

**UNIVERSITATEA “PETRU MAIOR”
TÂRGU MURES**

**FACULTATEA DE INGINERIE
ELECTRICĂ**

PROIECT
SISTEME SCADA

Şef lucr. Dr. Ing. Turc Traian

Masterand: Rogoz Radu Ionuţ

Instalație automată pentru îmbutelierea laptelui

Tema Proiectului: *Realizarea unui sistem SCADA pentru monitorizarea și controlul unei instalații automate de îmbuteliere a laptelui utilizate în domeniul prelucrării laptelui .*

Structura Proiectului:

Prezentarea generală a sistemului SCADA

Proiectarea software

- **Construcția interfeței utilizator**
- **Implementarea**
- **Descrierea modului de funcționare a sistemului**

Bibliografie

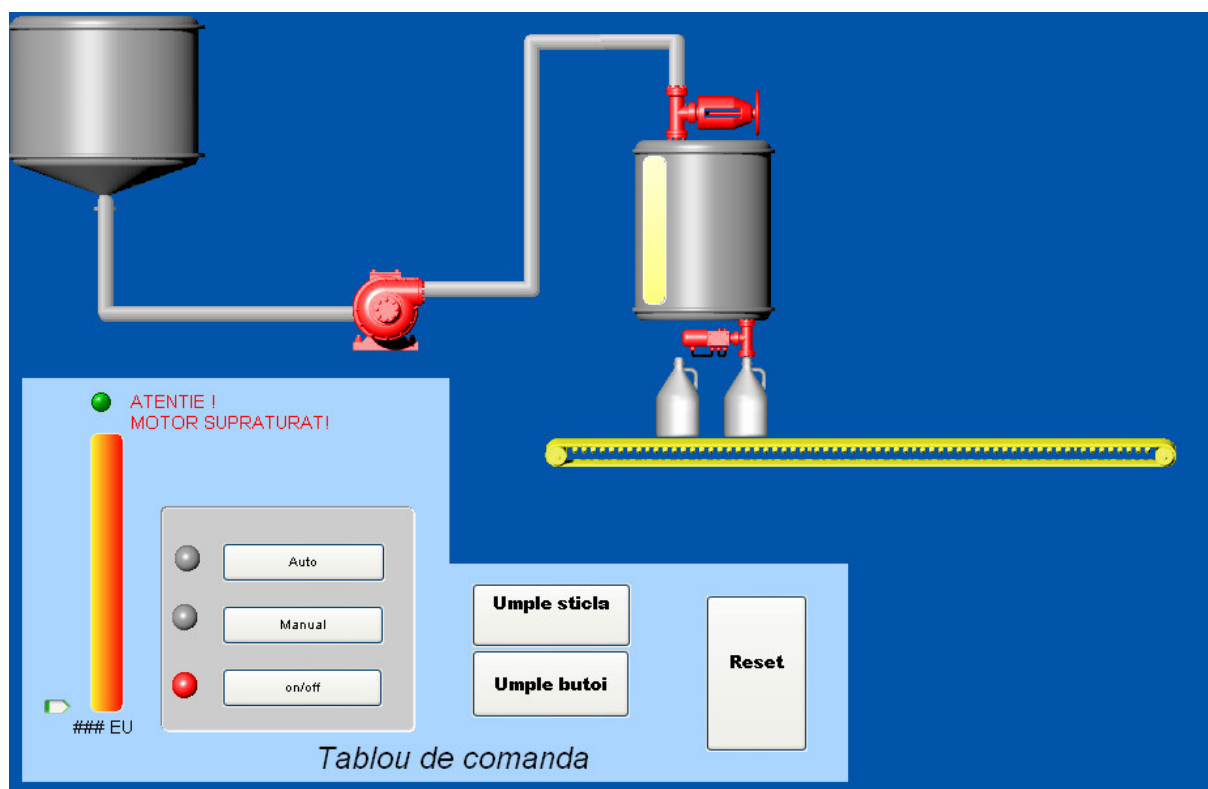
1. Prezentarea Sistemului SCADA

Un sistem SCADA prezintă de regulă informația operatorului sub forma unei schite sugestive. Aceasta înseamnă că operatorul poate vedea o reprezentare a instalației supravegheate. De exemplu, o imagine a unui sistem care poate afișa operatorului faptul că acesta lucrează sau nu, și condițiile de funcționare ale sistemului cu valorile parametrilor la un moment dat. Operatorul poate controla sistemul, software-ul HMI afișând variațiile parametrilor în timp real. HMI reprezintă Interfața om-mășină (*Human Machine Interface*). Pachetul HMI/SCADA include de obicei un program de desenare pe care operatorul sau personalul de întreținere îl folosește pentru a schimba modul în care punctele sunt reprezentate în interfața utilizator. Industria de HMI/SCADA a apărut din nevoia unui terminal prietenos pentru utilizator într-un sistem alcătuit cu unități PLC. Un PLC este programat să controleze automat un proces, însă faptul că unitățile PLC sunt distribuite într-un sistem amplu, colectarea manuală a datelor procesate de PLC este dificilă. De asemenea informațiile din PLC sunt de obicei stocate într-o formă brută, neprietenoasă. HMI/SCADA are rolul de a aduna, combina și structura informațiile din PLC printr-o formă de comunicație. Încă din anii 1990 rolul sistemelor SCADA în sistemele ingineresti civile s-a schimbat, necesitând o mai mare cantitate de operațiuni executate automat. Un HMI elaborat, poate fi de asemenea conectat la o bază de date pentru realizarea de grafice în timp real, analiza datelor, proceduri de întreținere planificate, scheme detaliate pentru un anumit senzor sau utilaj, precum și metode de depanare a sistemului. Din 1998, majoritatea producătorilor de PLC oferă sisteme HMI/SCADA integrate, cele mai multe folosind sisteme de comunicație și protocoale deschise, neproprietare. Majoritatea sistemelor HMI/SCADA oferă compatibilitate cu PLC - urile.

