

Aplicații SCADA - Conexiuni externe

Cuprins

Aplicații SCADA - Conexiuni externe	1
Obiective	1
Organizarea sarcinilor de lucru	1
1. Portul serial	2
Configurarea aplicației în vederea utilizării portului serial.	2
Programarea portului serial.	3
2. Utilizarea portului serial	5
Afișarea unui canal analogic	7
3. Utilizarea bazelor de date	8
Instalarea unei conexiuni MySQL.....	8
Data Base Exchange Control.....	11
Test de autoevaluare	12
Rezumat	13
Rezultate așteptate	14
Termeni esențiali.....	14
Recomandări bibliografice	15
Link-uri utile	15
Test de evaluare	16

Obiective

- Prezentarea portului serial și modul de utilizare a acestuia
- Prezentarea modului de conectare la o bază de date
- Conectare la o bază de date printr-o conexiune MySQL
- Conectare la o bază de date prin intermediul ODBC
- Setarea proprietăților pentru o conexiune de tip ODBC

Organizarea sarcinilor de lucru

- Parcurgeți cele trei capitole ale cursului.
- În cadrul fiecărui capitol urmăriți exemplele ilustrative și încercați să le realizați în medul de dezvoltare "Citect".
- Fixați principalele idei ale cursului, prezentate în rezumat.
- Completați testul de autoevaluare.
- Timpul de lucru pentru parcurgerea testului de autoevaluare este de 15 minute.

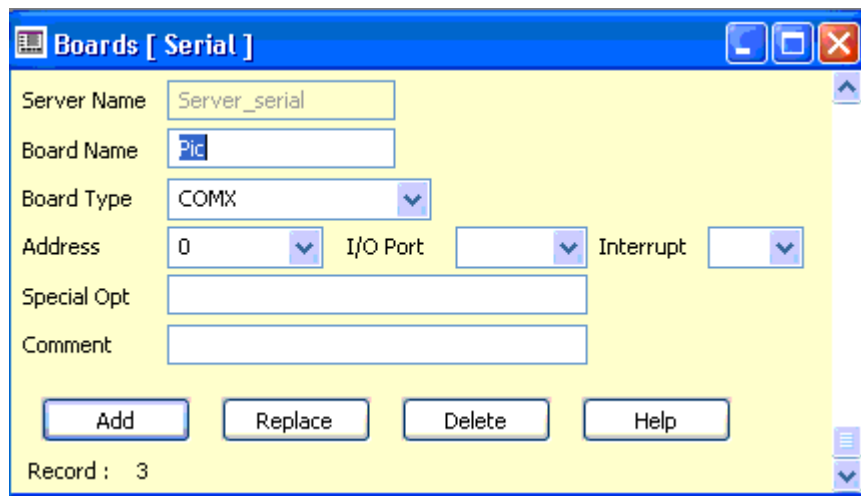
1. Portul serial

Numeroase sisteme de achiziție, comunică cu calculatorul prin intermediul portului serial. Mediul de dezvoltare aplicații SCADA Citect permite utilizarea portului serial pentru interfațarea cu diverse sisteme hardware de monitorizare și control. Utilizarea portului serial presupune scrierea unor funcții pentru conectarea și transferul datelor prin portul serial. De asemenea trebuie făcute niște setări în vederea utilizării portului serial.

Configurarea aplicației în vederea utilizării portului serial.

Se va configura o placă de achiziție (Board) astfel

- Din Citect Project Editor -> Communications -> Board , se adaugă :



Se configurează portul de comunicație astfel

- Din Citect Project Editor -> Communications -> Ports , se adaugă :

Programarea portului serial.

Pentru utilizarea portului serial trebuiesc scrise funcții pentru conectarea și transferul datelor prin portul serial.

Prima operație care se execută în vederea utilizării portului serial este operația de deschidere a portului serial. Aceasta operație se bazează pe utilizarea funcției **FUNCTION ComOpen(STRING nume_Port,0)**. Aceasta funcție are nevoie de un parametru de tip "string" pentru a preciza numele portului. Dacă portul cu numele "nume_Port", se poate deschide, funcția returnează un INT care reprezintă "handler-ul" cu care se vor face celelalte operații (citire, scriere, închidere).

