Functii pentru caractere, siruri C si clase, utilizare USB

Citirea caracerelor si a sirurilor de caractere

Majoritatea calculatoarelor au instalat cel putin un editor de text. Functiile referitoare la siruri de caractere, respectiv la texte au deci o importanta capitala in implementarea diferitelor aplicatii de procesare texte. Sirurile de caractere sunt realizate din "insiruirea" mai multor caractere.

Vom incerca sa utilizam cateva functii referitoare la sirurile de caractere, disponibile in limbajul C++. Limbajul de programare C++, dispune de tipul de date *string*, cunoscut si sub numele de clasa sir C++.

- **Citirea unui caracter de la tastatura**

In capitolele anterioare am realizat o serie de aplicatii in care se astepta introducerea unui caracter sau a unui text.

Am utilizat astfel, la sfarsitul unui program, urmatoarea secventa de mai jos pentru a intreba utilizatorul daca vrea sa continue aplicatia sau renunta:

```
// Citirea unui caracter de la tastatura
#include "stdafx.h"
#include < iostream >
using namespace std;

int main(void)
{
        system("TITLE Citirea unui caracter de la tastatura");// Titlul
ferestrei consola
        system("COLOR F9"); // Fundal alb caractere albastre
        char car;
        do{
                cout << "\n\n\tApasati tasta E pentru iesire sau C pentru
continuare :";
                cin >> car;
                if (car !='E' && car !='e')
                        cout << "\n\tAti ales -Continuare !";
                else
                        cout << "\n\tAti ales -Iesire";
        }while (car !='E' && car !='e');
    return 0;
}
```

Dupa cum se observa, continuarea programului se face prin apasarea oricarei taste. Numai la apasara tastei E sau e se face iesirea din program. In cazul in care se apasa tasta Enter sau spatiu,

nu se afiseaza nimic, functia cin << ignora caracterele spatiu si Enter de la inceput.

Pentru a lua in considerare si tasa Enter, vom folosi metoda **get** a obiectului **cin** . Metoda unui obiect se mai numeste si functia membru. In general, se utilizeaza expresia: "se invoca metoda **get** a obiectului **cin** .

```
// Citirea caracterului Enter
#include "stdafx.h"
#include < iostream >
using namespace std;

int main(void)
{
	system("TITLE Citirea caracterului 'Enter'");// Titlul ferestrei
consola
	system("COLOR F9"); // Fundal alb caractere albastre
	char car;
	do{
		cout << "\n\n\tApasati tasta E pentru iesire sau Enter pentru
continuare :";
		cin.get(car);
		if (car !='E' && car !='e')
			cout << "\n\tAti ales -Continuare !";
		else
			cout << "\n\tAti ales -Iesire";
	}while (car !='E' && car !='e');
    return 0;
}
```

In cazul de sus continuarea aplicatiei se face apasand Enter. In cazul in care se introduce un caracter, urmat de Enter, programul se repeta de doua ori, adica de numarul de caractere introdus. Pentru a corecta aceasta problema, va trebui sa stergem bufferul de caractere dupa executia instructiunii **cin.get(car);** prin invocarea metodei **ignore** a obiectului **cin**. Trebuie avut insa grija sa nu stergem bufferul daca in el se afla numai caracterul Enter. Vom conditiona deci executia instructiunii : **cin.ignore();** cu o instructiune conditionala de genul **if (car !='\n')**

```
// Citirea caracterului Enter
#include "stdafx.h"
#include < iostream >
using namespace std;

int main(void)
{
	system("TITLE Citirea caracterului 'Enter'");// Titlul ferestrei
consola
	system("COLOR F9"); // Fundal alb caractere albastre
	char car;
	do{
```

```
                cout << "\n\n\tApasati tasta E pentru iesire sau Enter pentru
continuare :";
                cin.get(car);
                if (car !='\n')
                cin.ignore();
                if (car !='E' && car !='e')
                        cout << "\n\tAti ales -Continuare !";
                else
                        cout << "\n\tAti ales -Iesire";
        }while (car !='E' && car !='e');
    return 0;
}
```

- **Citirea unui sir de caractere**

Din aplicatiile anterioare s-a observat ca utilizand instructiunile:
string nume;
cin >> nume;
nu putem citi numele si prenumele unei persoane pentru ca **cin** , va atribui variabilei nume, caracterele introduse pana la primul caracter spatiu, ignorand restul caracterelor. In cadrul tablourilor am rezolvat aceasta problema, invocand metoda **getline** a obiectului **cin** astfel:

```
// Citirea unui sir de caractere ce contine caractere "spatiu"
#include "stdafx.h"
#include < iostream >
using namespace std;

int main(void)
{
        system("TITLE Citirea unui sir de caracter de la tastatura");//
Titlul ferestrei consola
        system("COLOR F9"); // Fundal alb caractere albastre
        char car;
        char nume[80];
        do{
                cout << "\n\n\tIntroduceti numele d-voastra :";
                cin.getline(nume,80);
                cout << "\n\tNumele d-voastra este :" << nume ;
                cout << "\n\n\tApasati tasta E pentru iesire sau Enter pentru
continuare :";
                cin.get(car);
                if (car !='\n')
                cin.ignore();
        }while (car !='E' && car !='e');
    return 0;
}
```

Functii pentru caractere si siruri de caractere

In spatiul de nume std:: functiile pentru siruri sunt grupate in biblioteca "cstring" . Bibloteca "string" furnizeaza functii pentru calsa sir din C++.

- ## Functii pentru caractere

Exista situatii in care trebuie efectuam operatii asupra unui carater cum ar fi: sa testam valoarea unui caracter sau sa-l convertim din minuscula in majuscula etc. Exista functii special definite pentru operatii pe caractere.

Functii pentru conversia tipului de caracter

Sa reluam aplicatia de sus si sa folosim functia **touper** prin care sa convertim caracterul citit in litera mare astfel nu va mai fi necesara dubla testare (litera mare sau mica)

```
// Utilizrea functiei "toupper"
#include "stdafx.h"
#include < iostream >
using namespace std;

int main(void)
{
        system("TITLE Utilizrea functiei 'toupper'");// Titlul ferestrei
consola
        system("COLOR F9"); // Fundal alb caractere albastre
        char car;
        char nume[80];
        do{
                cout << "\n\n\tIntroduceti numele d-voastra :";
                cin.getline(nume,80);
                cout << "\n\tNumele d-voastra este :" << nume ;
                cout << "\n\n\tApasati tasta E pentru iesire sau Enter pentru
continuare :";
                cin.get(car);
                car=toupper(car);
                if (car !='\n')
                cin.ignore();
        }while (car !='E');
    return 0;
}
```

Functii pentru verificarea valorii unui caracter

De multe ori trebuie sa verificam valoarea unui caracter de exemplu: sa testam daca un caracter este o litera sau o cifra, sa testam o litera este majuscula sau minscula etc.