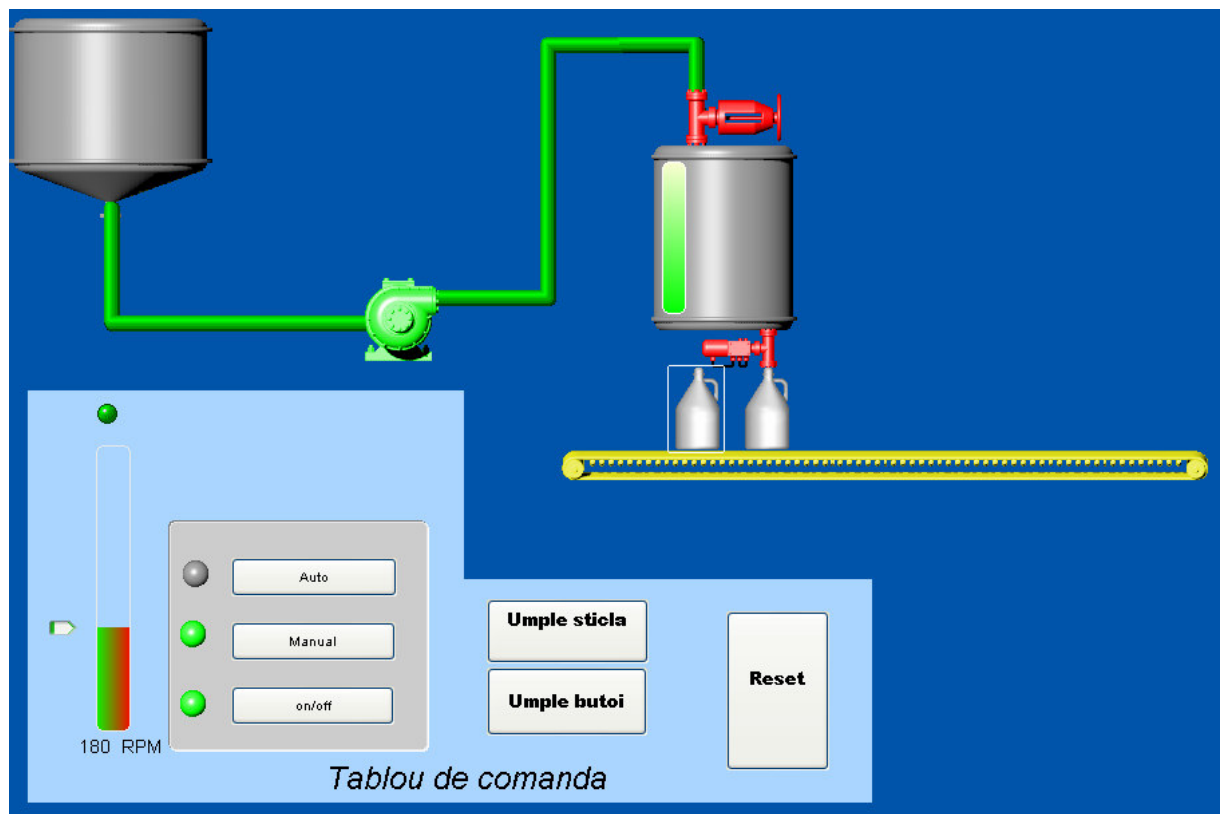
2. Construcția interfeței utilizator

La realizarea interfeței vizuale s-au folosit componente vizuale, specifice mediului CitectSCADA, pentru a reprezenta instalația automată reală. Simbolurile cele mai semnificative utilizate sunt :

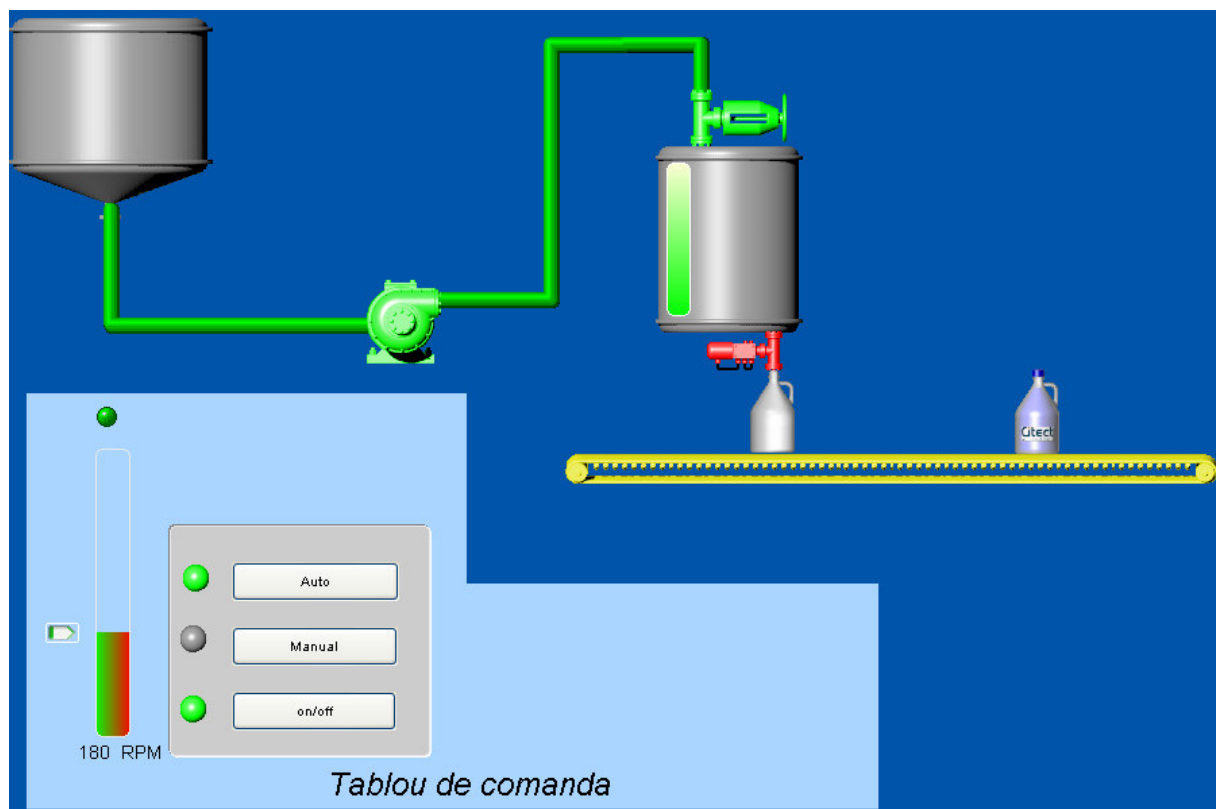
- 2 elemente de tip tank_cilindric;
- O pompa(se pornește/oprește de la butonul on/off);
- 2 robinete (sus robinet 1, jos robinet 2);
- 2 sticle;
- O banda rulanta pentru transport;
- Un tablou de comanda care conține șase butoane;



1. Butonul **on/off** –schimbă starea sistemului din oprit în pornit, respective din pornit în oprit.
2. Butonul **Manual** – setează sistemul în modul manual , eliberând butoanele Umare sticla, Umare butoi , și Reset (resetează sistemul).