Funcția pentru deschiderea portului serial

```

INT FUNCTION StartSerial(STRING sPort)
    INT    hPort;
    hPort = ComOpen(sPort, 0);
    IF hPort < 0 THEN
        Prompt("Nu s-a deschis portul: " + sPort);
        RETURN -1;
    END
    Prompt("Portul: " + sPort+ " a fost deschis.");
    hPrt=hPort;
    !TaskNew("SerialRead", hPrt, 0);
    !TaskNew("SerialWrite", hPrt, 0);
    !ComClose(hPort);
    buf="A0"
    sem=0;
    RETURN 1;
END

```

Funcția pentru închiderea portului serial

```
INT FUNCTION SerialClose(INT hPort)
    ComClose(hPort);
    Prompt("Portul a fost inchis.");
    RETURN 0;
END
```

Funcția pentru scrierea la portul serial

```
INT FUNCTION SerialWrite(INT hPort, STRING buffer)
    INT SerialWriteError;
    INT length;
    !WHILE 1 DO
        length=StrLength(buffer);
        SerialWriteError = ComWrite(hPort, buffer, length, 2);
        IF SerialWriteError THEN
            Prompt("Error Writing port");
            ComReset(hPort);
            RETURN -1;
        END
    !END
    RETURN 0;
END
```

Funcția pentru citirea la portul serial

```
INT FUNCTION SerialRead(INT hPort)
    STRING buffer
    INT length;
    INT total;
    INT SerialReadError;
    total = 0;
    !WHILE 1 DO
        length=20;
        SerialReadError = ComRead(hPort, buffer, length, 0);
        IF SerialReadError THEN
            Prompt("Eroare la citirea P S " + SerialReadError :
#####);
            ComReset(hPort);
            RETURN -1;
```

```

ELSE
    !Prompt(length : #### + " caractere citite de la P S
" );
    IF StrLength(buffer)>0 THEN
        buf_r=buffer;
        nr_r=StrLength(buffer);
        Val_a0=StrToInt(StrTrim(StrMid( buffer,3,StrLengt
h(buffer)-3)));
    END
END
!END
RETURN 0;
END

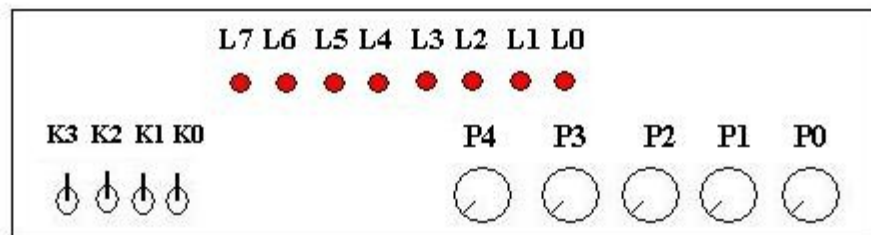
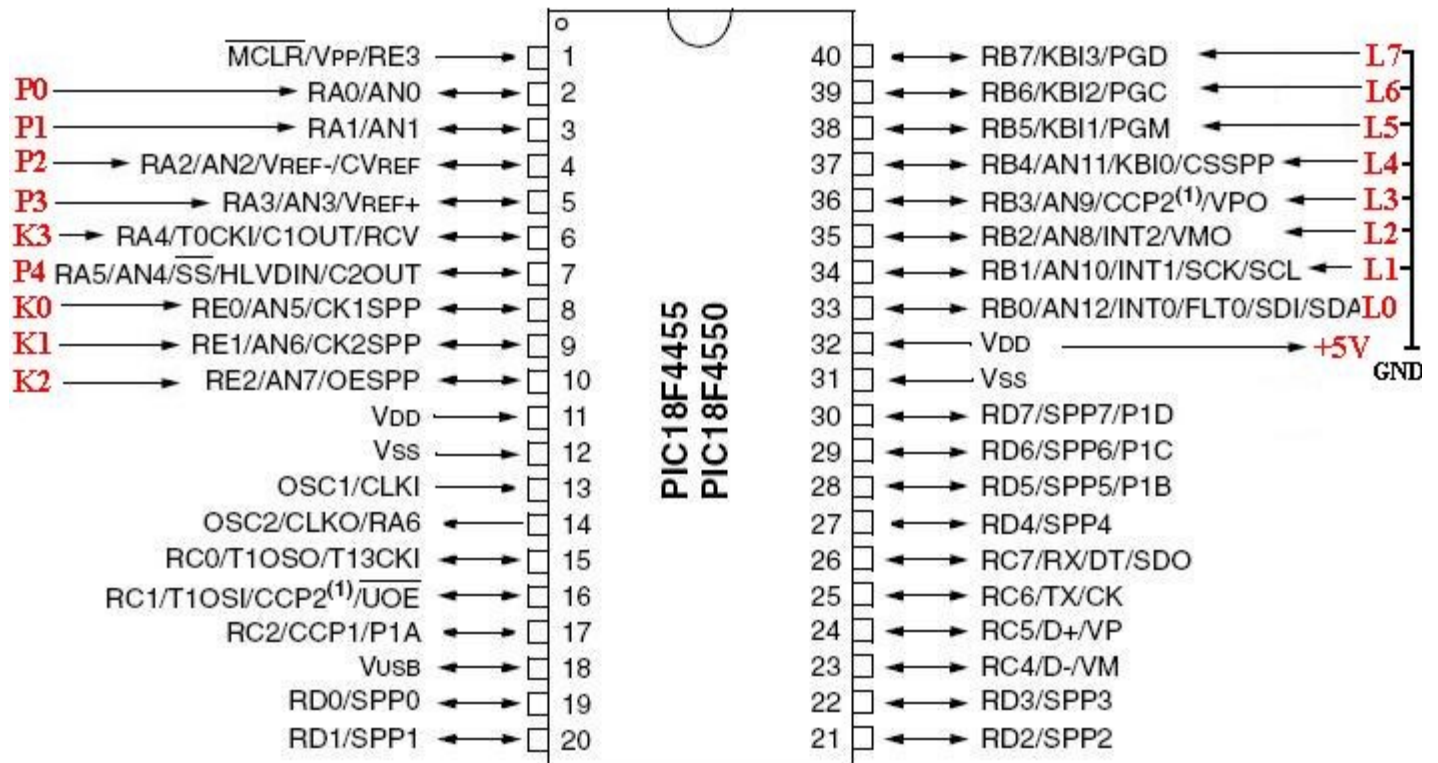
```