Vom realiza in continuare un program care cere o litera, o analizeaza si afiseaza tipul

caracterului.

```
//  Functii pentru verificarea valorii unui caracter
#include "stdafx.h"
#include < iostream >
using namespace std;

int main(void)
{
        system("TITLE Utilizrea functiilor pe caracter");// Titlul ferestrei
consola
        system("COLOR F9"); // Fundal alb caractere albastre
        char car;
        char car_t;
        do{
                cout << "\n\n\tIntroduceti un caracter :";
                cin.get(car_t);
                if (isalnum(car_t)){
                        if (isalpha(car_t)){
                                if (isupper(car_t))
                                        cout << "\n\n\tAti introdus un
caracter majuscula :";
                                else
                                        cout << "\n\n\tAti introdus un
caracter minuscula :";
                        }else{
                                cout << "\n\n\tAti introdus o cifra :";
                        }
                }else{
                cout << "\n\n\tNu ati introdus nici cifra nici caracter :";
                }
                cin.ignore();
                cout << "\n\n\tApasati tasta E pentru iesire sau Enter pentru
continuare :";
                cin.get(car);
                car=toupper(car);
                if (car !='\n')
                cin.ignore();
        }while (car !='E');
    return 0;
}
```

- ## Functii pentru siruri de caractere

Determinarea lungimii unui sir

Functia **strlen**

Forma generala: **strlen (nume_sir)**
Functia determina lungimea unui sir. Functia returneaza un numar întreg ce reprezinta

lungimea unui sir de caractere, fara a numara terminatorul de sir.

Urmatoarea aplicatie determina lungimea a trei siruri declarate in moduri diferite.

```
// Programul afiseaza lungimea unui sir de caractere
#include "stdafx.h"
#include < iostream >
#include < string >
using namespace std;
int main(void)
{
 system("TITLE Determinarea lungimii unui sir ");
 system("COLOR F9");
 cout << "\n\t- Programul afiseaza lungimea diverselor siruri de
caractere.\n\n\n";
 char* fc ="Facultatea de Inginerie";
 cout <<"\n\n\tLungimea sirului: " << fc << " este: " << strlen(fc);
 char ctd[55]="Inginerie Electrica";
 cout <<"\n\n\tLungimea sirului: " << ctd << " este: " << strlen(ctd);
 cout <<"\n\n\tLungimea sirului: " << "Sectia Calculatoare" << " este: " <<
strlen("Sectia Calculatoare");
 cin.get();
 return 0;
}
```

Functia **strlen** poate fi folosita de asemenea pentru a determina lungimea unui sir preluat de la utilizator.

```
// Programul afiseaza lungimea unui sir de caractere introdus de la
tastatura.
#include "stdafx.h"
#include < iostream >
#include < string >
using namespace std;
int main(void)
{
        system("TITLE Determinarea lungimii unui sir preluat ");
        system("COLOR F9");
        cout << "\n\t- Programul afiseaza lungimea unui sir preluat de la
tastatura.\n\n\n";
        char n_pr[80];
        cout << "\n\t- Introduceti numele d-voastra:";
        cin.getline(n_pr,80);
        cout <<"\n\n\tLungimea sirului: " << n_pr << " este: " <<
strlen(n_pr);
        cin.ignore();
        cin.get();
        return 0;
}
```

Sa combinam functiile pe caractere cu functiile pe siruri pentru a verifica daca un sir este introdus corect. Sa luam de exemplu o adresa de mail. Vom verifica daca are cel putin 5 caractere si contine caracterele . @.

```
// Programul cere adresa de mail si verifica daca:
// - adresa are cel putin 5 caractere
// - adresa contine caracterul .
#include "stdafx.h"
#include < iostream >
#include < string >
using namespace std;
int main(void)
{
        system("TITLE Validarea unui sir preluat ");
        system("COLOR F9");
        bool dim=false, at=false, punct=false;
        cout << "\n\tProgramul cere adresa de mail si verifica daca:.\n";
        cout << "\n\t - adresa are cel putin 5 caractere";
        cout << "\n\t - adresa contine caracterul @";
        cout << "\n\t - adresa contine caracterul .";
        char adresa[80];
        cout << "\n\n\n\t- Introduceti adresa de mail:";
        cin.getline(adresa,80);
        if (strlen(adresa) >5){
                dim=true;
        }
        for (int i=0; i <= strlen(adresa); i++){
                if (adresa[i]=='@')
                        at=true;
                if (adresa[i]=='.')
                        punct=true;
        }
        if(((at)&(punct)&(dim))==0){
                if (dim==0)
                        cout <<"\n\n\tLungimea adresei: " << adresa << " este
este mai mica de 5 caractere!";
                if (at==0)
                        cout <<"\n\n\tAdresa : "<< adresa << "nu contine @ ";
                if (punct==0)
                        cout <<"\n\n\tAdresa : "<< adresa << " nu contine .";
                }else{
                        cout <<"\n\n\tAdresa : "<< adresa << " este corecta
!";
                }
        cin.get();
        return 0;
}
```

Operatia efectuata anterior se numeste operatia de validare camp.

Copierea unui sir intr-un alt sir

Forma generala: **strcpy (sir_destinatie, sir_sursa)**

Functia copiaza sirul sursa în sirul destinatie. Pentru a fi posibila copierea, lungimea sirului destinatie trebuie sa fie mai mare sau egala cu cea a sirului sursa, altfel pot apare erori.

Sa luam o aplicatie de geniul:

```
//Copierea unui sir intr-un alt sir

#include "stdafx.h"
#include < iostream >
#include < string >
using namespace std;
int main(void)
{
        system("TITLE Copierea unui sir intr-un alt sir  ");
        system("COLOR F9");
        char* sir1="Alfa";
        char* sir2="Beta";
        sir1=sir2;
        cout << "\n\n\tSirul 1 este: " << sir1;
        cin.get();
        return 0;
}
```

Aplicatia nu cpiaza continutul sir 2 peste sir 1 ci atribuie pointerului sir1 valoarea pointerului sir2.

Pentru a demonstra acest lucru, vom mai introduce un pointer numit sir_1 care va indica tot spre sir1

```
//Copierea unui sir intr-un alt sir
#include "stdafx.h"
#include < iostream >
#include < string >
using namespace std;
char* sir_1;
int main(void){
        system("TITLE Copierea unui sir intr-un alt sir  ");
        system("COLOR F9");
        char* sir1="Alfa";
        char* sir2="Beta";
        sir_1=&sir1[0];// pointerul sir_1 indica spre sir1
        sir1=sir2;
        cout << "\n\n\tSirul 1 este: " << sir1;
        // daca in urma operatiei sir1=sir2 s-ar copia sir2 peste sir1,
instructiunea de jos
        // ar trebui sa afiseze tot "Beta" insa afiseaza "Alfa"
        cout << "\n\n\tSirul 1 este: " << sir_1;
```

```
        cin.get();
        return 0;
}
```

Pentru a copia efectiv sirul, vom folosi functia **strcpy** :

```
//Copierea unui sir intr-un alt sir
#include "stdafx.h"
#include < iostream >
#include < string >
using namespace std;
char* sir_1;
int main(void){
        system("TITLE Copierea unui sir intr-un alt sir  ");
        system("COLOR F9");
        char sir1[5]="Alfa";
        char sir2[5]="Beta";
        sir_1=&sir1[0];// pointerul sir_1 indica spre sir1
        strcpy(sir1,sir2);
        cout << "\n\n\tSirul 1 este: " << sir1;
        cout << "\n\n\tSirul 1 este: " << sir_1;
        cin.get();
        return 0;
}
```

De data aceasta se afiseaza acelasi lucru deci copierea s-a facut efectiv.
In cazul ca vrem sa copiem un numar limitat de caractere, folosim functia **strncpy** astfel:

```
//Copierea unui subsir intr-un alt sir
#include "stdafx.h"
#include < iostream >
#include < string >
using namespace std;
int main(void){
        system("TITLE Copierea unui sir intr-un alt sir  ");
        system("COLOR F9");
        char sir1[10]="Alfa-Soft";
        char sir2[5]="Beta";
        strncpy(sir1,sir2,3);
        cout << "\n\n\tSirul 1 este: " << sir1;
        cin.get();
        return 0;
}
```

Raspunsul aplicatiei va fi de data aceasta: "Beta-Soft"

Concatenarea a doua siruri

Forma generala: **strcat (sir_destinatie, sir_sursa)**

Operatia de adunare a doua siruri adica operatia de a copia un sir la sfarsitul altui sir se numeste concatenare.