3. Butonul **Auto** - seteaza sistemul in modul automat , blocand accesul la butoanele Uple sticla, Uple butoi , si Reset.



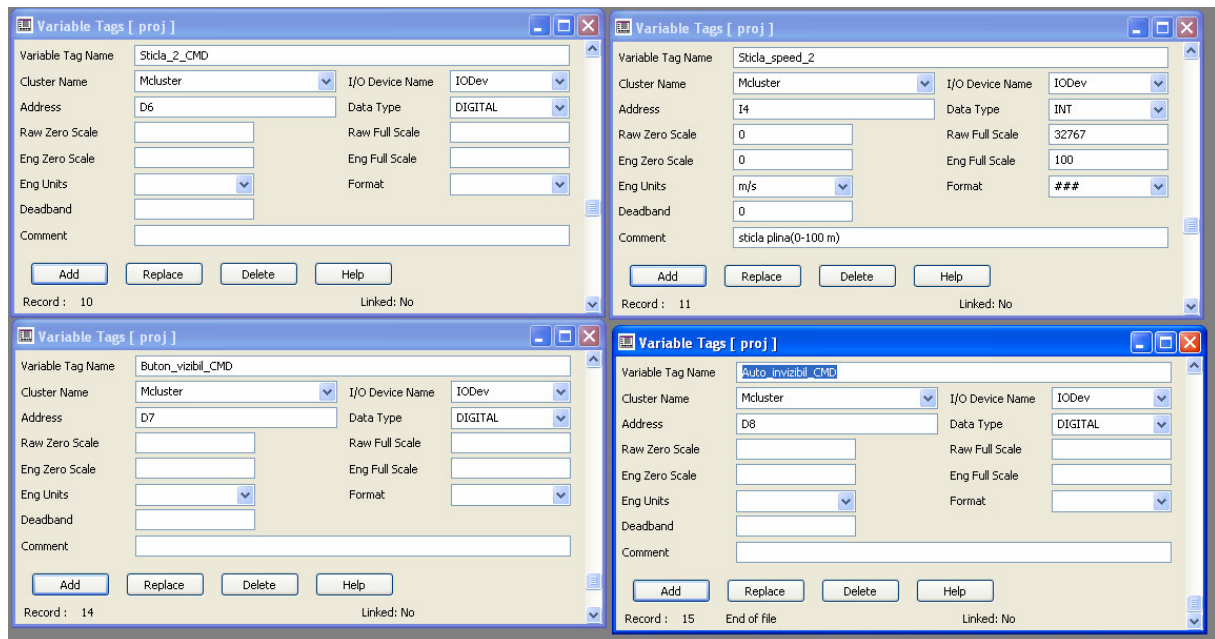
3. Configurarea variabilelor folosite:

The figure displays four screenshots of the 'Variable Tags' configuration window for a project named 'proj'. Each window shows the configuration for a specific variable tag, including its name, cluster, address, data type, scales, units, and comment.

- Record 1:** Variable Tag Name: Pump_1_CMD, Cluster Name: Mcluster, I/O Device Name: IODev, Address: D1, Data Type: DIGITAL, Comment: Pump 1 command-on/off.
- Record 2:** Variable Tag Name: Pump_1_Mode, Cluster Name: Mcluster, I/O Device Name: IODev, Address: D2, Data Type: DIGITAL, Comment: Pump 1 mode-0=auto, 1-Manual.
- Record 3:** Variable Tag Name: Pump_1_Speed, Cluster Name: Mcluster, I/O Device Name: IODev, Address: I1, Data Type: INT, Eng Units: RPM, Comment: Pump 1 Speed(0-500 rpm).
- Record 4:** Variable Tag Name: Butoi_1_speed, Cluster Name: Mcluster, I/O Device Name: IODev, Address: I2, Data Type: INT, Eng Units: litres, Comment: butoi plin(0-100 L).

The figure displays four screenshots of the 'Variable Tags' configuration window for a project named 'proj'. Each window shows the configuration for a specific variable tag, including its name, cluster, address, data type, scales, units, and comment.