2. Utilizarea portului serial

Majoritatea aplicațiilor pentru achiziția și monitorizarea proceselor, utilizează sisteme de achiziție conectate la portul serial sau la alte porturi dar sunt accesate prin intermediul unui port serial virtual.

Un port virtual, emulează un port fizic. Astfel un sistem de achiziție care dispune de un port USB poate fi conectat la calculator prin intermediul portului USB. Pentru a utiliza conexiunea prin intermediul portului serial, se crează un port serial virtual din USB, astfel aplicația SCADA vede din punct de vedere logic, un port serial.

Să presupunem ca avem un sistem de achiziție date realizat pe baza unui sistem de dezvoltare PIC. Sistemul de achiziție dispune de 5 intrări analogice 4 digitale și 8 ieșiri digitale, conectate astfel:



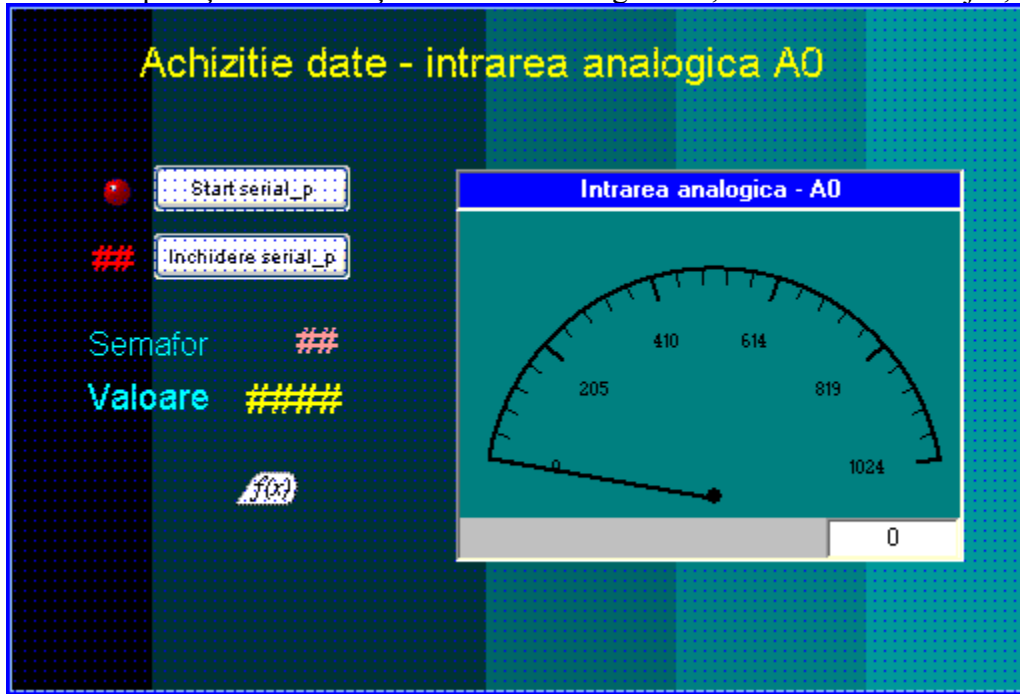
- Comanda pentru citirea simultana a celor cinci canale analogice este: **Ai**
- Comanda pentru citirea unei intrări analogice este: **Ai** unde $i=0..4$,
- Comanda pentru înscrierea celor 9 ieșiri digitale este **On** unde n este numărul în zecimal care va fi afișat în binar pe cele 8 ieșiri digitale și în sfârșit comanda D1 pentru citirea unui număr zecimal care codifica cele 4 intrări digitale.