Functia **strcat** concateneaza cele doua siruri: sirul sursa este adaugat la sfârsitul sirului destinatie. Tabloul care contine sirul destinatie trebuie sa aiba cel putin dimensiunea celor doua siruri.

Pentru a concatena doua siruri, procedam astfel:

```
//Adaugarea unui sir la un alt sir
#include "stdafx.h"
#include < iostream >
#include < string >
using namespace std;
int main(void){
        system("TITLE Adaugarea unui sir la un alt sir  ");
        system("COLOR F9");
        char sir1[15]="Alfa-Soft-";
        char sir2[5]="Beta";
        strcat(sir1,sir2);
        cout << "\n\n\tSirul 1 este: " << sir1;
        cin.get();
        return 0;
}
```

Raspunsul aplicatiei va fi de data aceasta: "Alfa-Soft-Beta". Trebuie sa avem grija ca dimensiunea sirului sir1 sa fie suficienta pentru a adauga sir2.

Compararea a doua siruri

Forma generala: **strcmp (sir_1, sir_2)**

Functia compara cele doua siruri date ca argument si returneaza o valoare întreaga egala diferenta dintre codurile ASCII ale primelor caractere care nu coincid.

Pentru a compara doua siruri, vom folosi functia **strcmp** . Functia returneaza 0 daca sirurile sunt egale, un numar negativ daca primul sir< al doilea sir respectiv pozitiv daca primul sir > al doilea sir.

```
//Compararea a doua siruri;
#include "stdafx.h"
#include < iostream >
#include < string >
using namespace std;
```

```
int main(void){
        system("TITLE Compararea a doua siruri  ");
        system("COLOR F9");
        char sir1[10]="Alfa";
        char sir2[5]="Alfa";
        if (strcmp(sir1,sir2)==0)
        cout << "\n\n\tCele doua siruri sunt egale !" ;
        else
        cout << "\n\n\tCele doua siruri sunt diferite !" ;
        cin.get();
        return 0;
}
```

Raspunsul este desigur "Cele doua siruri sunt egale". Daca am folosi conditia :"if (sir1==sir2) raspunsul ar fi tot timpul fals deoarece se compara defapt doua adrese ale celor doua siruri.

Conversia unui intreg intr-un sir

```
//  Conversia unui intreg intr-un sir
#include "stdafx.h"
#include < iostream >
using namespace std;

int main(void)
{
        system("TITLE Conversia unui intreg intr-un sir");
        system("COLOR F9");
        int n;
        char buffer [33];
        cout << "\n\n\tIntroduceti un numar intreg: " ;
        cin >> n;
        itoa(n,buffer,10);//functia pentru conversia unui intreg intr-un sir
        cout << "\n\n\tSirul rezultat din convertirea numarului este: " <<
buffer;
        cin.ignore();
        cin.get();
  return 0;
}
```

Pozitia unde apare subsirul s2 in sirul s1

Forma generala: **strstr(s1, s2)**

Functia returneaza pozitia unde apare subsirul s2 in sirul s1 sau altfel spus, functia returneaza un pointer la începutul subsirului s2, sau NULL daca subsirul nu este gasit.

Urmatoarea aplicatie foloseste functia *strstr(s1, s2)* pentru a gasi pozitia unde se gaseste un

subsir si afiseaza sirul initial incepand din aceasta pozitie.

```
// Programul cere un sir si un subsir de caractere .
// Programul afiseaza sirul din pozitia unde incepe subsirul in cadrul
sirului .
// Se foloseste functia
// Functia strstr(s1, s2) - returneaza pozitia unde apare subsirul s2 in
sirul s1
#include "stdafx.h"
#include < iostream >
#include < string >
using namespace std;

int main(void)
{
system("TITLE Pozitia unui subsir in cadrul unui sir ");
system("COLOR F9");
char s1[100],s2[25],*p;
cout << "\n\n\tIntroduceti sirul = ";
cin.getline(s1,80);
cout << "\n\tIntroduceti subsirul = ";
cin.getline(s2,80);
p=strstr(s1,s2);
if (p!=NULL)
cout << "\n\n\n\tSirul este: = " << p;
else
cout << "\n\n\n\tSubsirul:  " << s2 << " nu s-a gasit";
cin.ignore();
cin.get();
return 0;
}
```

Conversia unui sir intr-un intreg

Pentru a converti un text ce reprezinta un intreg folosim functia **atoi**

```
//  Conversia unui sir intr-un intreg
#include "stdafx.h"
#include < iostream >
using namespace std;

int main(void)
{
        system("TITLE  Conversia unui sir intr-un intreg ");
        system("COLOR F9");
        char nr_ascii [10]="123";
        int nr=atoi(nr_ascii);
        cout << "\n\n\tValoarea zecimala a sirului este: " << nr ;
```

```
        cout << "\n\n\tPatratul valoarii zecimale a sirului este: " << nr*nr;
        cin.ignore();
        cin.get();
  return 0;
}
```

Siruri de caracere in spatiul System

Diverse conversii

In spatiul de nume System se foloseste System::Convert::... pentru diverse conversii.

```
// Conversia unui sir intr-o valoare numerica in System
// Functia ReadLine citeste tot timpul un sir de caractere
// Se foloseste System::Convert::... pentru diverse conversii
#include "stdafx.h"
#include < iostream >
using namespace std;
using namespace System;

int main(void)
{
        std::system("TITLE Conversia unui sir intr-o valoare numerica");
        std::system("COLOR F9");
        double raza; // valoarea numerica a razei
        String^ raza_s; // sirul ce contine valoarea citita
        double pi= System::Math::PI;
        Console::Write( L"\n\n\tIntroduceti raza cercului:" );
        raza_s= Console::ReadLine();
        raza = System::Convert::ToDouble( raza_s );
        Console::WriteLine( "\n\n\tAria cercului de raza: "+ raza +" este:
"+pi*raza*raza );
        Console::ReadLine();
    return 0;
}
```

La fel procedam si pemtru aplicatii CLR Windows Form Applications.

Pentru a realiza o aplicatie care permite sa zicem introducerea unei valori numerice care reprezinta un unghi in radiani dupa care afiseaza sinusul acelei valori, avem nevoie de o functie ce converteste valoarea in Double, pentru a calcula sinusul apoi avem nevoie de conversia valorii sinusului intr-un text in vederea afisarii intr-un obiect de tip label.

