

Aplicația 5 - Aplicații SCADA in energetică - Scheme monofilare

Obiective

- Realizarea unui proiect SCADA pentru simularea funcționării unei scheme monofilare.
- Utilizarea instrucțiunilor Citect și a funcțiilor definite de utilizator pentru implementarea funcționalității unei scheme monofilare.
- Realizarea unei pagini grafice care să simuleze conectarea secvențială a liniilor folosind instrucțiunea FOR și funcția predefinită sleep().
- Realizarea unei pagini grafice care să simuleze conectarea secvențială a liniilor și în care se condiționează acționarea separatoarelor de poziția intreruptoarelor

Prezentare teoretică

Scheme monofilare

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

O schemă monofilară conține o serie de elemente de comutație cum ar fi separatoare, intreruptoare, etc. Există o ordine în care se acționează cele două elemente, în sensul că separatorul nu poate fi niciodată acționat în sarcină.

În sistemele energetice, închiderea respectiv deschiderea unui separator sau a unui intreruptor nu este instantanee. În aplicațiile SCADA se ține cont de acest fenomen. După comanda acționării unui element se scurge un interval de timp până la efectuarea completa a comenzii. După acționarea unui element e nevoie de un semnal care să confirme efectuarea comenzii.

Pentru testarea aplicațiilor SCADA fără conectare la sistemul real de achiziție și comandă, închiderea respectiv deschiderea elementelor de comutație, ar putea fi simulată printr-o mărime analogică, având valori între 0 și 100%. Pe schema, această valoare ar putea fi simulată prin utilizarea unui control de tip "Slider" care ar fi în legătura cu un tag de tip analogic.

Starea fiecărui element plasat pe HMI este controlată de tag-ul corespunzător.

În multe cazuri, în cadrul schemelor monofilare se întâlnesc mai multe elemente de același tip. În acest caz se recomandă definirea tag-urilor de tip Array și utilizarea instrucțiunilor repetitive.

În cazul sistemelor de alimentare cu energie, având mai multe linii, pentru a evita șocurile de sarcină, conectarea respectiv deconectarea acestora se face secvențial într-o ordine prestabilită, în funcție de importanța utilizatorului. Pentru a memora prioritatea consumatorilor, se recomandă utilizarea unui vector de priorități.

Tag-urile sunt actualizate de sistemul de achiziție și comandă. Acesta este prevăzut atât cu elemente pentru citirea stărilor cât și cu elemente de comandă.

Logica de funcționare a unei scheme monofilare se implementează într-o pagină grafică prin intermediul instrucțiunilor și funcțiilor scrise în limbajul "Cicode".

Instrucțiuni des utilizate:

- **Utilizarea instrucțiunilor decizionale**

Instrucțiunea **if** se folosește pentru a selecta execuția unei instrucțiuni (sau a unui grup de instrucțiuni) funcție de valoarea logică a unei expresii relaționale

Formatul instrucțiunii:

Instrucțiunea **if** are următoarele formate:

```
If expresie relațională THEN  
    instrucțiune(instrucțiuni);  
END
```

sau

```
If expresie relațională THEN  
    instrucțiune(instrucțiuni);  
ELSE  
    instrucțiune(instrucțiuni);  
END
```

- **Utilizarea funcțiilor**

Formatul pentru definirea unei funcții fără parametri și fără returnare de valori:

Pentru definirea unei astfel de funcții se folosește următorul format:

```
FUNCTION nume_funcție()  
    declarații;  
    .  
    .  
    .  
    declarații;  
END
```

Formatul pentru definirea unei funcții cu parametri și fără returnare de valori:

Pentru definirea unei astfel de funcții se folosește următorul format:

```
FUNCTION nume_funcție(Argumente)  
    declarații;  
    .  
    .  
    .  
    declarații;  
END
```

Formatul pentru definirea unei funcții cu parametri și cu returnare de valori:

Pentru definirea unei astfel de funcții se folosește următorul format:

```
Tip valoare returnată FUNCTION nume_funcție(Argumente)  
    declarații;  
    .  
    .  
    .  
    declarații;  
    RETURN valoare  
END
```

- **Utilizarea instrucțiunilor repetitive**

Instrucțiunea **for** , se folosește pentru a executa repetitiv o instrucțiune sau o secvență de instrucțiuni. De obicei implementează structura ciclica cu număr cunoscut de pași.