Vom utiliza în continuare acest sistem de achiziție pentru a dezvolta diverse aplicații SCADA. Pentru început vom realiza aplicații simple care citesc intrări ale acestui sistem de achiziție.

Afișarea unui canal analogic

Să presupunem ca avem conectat un sistem de achiziție pe portul serial. Sistemul de achiziție dispune de intrări analogice. Pentru citirea intrării analogice 0 este necesara comanda "A0", iar sistemul răspunde cu valoarea analogică.

Pentru a realiza o aplicație care să afișeze intrarea analogica A0, similară cu cea de jos,



trebuie să folosim o funcție de genul:

```

INT FUNCTION Achiz
IF k1=1 THEN
  IF sem = 1 THEN
    sem=0;
    SerialRead(hPrt);
    SerialWrite(hPrt, "A0");
  ELSE
    sem=1;
    SerialRead(hPrt);
  END
END
RETURN 0
END

```

unde k este o variabilă care e setată dacă portul serial este deschis, iar „sem” este o variabilă care determină faza de citire sau faza de scriere.

3. Utilizarea bazelor de date

Aplicațiile SCADA "Citect" pot utiliza baze de date ca sursa de date în vederea transferului acestora spre și dinspre alte aplicații. În mod nativ „Citect” utilizează o serie de baze de date cum ar fi

- MS SQL Server 2000
- dBase DBF
- Access MDB
- Excel

Limbajul de programare "Cicode" inclus în mediul de dezvoltare „Citect” cuprinde funcții pentru implementarea limbajului SQL. Modul de utilizare al acestora se găsește în help-ul aplicației în ramura:

Cicode Programming Reference > Cicode Function Categories > SQL Functions Introduction > SQL Functions > SQLConnect

Se pot deci accesa baze de date care au implementat limbajul SQL, utilizând tehnologia ODBC.

Ne propunem în continuare interfațarea aplicațiilor "Citect" cu baza de date MySQL. MySQL este o bază de date SQL „open source” care ne va permite utilizarea acesteia pentru a transfera informații spre și dinspre alte aplicații chiar și pentru transferul datelor spre aplicații WEB.

Instalarea unei conexiuni MySQL

"Citect" nu utilizează în mod nativ baza de date MySQL, de aceea conexiunea la o baza de date MySQL se face prin utilizarea tehnologiei ODBC (Open Database Connectivity). ODBC furnizează un standard pentru metodele și procedurile software API (application programming interface) în vederea utilizării bazelor de date respectiv pentru utilizarea sistemului de gestiune al bazei de date numit DBMS (database management systems). Utilizarea ODBC asigură independența față de limbajul de programare, baza de date sau sistemul de operare.

Majoritatea producătorilor de baze de date oferă drivere pentru conexiuni ODBC. MySQL oferă un astfel de driver ODBC, care se poate descărca de pe site-ul:

<http://dev.mysql.com/downloads/connector/odbc/5.1.html>, găzduit de site-ul <http://www.mysql.com>.

De pe acest site se poate de altfel descărca baza de date MySQL de pe link-ul:

<http://dev.mysql.com/downloads>.