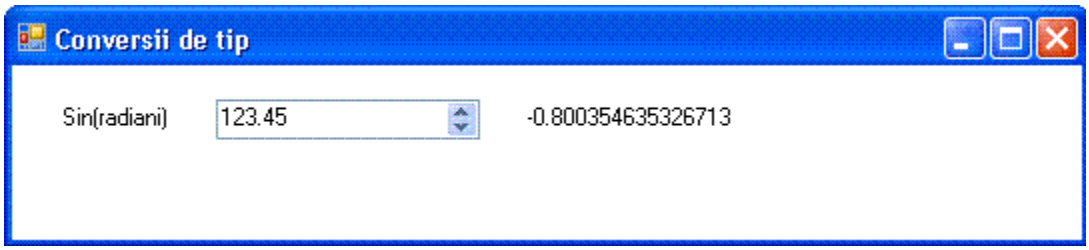
Deschidem un nou proiect Windows Forms Application intitulat "convers_01" pe care plasam un obiect de tip numericUpDown numit numericUpDown1 si doua obiecte de tip label numite label1,label 2.

Completam procedura deschisa pe evenimentul "ValueChanged" cu:

```
        double rad2;
        rad2=System::Convert::ToDouble(this->numericUpDown1->Value);
        this->label2->Text
=System::Convert::ToString(System::Math::Sin(rad2));
```

Metoda: **System::Convert::ToDouble(this->numericUpDown1->Value)**- converteste valoarea introdusa in Double iar **System::Convert::ToString(System::Math::Sin(rad2))** converteste vaoarea double in text.

Dupa rularea aplicatiei, tastarea unei valori, apoi tasta Enter, se obtine:



Obiectul numericUpDown poate fi inlocuit cu un obiect de tipul textBox numit textBox1. Avantajul este ca dupa fiecare tastare a unei cifre se afiseaza o noua valoare a sinusului. Dezavantajul este ca daca se tasteaza alte caractere inafara cifrelor, programul da o eroare de exceptie care trebuie tratata. In acest caz procedura deschisa pe evenimentul "ValueChanged" a obiectului textBox1 devine:

```
                double rad2;
                try {
                        rad2=System::Convert::ToDouble(this->textBox1->Text
);
                        this->label2->Text
=System::Convert::ToString(System::Math::Sin(rad2));
                }
                catch(System::FormatException^){
                                this->label2->Text="Eroare format";
                }
```

Metode pentru siruri utilizate in WFA

Vom utiliza in continuare cateva metode referitoare la siruri de caractere. Vom realiza o aplicatie WFA care afiseaza in mod continuu numarul de caractere introduse de la tastatura. In cadrul acestei aplicatii vom utiliza un obiect label care ne va permite sa afisam numarul de caractere introdus de la tastatura prin intermediul unui obiect de tip TextBox. Vom utiliza un

obiect ProgressBar care ne va permite sa afisam grafic procentul de caractere introdus. Afisarea se va face procentual relativ la numarul maxim de 300 de caractere setat pentru aceasta aplicatie. Se va afisa de asemenea si procentul sub forma numerica, atat pe 15 caractere cat si pe maxim 5 caractere.

Vor fi folosite printre altele metodele :

- **Length** -pentru determinarea lungimii unui sir de caractere
- **Trim** -pentru eliminarea spatiilor din cadrul unui sir de caractere
- **System::String::Substring** - pentru a substrage un subsir dintr-un sir de caractere
- **System::Convert::ToString** - pentru a converti diverse tipuri in string

Deschidem un nou proiect Windows Forms Application intitulat "text_bar" pe care plasam:

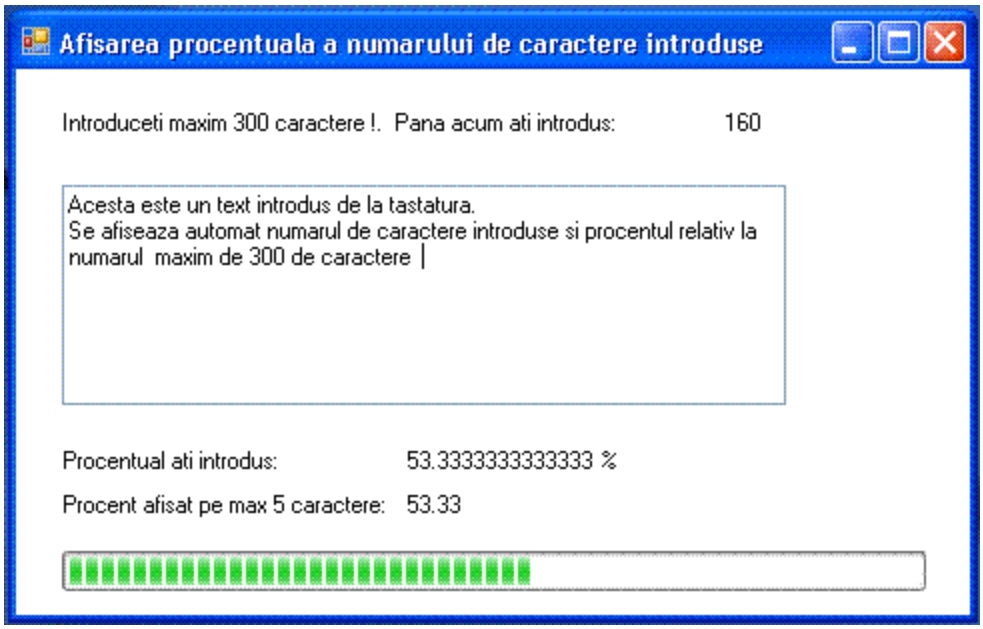
- 7 obiecte de tip **label**
- un obiect de tip **textBox** caruia ii setam proprietatea "multiline" la "True".
- un obiect de tip **progressBar**
- un obiect de tip **timer** caruia ii setam proprietatea "enabled" la "True" si interval la 100.

Completam procedura deschisa pe evenimentul Tick al obiectului timer1 cu :

```
double nr_car=(this->textBox1->Text)->Length;
this->label1->Text=System::Convert::ToString(nr_car);
double proc=System::Convert::ToDouble(100*nr_car/300);
String^ proc_s=System::Convert::ToString(proc);
this->label2->Text=proc_s;
String^ proc_st=proc_s->Trim();
String^ proc_s5;
if (proc_st->Length > 5)
        proc_s5=proc_s->System::String::Substring(0,5);
else
        proc_s5=proc_s;

if (proc>100)
        proc=100;
this->label3->Text=proc_s5;
this->progressBar1->Value =proc;
```

Rulam aplicatia si obtinem:



Vehicularea sirurilor de caractere prin intermediul portului serial

Portul serial a fost utilizat pentru a realiza prima legatura intre doua calculatoare. Portul serial este des utilizat si in prezent pentru a conecta diverse dispozitive la calculator. Chiar daca majoritatea calculatoarelor nu mai dispun fizic de un port serial, se utilizeaza des porturi seriale virtuale, realizate prin intermediul portului USB (Universal Serial Bus). Cu alte cuvinte exista o serie de dispozitive conectate la calculator prin intermediul USB insa din punct de vedere logic ele sunt conectate printr-un port serial virtual. Pentru programator, aceasta conectare este identica cu o conectare prin intermediul unui port serial fizic. Conectarea prin intermediul portului USB fiind deci "transparenta" din punctul de vedere al scrierii aplicatiilor, acestea fiind scrise la fel cu aplicatiile pentru portul serial.