Formatul instrucțiunii:

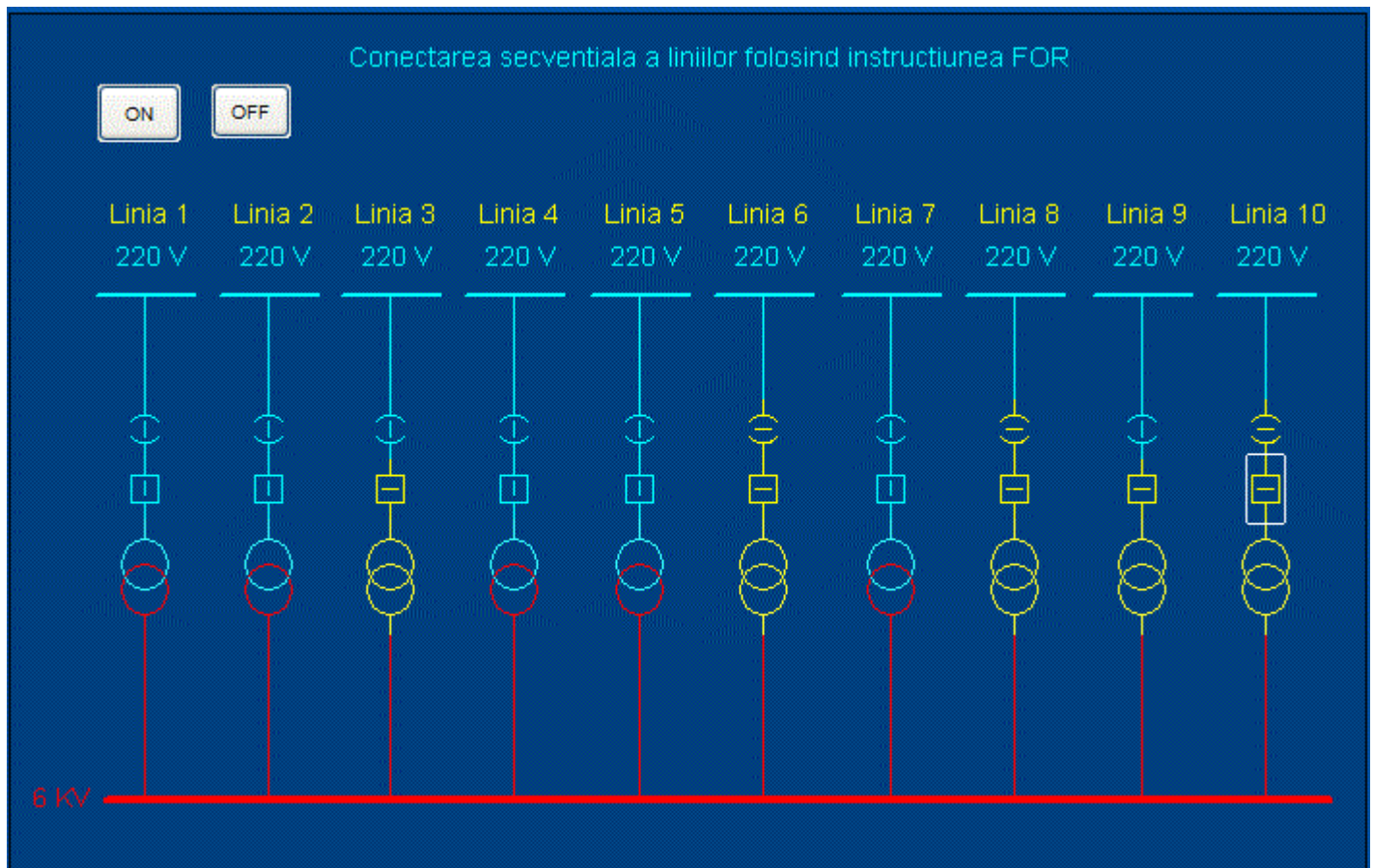
Instrucțiunea **for** are următorul format:

```
FOR Variabila=expresie1 TO expresie2 DO  
  instrucțiune(instrucțiuni;)  
END
```

Unde expresie1 este valoarea de start a variabilei iar expresie2 este valoarea de stop a variabilei.

