente de programare C++ utile in ingineria electrica, ISBN: 978-973-755-576-2, Ed. MatrixRom, 2009
- [7] T. Turc: Programare avansata C++ pentru ingineria electrica, ISBN: 978-973-755-588-5,Ed. MatrixRom, 2009.
- [8] Boldur Barbat - Informatica industrială - Programarea în timp real – Institutul Central pentru Conducere si informatica 1984
- [9] Ioan Babuita – Conducerea automata a proceselor – Ed. Facla 1985
- [10] Ghercioiu-National în struments - Orizonturi în instrumentatie 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 -