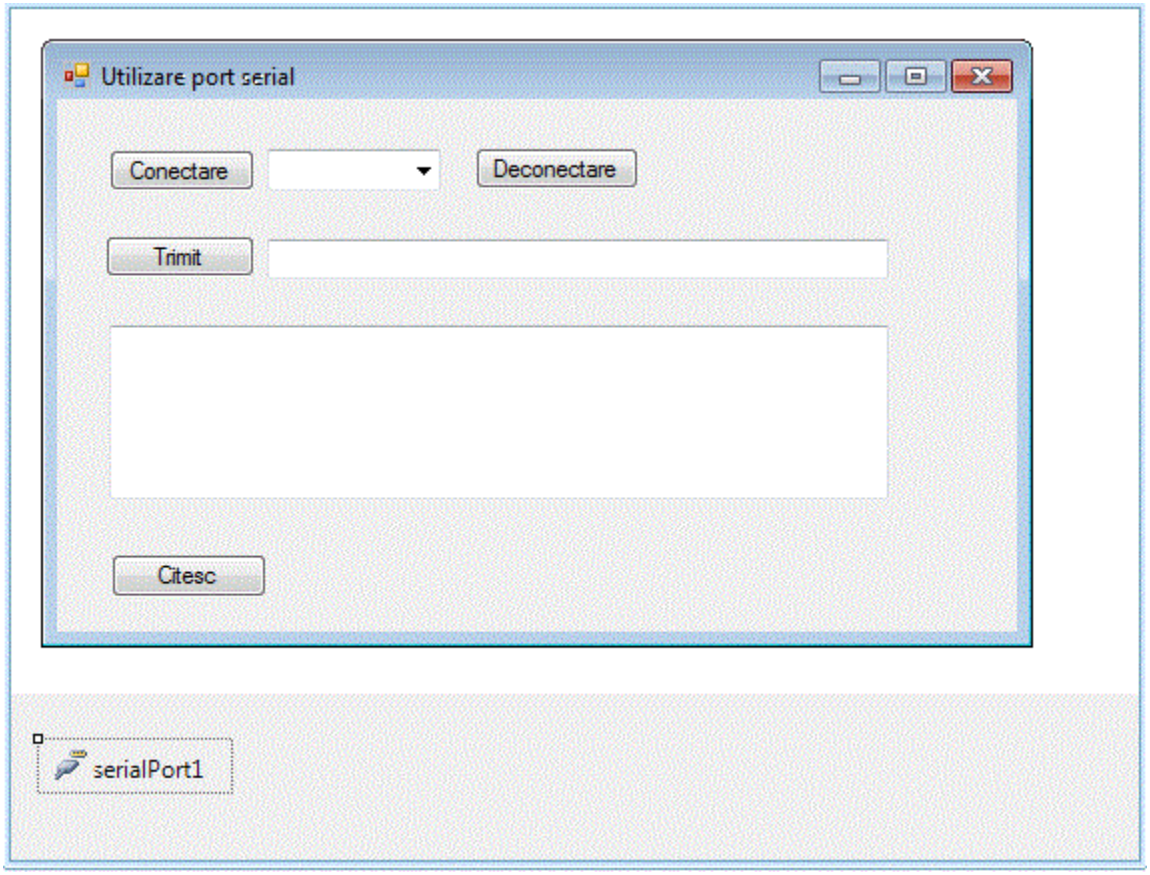
- **Configurarea si initializarea portului serial**

Pentru a putea fi utilizat, portul serial trebuie configurat, cu alte cuvinte trebuiesc stabilite valorile pentru diversi parametri, cum ar fi: viteza de lucru, lungimea cuvantului, paritate etc.

In vederea utilizarii portului serial, deschidem un nou proiect Windows Forms Application intitulat "Rs_232_v0" . In cadrul acestei aplicatii, e nevoie de o interfata care sa permita selectarea portului serial, deschiderea si inchiderea acestuia, stabilirea unui sir de caractere pentru a fi trimis prin portul serial, trimiterea acestuia, citirea unui sir de caractere si afisarea sirului citit. Vom plasa deci

- un obiect de tip **ListBox** cu numele "Porturi_s" pentru alegerea portului serial

- 4 obiecte de tip **button** 2 pentru conectarea si deconectarea la portul serial(cu numele "but_con" respectiv "but_dec"), iar 2 pentru trimiterea si citirea unui sir de caractere(cu numele "but_tx" respectiv "but_cit").
- un obiect de tip **textBox** cu numele "tx_tx" pentru introducerea sirului ce urmeaza a fi transmis.
- un obiect de tip **textBox** cu numele "tx_rx" pentru afisarea siruliui receptionat. acestui obiect ii setam proprietatea "multiline" la "True".
- un obiect de tip **SerialPort**



Obiectul de tip "ListBox" este folosit pentru selectarea unui port serial. La un moment dat pot fi deschise mai multe porturi seriale, deci trebuie sa dispunem de o metoda prin care sa selectam portul serial dorit.

Obiectul de tip "ListBox" este cel mai potrivit pentru a selecta unul din porturile seriale cu conditia ca in momentul lansarii aplicatiei sa fie insctrise elementele listei cu numele porturilor deschise in acel moment. Pentru aceasta, pe evenimentul "Load" al form-ului Form1 vom completa procedura deschisa, cu:

```
                        int i;
                        array < System::String^,1 >^ Nume_porturi;
                        Nume_porturi = serialPort1->GetPortNames();
                        this->Porturi_s->Items->Clear();

                        //Adaug porturile exixtente

                        for(i=0;i < Nume_porturi->Length;i++)
                        {
                                this->Porturi_s->Items->Add(Nume_porturi[i]);
                        }

                        //Pozitionarea listei pe primul element
                        this->Porturi_s->SelectedIndex = 0;
```

Pentru conectarea la portul selectat, folosim butonul "but_con" pe a carui eveniment "Click", punem:

```
// Se incearca deschiderea portului selectat
try
{
 //Preluarea numelui de port selectat.
 this->serialPort1->PortName=System::Convert::ToString(this->Porturi_s-
>Items[this->Porturi_s->SelectedIndex]);
 //Deschiderea portului serial .
 this->serialPort1->Open();
 //Schimbarea starii butoanelor
 this->but_con->Enabled = false;
 this->Porturi_s->Enabled = false;
 this->but_dec->Enabled = true;
 this->but_st->Enabled = true;
 this->tx_tx->Clear();
 this->tx_tx->AppendText("AA");
 //Stergerea textului si afisarea textului de conectare.
 this->tx_rx->Clear();
 this->tx_rx->AppendText("Portul serial conectat.\r\n");
}
catch(...)
{
 // In caz de erori, se inchide portul
 but_dec_Click(this, gcnew EventArgs());
}
```

Pentru deconectarea de la portul conectat, folosim butonul "but_dec" pe a carui eveniment "Click", punem:

```
```

```
                        //Refacerea starii initiale a butoanelor
                        this->but_dec->Enabled = false;
                        this->but_con->Enabled = true;
                        this->Porturi_s->Enabled = true;

                        try
                        {
                                //dezafectarea bufferelor utilizate;
                                this->serialPort1->DiscardInBuffer();
                                this->serialPort1->DiscardOutBuffer();
                                this->tx_rx->Clear();
                                this->tx_rx->AppendText("Portul serial a fost
deconectat.\r\n");
                                //Inchiderea portului serial
                                this->serialPort1->Close();
                        }
                        catch(...){}
```

Pentru trimiterea sirului din textBox-ul "tx_tx", folosim butonul "but_tx" pe a carui eveniment "Click", punem:

```
                         //Se trimite portului selectat textul din tx_tx.
                        try

                        {
                                //Write the data in the text box to the open
serial port

                                this->serialPort1->Write(tx_tx->Text);
                        }

                        catch(...)
                        {
                                // In caz de erori
                                // Se inchide portul
                                but_dec_Click(this, gcnew EventArgs());
                        }
```

Obiectul "SerialPort1" declanseaza eveniventul "DataReceived" in momentul cand portul serial recepioneaza caractere.