După descărcarea driver-ului **mysql-connector-odbc-5.1.5-win32.msi** utilizând link-ul:

<http://dev.mysql.com/downloads/connector/odbc/5.1.html>, se instalează prin rularea aplicației descărcate, după care se intra în :

Control Panel -> Other Control Panel Options -> DBE Administrator pentru a configura DBE (Data Base Engine) și pentru a adăuga o nouă conexiune, sau altfel spus un "Alias" pentru o nouă baza de date. Cu click dreapta -> New, putem alege un driver ODBC. Dacă în prealabil am instalat driverul odbc MySQL, acesta trebuie să se regăsească în lista. Alegem deci MySQL ODBC 5.1 Driver. și dăm numele "Alias-ului" să zicem "MySQL_citect". Click-dreapta pe numele alias creat->alegem

ODBC administrator -> Configure -> Putem să alegem Serverul bazei de date user-ul și baza de date.

Pentru Windows 7 instalarea unei conexiuni la o baza de date MySQL se face în mod similar, cu deosebirea ca se intră în: Control Panel -> System and Security -> Administrative Tools -> Data Sources (ODBC). Se alege din meniu System DSN și prin apăsarea butonului "Add" se adaugă o noua conexiune.

În cadrul unei aplicații "Citect" avem posibilitatea să interogăm baza de date utilizând aliasul: "MySQL_citect" creat anterior.

Să presupunem că am crea variabilele locale hSQL de tip INT și mes de tip STRING după care creăm funcția connect_mysql utilizând Citect Editor

Funcția care realizează conectarea la baza de date "MySQL_citect" arată astfel:

```
FUNCTION connect_mysql()  
    hSQL = SQLConnect("DSN=MySQL_citect");  
    IF hSQL > -1 THEN  
        mes="Conexiunea MySQL_citect realizata";  
    ELSE  
        mes="Eroare de conectoare MySQL_citect";  
    END  
END
```

Unde mes este o variabila locala de tip string folosita pentru a afișa diverse mesaje.

În cazul în care vrem să lansăm un dialog pentru conectarea la o anumită bază de date scriem

```
FUNCTION connect_mysql()  
    hSQL = SQLConnect("DLG=1");  
    IF hSQL > -1 THEN  
        mes="Conexiunea MySQL_citect realizata";  
    ELSE  
        mes="Eroare de conectoare MySQL_citect";  
    END  
END
```

Să presupunem că în cadrul bazei de date deschise, se află și tabela ap_electr, care conține câmpurile den_ap și tens_n. Citirea valorilor den_ap și tens_n se face astfel:

```
FUNCTION citesc_mysql()  
    err1=SQLExec (hSQL, "SELECT * FROM ap_electr ");  
    err2=SQLNext (hSQL);
```

```
        dn_ap = SQLGetField(hSQL, "den_ap");  
        u_n = SQLGetField(hSQL, "tens_n");  
END
```

err1 fiind un cod de eroare întors de funcția SQLExec iar err2 fiind un cod de eroare întors de funcția SQLNext, iar dn_ap și u_n sunt variabile locale care vor fi înscrise cu valorile citite din prima înregistrare aflată în baza de date.

Pentru a citi restul înregistrărilor utilizăm funcția de mai jos, funcție care ne permite să navigăm pe următoarele înregistrări.

```
FUNCTION next_mysql()  
err3=SQLNext(hSQL);  
    IF err_sel<>0 THEN  
        mes="Nu exista inregistrare";  
        u_n="0";  
    ELSE  
        mes = SQLGetField(hSQL, "den_ap");  
        u_n = SQLGetField(hSQL, "tens_n");  
    END  
END
```

Data Base Exchange Control

În cazul în care nu dorim să folosim datele citite în cadrul aplicației, dorim doar citirea și afișarea datelor, se poate folosi un obiect Data Base Exchange Control.

								
Filter	Download	Upload	Hide Cols	Show All	Print	Setting	Update	Refresh
Interfata cu baza de date								
	ap_id	den_ap	tens_n	crnt_n	def			
▶	1	TV-Color-Sony	110	10.5	1.27			
	3	Motor trifazat	380	16.75	2.55			
	4	TV-Color-Sony++	220	10.5	1.27			
	5	Bec fluorescent	220	1.3	1			
	6	Motor trifazat	380	16.75	2.55			
	7	Bec obisnuit	380	2.05	1			
	9	Sursa in comutatie	250	9	1			
	10	Aparat sudura	380	3	1.33			
	11	Calculator electronic	220	2.5	1			
	22	TV-Color-Sony++	380	10.5	1.27			
*								

Modul de setare proprietăți.