Tematica de laborator

Utilizând mediul de dezvoltare SCADA-CITECT creați o pagină grafică care să simuleze o schemă monofilară în care se conectează secvențial liniile. Se va folosi instrucțiunea FOR și funcția predefinită sleep(). Se va crea un nou proiect cu numele **Labs** în care se va crea pagina **labs5_01** similară cu pagina grafică de jos:



În cadrul acestei scheme monofilare se întâlnesc mai multe elemente de același tip. Vom defini tag-uri de tip Array și vom utiliza instrucțiuni repetitive pentru controlul acestor tag-uri.

Pentru închiderea respectiv deschiderea elementelor de comutație, vom introduce două tag-uri de tip Array:

Tag-uri aferente					
Nume	Tip	Domeniu	Um	Array Size	Comentariu
sep	DIGITAL	-	-	11	Separatoarele sep[1]-sep[10]
intr	DIGITAL	-	-	11	Intreruptoarele intr[1]-intr[10]

Pe evenimentele click ale butoanelor "ON" respectiv "OFF" sunt afectate doua variabile tag locale:

Tag-uri aferente					
Nume	Tip	Domeniu	Um	Comentariu	
cmd_on	DIGITAL	-	-	Comanda inchiderea elementelor de comutatie	
cmd_off	DIGITAL	-	-	Comanda deschiderea elementelor de comutatie	
i	INT	-	-	Index	

Se stabilește vectorul de priorități de conectare sub forma:

INT Prior[11]=0,1,5,7,4,2,9,3,6,8,10;

Se va crea funcția atribuita butonului ON numita: FUNCTION sch1_on() respectiv funcția atribuită butonului OFF numită: FUNCTION sch1_off().

```

FUNCTION sch1_on()
    INT i;
    FOR i=1 TO 10 DO
        sep[Prior[i]]=1;
        Sleep(1);
        intr[Prior[i]]=1;
        Sleep(1);
    END
END

FUNCTION sch1_off()
    INT i;
    FOR i=1 TO 10 DO
        intr[Prior[11-i]]=0;
        Sleep(1);
        sep[Prior[11-i]]=0;
        Sleep(1);
    END
END

```

Se va crea pagina **labs5_02** similară cu pagina grafică **labs5_01** cu deosebirea ca în această pagina se va tine cont de ordinea de comutare referitoare la separatori și intreruptori. Se va rescrie deci funcția atribuită butonului ON și se va numi: FUNCTION sch2_on() respectiv funcția atribuită butonului OFF și se va numi: FUNCTION sch2_off()

```

FUNCTION sch2_on()
    INT i;
    FOR i=1 TO 10 DO
        IF intr[Prior[i]]=0 THEN
            sep[Prior[i]]=1;
        END
        Sleep(1);
        intr[Prior[i]]=1;
        Sleep(1);
    END
END

FUNCTION sch2_off()
    INT i;
    FOR i=1 TO 10 DO
        intr[Prior[11-i]]=0;
        Sleep(1)
        IF intr[Prior[11-i]]=0 THEN
            sep[Prior[11-i]]=0;
        END
        Sleep(1);
    END
END

```

Cerințe de rezolvat

- Crearea unui nou proiect "Labs"
- Realizarea paginii grafice "labs5_01"
- Implementarea și verificarea funcționalității pagini grafice "labs5_01"
- Realizarea paginii grafice "labs5_02"
- Implementarea și verificarea funcționalității pagini grafice "labs5_02"

Derularea activităților

- Se crează nou proiect "Labs"
- Dacă nu se reușește crearea unui nou proiect valid, se poate utiliza proiectul "**Sch_el_start**" proiect în care au fost parcurși toți pașii pentru crearea unui nou proiect.
Proiectul se poate descărca de pe www.science.upm.ro/~traian
După ce s-a download-at acest fișier, din Citect Explorer->Restore se încarcă acest proiect și i se atribuie numele "**Labs**".
- Se realizează pagina grafică "labs5_01"
- Se introduc elementele de funcționalitate
- Se realizează pagina grafică "labs5_02" în care se vor introduce și elemente de condiționalitate între separatoare și întreruptoare
- Se introduc elementele de funcționalitate

Prezentarea rezultatelor

Punctarea activitatilor (total 6 puncte)		
Nr	Denumire activitate	Punctaj
1	Se verifica aspectul grafic al pagini grafice "labs5_01"	1
2	Se verifica functionalitatea pagini grafice "labs5_01"	1
3	Se verifica aspectul grafic al pagini grafice "labs5_02"	2
4	Se verifica functionalitatea pagini grafice "labs5_02"	2