Vom colpleta deci procedura deschisa pe evenimentul "DataReceived" cu:

```
 tx=this->serialPort1->ReadExisting();
```

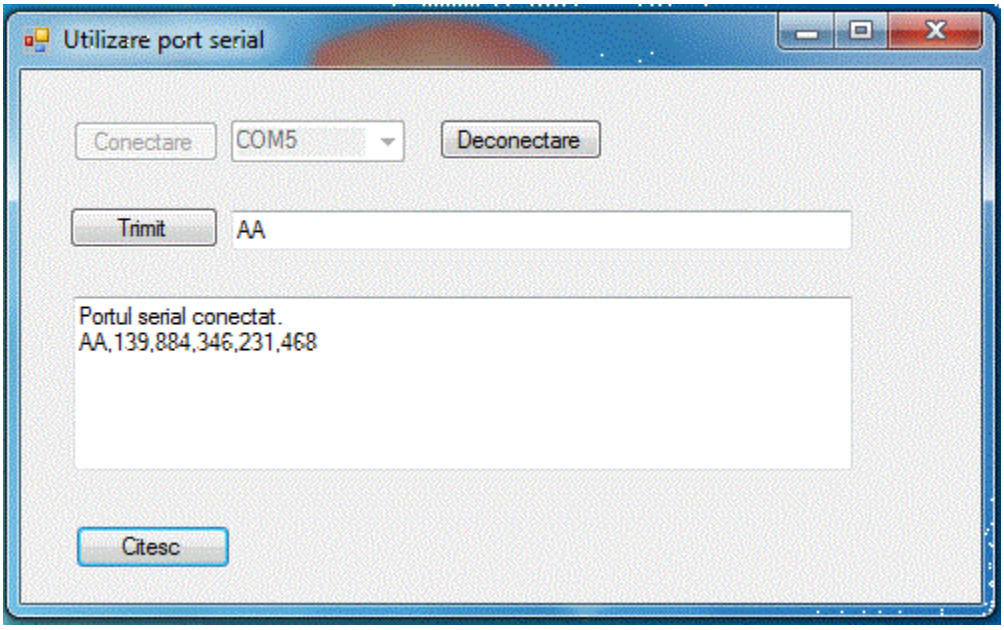
Unde tx este declarat in # pragma region astfel:

```
static System::String^ tx;
```

Pentru citirea sirului receprionat de la portul serial de catre obiectul **SerialPort** , folosim butonul "but_rx" pe a carui eveniment "Click", punem:

```
this->tx_rx->AppendText(tx);
```

Sa presupune ca avem conectat pe portul serial 5 (COM5) un sistem de achizitie date care la comanda "AA" raspunde cu valoarea celor cinci parameri analogici achizitionati. Rulam aplicatia si dupa alegerea portulii serial "COM5" , Conectare , Trimitere text "AA" si "Citesc", obtinem:



- **Configurarea si initializarea portului serial- C#**

C# - selectarea portului dorit

```
            int i, j;
            // Listez porturile seriale
            Nume_porturi = System.IO.Ports.SerialPort.GetPortNames();
            this.Porturi_s.Items.Clear();
```

```
            //Adaug porturile exixtente

            for (i = 0; i < Nume_porturi.Length; i++)
            {
                this.Porturi_s.Items.Add(Nume_porturi[i]);

            }

            //Pozitionarea listei pe primul element
            this.Porturi_s.SelectedIndex = 0;
```

C# - Conectarea la portul dorit

```
            if (!this.serialPort1.IsOpen)
            {
                this.serialPort1.PortName =
System.Convert.ToString(this.Porturi_s.Items[this.Porturi_s.SelectedIndex]);
                this.serialPort1.Open();
                this.label1.Text = "Portul a fost deschis";
            }
            else
            {
                this.label1.Text = "Portul este deja deschis deschis";
            }
            this.textBox1.Text = "AA";
            this.serialPort1.Write("AA");
```

C# - Deconectarea de la portul serial

```
            this.serialPort1.Close();
            this.label1.Text = "Portul este deconectat!";
```

C# - scrierea si citirea portului serial

```
            cda = this.textBox1.Text;

            if (this.serialPort1.IsOpen)
            {
               this.serialPort1.Write(cda);
               txt = "";
               txt = this.serialPort1.ReadExisting();
```

```
                if (txt.Length > 0)
                {
                   this.textBox2.Text = txt;
                }
                else
                {
                   this.label1.Text = "Nu vin date!";
                }
            }
```

C# - aplicatia - RS_232_v01

```
namespace RS_232_v01
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        string txt, cda;
        int k;
        static String[] Nume_porturi = new String[11];
        private void Form1_Load(object sender, EventArgs e)
        {
            int i, j;
            // Listez porturile seriale
            Nume_porturi = System.IO.Ports.SerialPort.GetPortNames();
            this.Porturi_s.Items.Clear();

            //Adaug porturile exixtente

            for (i = 0; i < Nume_porturi.Length; i++)
            {
                this.Porturi_s.Items.Add(Nume_porturi[i]);
            }

            //Pozitionarea listei pe primul element
            this.Porturi_s.SelectedIndex = 0;
            this.label1.Text = "Conectati-va la portul serial!";

        }

        private void but_con_Click(object sender, EventArgs e)
        {
            if (!this.serialPort1.IsOpen)
            {
                this.serialPort1.PortName =
System.Convert.ToString(this.Porturi_s.Items[this.Porturi_s.SelectedIndex]);
                this.serialPort1.Open();
                this.label1.Text = "Portul a fost deschis";
```

```
            }
            else
            {
                this.label1.Text = "Portul este deja deschis deschis";
            }
            this.textBox1.Text = "AA";
            this.serialPort1.Write("AA");
        }

        private void but_dec_Click(object sender, EventArgs e)
        {
            this.serialPort1.Close();
            this.label1.Text = "Portul este deconectat!";

        }

        private void timer1_Tick(object sender, EventArgs e)
        {
            k++;
            if (k > 123456789)
                k = 0;
            this.label2.Text = k.ToString();

            cda = this.textBox1.Text;

            if (this.serialPort1.IsOpen)
            {
               this.serialPort1.Write(cda);
               txt = "";
               txt = this.serialPort1.ReadExisting();
               if (txt.Length > 0)
               {
                  this.textBox2.Text = txt;
               }
               else
               {
                  this.label1.Text = "Nu vin date!";
               }
            }
        }
    }
}
```