- Record 5:** Variable Tag Name: Robinet_1_CMD, Cluster Name: Mcluster, I/O Device Name: IODev, Address: D3, Data Type: DIGITAL, Comment: robinet1 on/off.
- Record 6:** Variable Tag Name: Robinet_2_CMD, Cluster Name: Mcluster, I/O Device Name: IODev, Address: D4, Data Type: DIGITAL, Comment: robinet1 on/off.
- Record 8:** Variable Tag Name: Sticla_1_CMD, Cluster Name: Mcluster, I/O Device Name: IODev, Address: D5, Data Type: DIGITAL, Comment: (empty).
- Record 9:** Variable Tag Name: Sticla_speed_1, Cluster Name: Mcluster, I/O Device Name: IODev, Address: I3, Data Type: INT, Eng Units: m/s, Comment: sticla plina(0-100 m).



4. Funcții utilizate:

```
FUNCTION CIRCSPPEED ()
```

```
Pump_1_Speed=Pump_1_Speed+1;
```

```
END
```

```
FUNCTION plin()
```

```
//IF (Robinet_1_CMD=0)
```

```
IF (Butoi_1_speed<32767)
```

```
THEN
```

```
Robinet_1_CMD=1
```

```
END
```

```
IF (Butoi_1_speed<32767)
```

```
THEN
```

```
WHILE Butoi_1_speed<>32767
```

```
DO Butoi_1_speed=Butoi_1_speed+10
```

```
END
```

```
END
```

```
END
```

```
FUNCTION miscare_sticla2()
```

```
IF Sticla_speed_2=0 THEN
```

```
WHILE Sticla_speed_2<26
```

```
DO Sticla_speed_2=Sticla_speed_2+0.02
```

```

        END
ELSE
    Sticla_2_CMD=1
    WHILE Sticla_speed_2<100
    DO Sticla_speed_2=Sticla_speed_2+0.02
    END
END
END

```

```

FUNCTION sticla_plina()

```

```

    Robinet_1_CMD=0
    Robinet_2_CMD=1

    Butoi_1_speed=Butoi_1_speed-3000
    Sticla_1_CMD=1;
    WHILE Sticla_speed_1<100
    DO Sticla_speed_1=Sticla_speed_1+0.02
    END
    miscare_sticla2()

```

```

END

```

```

FUNCTION reset()

```

```

    Sticla_1_CMD=0;
    Sticla_2_CMD=0;

    Sticla_speed_1=0;
    Sticla_speed_2=0;

    Butoi_1_speed=32767;
    Pump_1_Speed=0;
    Pump_1_CMD=0;

    Buton_vizibil_CMD=0;

```

```

END

```

```

FUNCTION on_off()

```

```

    Toggle(Pump_1_CMD);
    Robinet_1_CMD=0;
    Robinet_2_CMD=0;
    Pump_1_Mode=1;

    IF (Buton_vizibil_CMD=0) THEN
        Buton_vizibil_CMD=1;
    ELSE Buton_vizibil_CMD=0;
    END

```

```

END

```

```

FUNCTION auto()

```

```

    Pump_1_CMD=1;

    sticla_plina();
    Robinet_2_CMD=0;
    plin();

```

```

END

```


5. Mod de funcționare:

Stare inițială : oprit.

Se apasă butonul **on/off** pentru a seta starea pornit a sistemului , se pornește pompa.

Pentru a seta viteza pompei se poate folosi butonul auto, ținând-ul apăsat, sau manual, mișcând indicatorul.

- Dacă se alege modul automat , efectul va fii următorul : robinet 1 închis , robinet 2 deschis , se va umple sticla , robinet 2 închis , după care se va umple rezervorul(robinet 1 deschis) .
- Dacă se alege modul manual , umplerea sticlei se va face cu ajutorul butonului Umple sticla (robinet 2 deschis, umplere sticlă, robinet 2 închis), iar umplerea butoiului , cu ajutorul butonului Umple butoi(robinet 1 deschis).

La accesarea butonului Reset se va reseta sistemul si va reveni la starea inițială

Bibliografie

1. www.citect.com/QuickStart_Tutorial
2. www.engineering.upm.ro/~traian/web_curs/Master/sacpi/cercetare/start.html
3. <http://en.wikipedia.org/wiki/SCADA>
4. <http://www.automatizari-scada.ro/>