- Apperance->Connection->Build->Use Databasa->Source, aleg "MySQL_citect"
- Apperance->Connection->Table Name, aleg ap_electr
- Setez Radio Button Data Table

Test de autoevaluare

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

In vederea utilizării portului serial, ce configurări sunt necesare ?

- ☐ a. Configurarea plăcii de achiziție (Board)
- ☐ b. Configurarea tag-urilor trend
- ☐ c. Configurarea portului de comunicație
- ☐ d. Configurarea I/O Devices

2. Handler-ul corespunzător unui port serial se obține în urma operației de:

- ☐ a. Citire de la portul serial
- ☐ b. Deschiderea portului serial
- ☐ c. Scriere la portul serial
- ☐ d. Închiderea portului serial

3. Care operații au nevoie de handler ?

- ☐ a. Operația de citire de la portul serial
- ☐ b. Operația de scriere la portul serial
- ☐ c. Operația de închidere a portului serial
- ☐ d. Operația de deschidere a portului serial

4. Ce este un port virtual ?

- ☐ a. Este un port generic
- ☐ b. Este un port emulat
- ☐ c. Este un port fizic
- ☐ d. Este un port auxiliar

Funcția SQLConnect este utilizată pentru:

- ☐ a. Conectarea la orice bază de date
- ☐ b. Conectarea directă la bază de date MySQL
- ☐ c. Conectare la baza de date MySQL prin intermediul unei conexiuni ODBC
- ☐ d. Conectarea la serverul MySQL

5. **Grila de evaluare:** 1-a, c; 2-b; 3-a, b, c; 4-b; 5-c.

Rezumat

Numeroase sisteme de achiziție, comunică cu calculatorul prin intermediul portului serial. Mediul de dezvoltare aplicații SCADA „Citect” permite utilizarea portului serial pentru interfațarea cu diverse sisteme hardware de monitorizare și control. Utilizarea portului serial presupune scrierea unor funcții pentru conectarea și transferul datelor prin portul serial. De asemenea trebuie făcute niște setări în vederea utilizării portului serial.

Configurarea plăcii de achiziție (Board)

- Din Citect Project Editor -> Communications -> Board

Configurarea portului de comunicație

- Din Citect Project Editor -> Communications -> Ports

Programarea portului serial.

Pentru utilizarea portului serial se utilizează funcțiile .

- Funcția pentru deschiderea portului serial
- Funcția pentru scrierea la portul serial
- Funcția pentru citirea de la portul serial
- Funcția pentru închiderea portului serial

Utilizarea bazelor de date

Aplicațiile SCADA "Citect" pot utiliza baze de date ca sursa de date în vederea transferului acestora spre și dinspre alte aplicații. În mod nativ „Citect” utilizează o serie de baze de date cum ar fi

- MS SQL Server 2000
- dBase DBF
- Access MDB
- Excel

Limbajul de programare "Cicode" inclus în mediul de dezvoltare „Citect” cuprinde funcții pentru implementarea limbajului SQL. Modul de utilizare al acestora se găsește în help-ul aplicației în ramura:

Cicode Programming Reference > Cicode Function Categories > SQL Functions Introduction > SQL Functions > SQLConnect

Se pot deci accesa baze de date care au implementat limbajul SQL, utilizând tehnologia ODBC.

Instalarea unei conexiuni MySQL

"Citect" nu utilizează în mod nativ baza de date MySQL, de aceea conexiunea la o bază de date MySQL se face prin utilizarea tehnologiei ODBC (Open Database Connectivity). ODBC furnizează un standard pentru metodele și procedurile software API (application programming interface) în vederea utilizării bazelor de date respectiv pentru utilizarea sistemului de gestiune al bazei de date numit DBMS (database management systems). Utilizarea ODBC asigură independența față de limbajul de programare, baza de date sau sistemul de operare.

Rezultate așteptate

După studierea acestui modul, ar trebui să știți:

- Să utilizați portul serial pentru a accesa diverse dispozitive atașate la acesta
- Să scrieți aplicații SCADA care comunică prin portul serial cu diverse sisteme de achiziție
- Să vă conectați la diverse baze de date
- Să realizați aplicații SCADA în care să transferați date din și dinspre diverse baze de date.