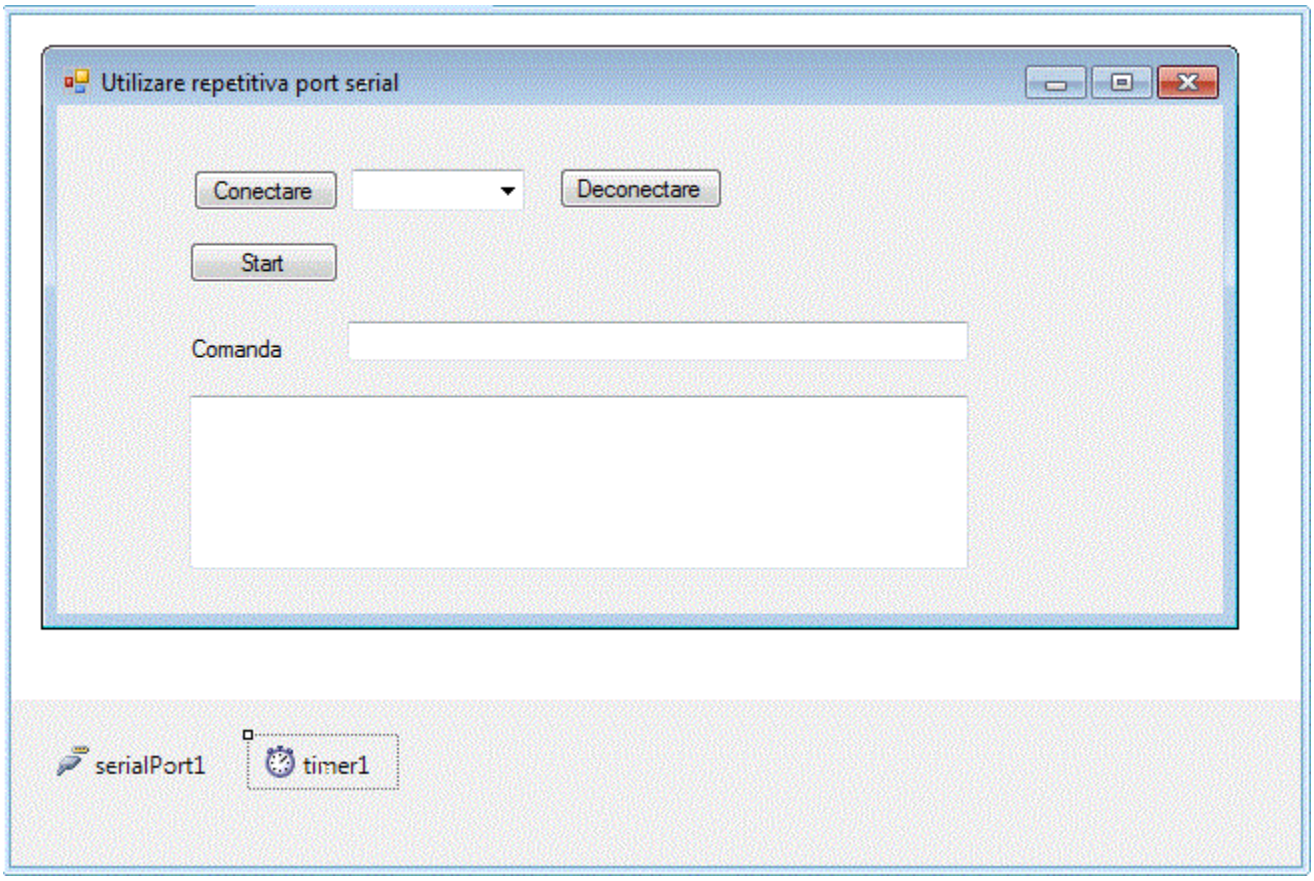
- **Utilizarea portului serial**

Pentru a obtine un nou set de date trebuie sa reluam operatiile anterioare. Am putea realiza o noua aplicatie in care sa folosim un "Timer" iar dupa operatia de conectare sa se rea automat restul operatiilor astfel afisarea valorilor citite sa se faca in mod continuu.

Vom crea deci un nou proiect Windows Forms Application intitulat "Rs_232_v1" . Nu mai avem nevoie deci de butoanele "Trimit" si "Citesc" ci doar de un buton "Start".

Plasam deci:

- un obiect de tip **ListBox** cu numele "Porturi_s" pentru alegerea portului serial
- 3 obiecte de tip **Button** 2 pentru conectarea si deconectarea la portul serial(cu numele "but_con" respectiv "but_dec"), iar unul pentru declansarea trimiterii si citirii unui sir de caractere repetitiv (cu numele "but_st").
- un obiect de tip **TextBox** cu numele "tx_tx" pentru introducerea sirului ce urmeaza a fi transmis.
- un obiect de tip **TextBox** cu numele "tx_rx" pentru afisarea siruliui receptionat. acestui obiect ii setam proprietatea "multiline" la "True".
- un obiect de tip **SerialPort** cu numele "serialPort1"
- un obiect de tip **Timer** cu numele "timer1"



Obiectul de tip "ListBox" este folosit pentru selectarea unui port serial. La un moment dat pot fi deschise mai multe porturi seriale, deci trebuie sa dispunem de o metoda prin care sa selectam portul serial dorit.

Obiectul de tip "ListBox" este cel mai potrivit pentru a selecta unul din porturile seriale cu conditia ca in momentul lansarii aplicatiei sa fie insctrise elementele listei cu numele porturilor deschise in acel moment. Pentru aceasta, pe evenimentul "Load" al form-ului Form1 vom completa procedura deschisa, cu:

```
                        int i;
                        array < System::String^,1 >^ Nume_porturi;
                        Nume_porturi = serialPort1->GetPortNames();
                        this->Porturi_s->Items->Clear();

                        //Adaug porturile exixtente

                        for(i=0;i < Nume_porturi->Length;i++)
                        {
                                this->Porturi_s->Items->Add(Nume_porturi[i]);
                        }

                        //Pozitionarea listei pe primul element
                        this->Porturi_s->SelectedIndex = 0;
```

Pentru conectarea la portul selectat, folosim butonul "but_con" pe a carui eveniment "Click", punem:

```
// Se incearca deschiderea portului selectat
try
{
 //Preluarea numelui de port selectat.
 this->serialPort1->PortName=System::Convert::ToString(this->Porturi_s-
>Items[this->Porturi_s->SelectedIndex]);
 //Deschiderea portului serial .
 this->serialPort1->Open();
 //Schimbarea starii butoanelor
 this->but_con->Enabled = false;
 this->Porturi_s->Enabled = false;
 this->but_dec->Enabled = true;
 this->but_st->Enabled = true;
 this->tx_tx->Clear();
 this->tx_tx->AppendText("AA");
 //Stergerea textului si afisarea textului de conectare.
 this->tx_rx->Clear();
 this->tx_rx->AppendText("Portul serial conectat.\r\n");
}
catch(...)
{
 // In caz de erori, se inchide portul
 but_dec_Click(this, gcnew EventArgs());
}
```

Pentru deconectarea de la portul conectat, folosim butonul "but_dec" pe a carui eveniment "Click", punem:

```
                        //Refacerea starii initiale a butoanelor
                        this->but_dec->Enabled = false;
                        this->but_con->Enabled = true;
                        this->Porturi_s->Enabled = true;

                        try
                        {
                                //dezafectarea bufferelor utilizate;
                                this->serialPort1->DiscardInBuffer();
                                this->serialPort1->DiscardOutBuffer();
                                this->tx_rx->Clear();
                                this->tx_rx->AppendText("Portul serial a fost
deconectat.\r\n");
                                //Inchiderea portului serial
                                this->serialPort1->Close();
                        }
                        catch(...){}
```

Pentru declansarea trimiterii sirului din textBox-ul "tx_tx", si citirii repetitive folosim butonul "but_st" pe a carui eveniment "Click", punem validarea timerului "timer1":

```
this->timer1->Enabled=true;
```