Termeni esențiali

Termen	Descriere
SCADA	Supervisory Control And Data Acquisition
Tag	Nume generic pentru elementele din procesul monitorizat codificate prin intermediul variabilelor
HMI	Human Machine Interface -Interfața dintre aplicație și utilizator
Limbaj Cicode	Limbaj de programare inclus în mediul de dezvoltare Citect SCADA
ODBC	Open Database Connectivity
SQL	Structured Query Language
DBMS	Database Management Systems)
DBEC	Data Base Exchange Control

Recomandări bibliografice

- [1] Traian Turc, Elemente de programare C++ utile in ingineria electrica, Ed.Matrixrom, Bucuresti,2010
- [2] Traian Turc, Programare avansata C++ pentru ingineria electrica, Ed.Matrixrom, Bucuresti,2010
- [3] Traian Turc, Programarea in limbaje de asamblare, uz intern, Univ."Petru Maior",Tg.Mures,2009
- [4] Traian Tur,Brevet de inventie nr:11863 "Sistem pentru automatizarea si monitorizarea proceselor industriale", OSIM, 2003
- [5] Jeff Kent, C++ fara mistere,Ed.Rosetti Educational 2004 .
- [6] Boldur Barbat - Informatica industrială - Programarea în timp real – Institutul Central pentru Conducere si informatica 1984
- [7] Ioan Babuita – Conducerea automata a proceselor – Ed. Facla 1985
- [8] Ghercioiu-National în struments - Orizonturi în instrumentatie 1995
- [10] C. J. Date - An introduction to Database Systems - Addison-Wesley Publishing Company
- [11] Lungu,- Baze de date. Organizare, proiectare si implementare - Editura All Educational

Link-uri utile

- 1. <http://www.free-scada.org/> - Free SCADA - 2009.
- 2. <http://www.7t.dk/igss/default.asp> - IGSS SCADA System - 2009
- 3. <http://www.7t.dk/igss/default.asp?showid=374> - IGSS Online SCADA Training - 2009
- 4. <http://www.7t.dk/free-scada-software/index.html>- IGSS Free SCADA Software -2009
- 5. <http://www.citect.com/> - CITECT SCADA -2009
- 6. http://www.citect.com/index.php?option=com_content&view=article&id=1457&Itemid=1314 - Download CITECT demo - 2009
- 7. <http://www.indusoft.com/index.asp> - INDUSOFT SCADA - 2009
- 8 <http://www.gefanuc.com/products/2819> - Proficy HMI/SCADA - CIMPLICITY - 2009.
- 9. <http://www.genlogic.com/> - Dynamic Graphics, Data Visualization, Human-Machine Interface (HMI) - 2010
- 10 <http://www.genlogic.com/demos.html> - On-Line Java and AJAX Demos - 2010
- 11 <http://www.free-scada.org/> - - 2009
- 12 <http://www.free-scada.org/> - - 2009

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. *FUNCTION ComOpen este:*

- ☐ a. O funcție definită de utilizator
- ☐ b. O funcție predefinită
- ☐ c. O funcție special destinată utilizării portului serial
- ☐ d. O funcție generală pentru deschiderea unei componente

2. *Handlerul atașat unui port serial este:*

- ☐ a. Un string
- ☐ b. Un text
- ☐ c. Un întreg
- ☐ d. Denumirea portului serial

3. *Pentru un port serial virtual trebuie setați următorii parametri:*

- ☐ a. Baud rate
- ☐ b. Data Bits
- ☐ c. Parity
- ☐ d. Port Number

4. *Pentru a instala o conexiune ODBC avem nevoie de :*

- ☐ a. Un cablu de conexiune
- ☐ b. Un driver
- ☐ c. O funcție predefinită
- ☐ d. O aplicație specială

Un obiect DBEC (Data Base Exchange Control) se folosește la

- ☐ a. Realizarea unei conexiuni la o bază de date
- ☐ b. Vizualizarea și editarea datelor dintr-o bază de date
- ☐ c. Pentru realizarea unei conexiuni ODBC
- ☐ d. Pentru aplicații SCADA cu baze de date

Grila de evaluare: 1-a, c; 2-b; 3-c; 4-b; 5-b, d.