Citirea si scrierea se foce deci pe evenimentul "Tick" al obiectuli "timer1"

```
                if (sem==0){
                        //Se trimite portului selectat textul din tx_tx.
                        try

                        {
                                //Write the data in the text box to the open
serial port

                                this->serialPort1->Write(tx_tx->Text);
                        }

                        catch(...)
                        {
                                // In caz de erori, se inchide portul

                                but_dec_Click(this, gcnew EventArgs());
                        }
                        sem=1;
```

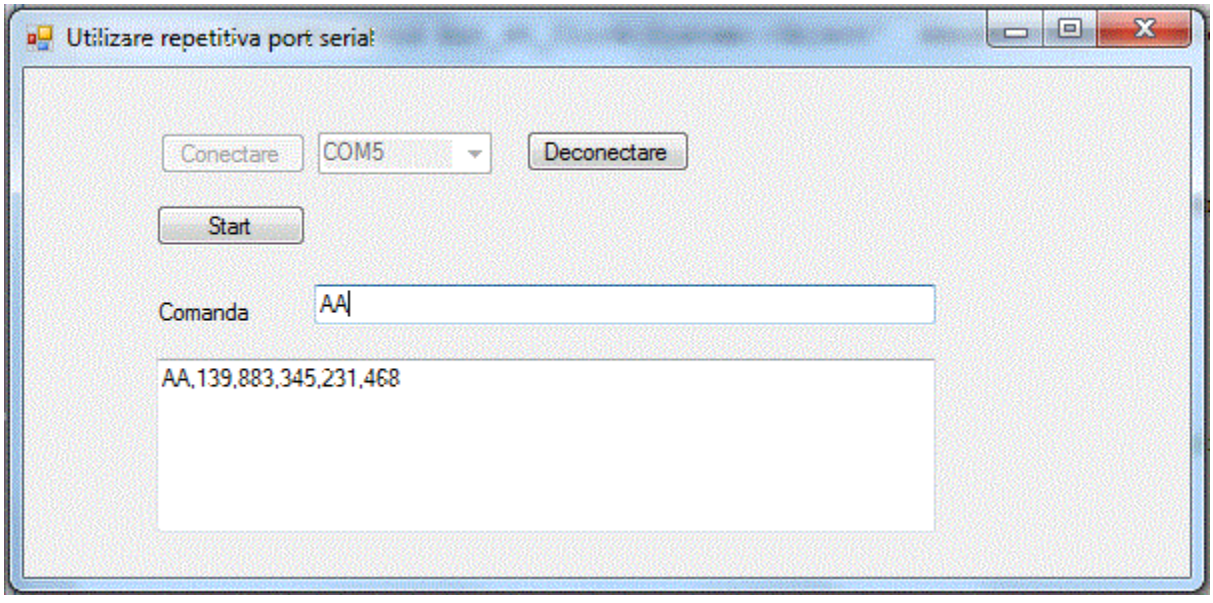
```
                }
                if (sem==1){
                        this->tx_rx->Clear();
                        this->tx_rx->AppendText(this->serialPort1-
>ReadExisting());
                        sem=0;
                }
```

Sem este o variabila statica definita in zona #pragma region astfel:

```
static int sem=0;
```

Variabila **sem** a fost folosita pe post de semafor, astfel la prima trecere se face scriere, la urmatoare citire, la urmatoare trecere scriere etc.

Sa presupune ca avem conectat pe portul serial 5 (COM5) un sistem de achizitie date care la comanda "AA" raspunde cu valoarea celor cinci parameri analogici achizitionati. Rulam aplicatia si dupa alegerea portului serial "COM5" apoi Start , obtinem:



Pentru a separa din cadrul sirului receptionat, fiecare parametru in parte, trebuie sa utilizam instructiuni pe siruri de caractere. Vom crea deci o noua aplicatie "Rs_232_v2" in care fiecare parametru transmis va fi afisat separat folosind in plus 5 obiecte de tip "Label". Nu mai aven nevoie de obiectul de tip **TextBox** pentru introducerea sirului ce urmeaza a fi transmis deoarece vom folosi numai comanda "AA" cu care citim simultan cei cinci paramerii.

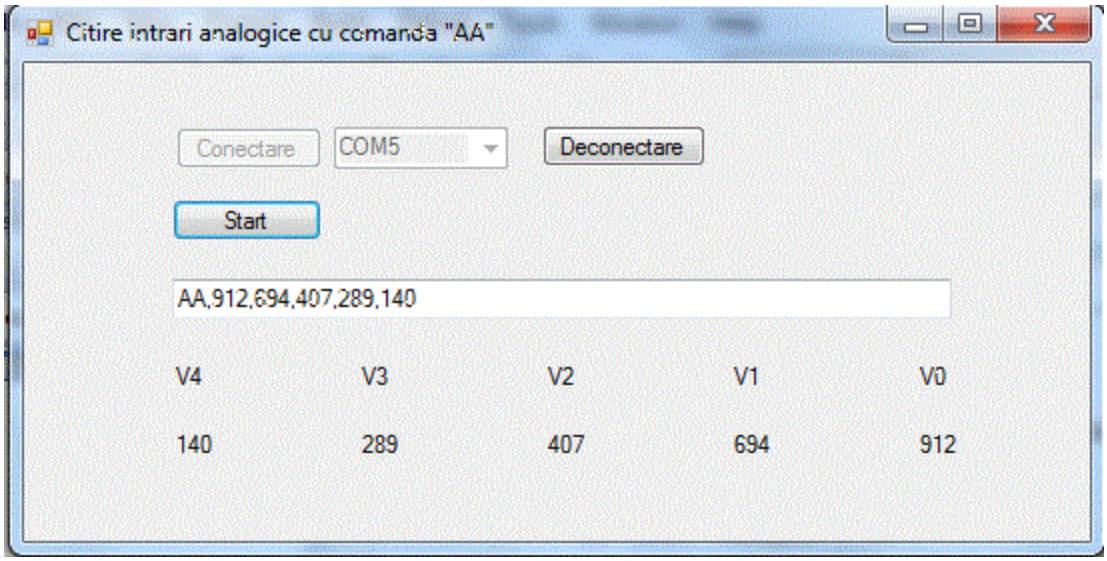
Aplicatia este similara cu aplicatia anterioara cu diferenta ca pe evenimentul "tick" vom plasa:

```
if (sem==0){
        //Se trimite portului selectat textul din tx_tx.
        try
        {
                //Write the data in the text box to the open serial port

                this->serialPort1->Write("AA");
        }
        catch(...)
        {
                // In caz de erori, se inchide portul
                but_dec_Click(this, gcnew EventArgs());
        }
sem=1;
}
else{
        String^ rec=this->serialPort1->ReadExisting();
        this->tx_rx->Clear();
        this->tx_rx->AppendText(rec);
        if (rec->Length > 0){
                int p1=rec->IndexOf(",");
                int p2=rec->IndexOf(",",p1+1);
                int p3=rec->IndexOf(",",p2+1);
                int p4=rec->IndexOf(",",p3+1);
                int p5=rec->IndexOf(",",p4+1);
                int p6=rec->Length;
                this->label11->Text=rec->Substring(p1+1,p2-p1-1);
                this->label10->Text=rec->Substring(p2+1,p3-p2-1);
                this->label9->Text=rec->Substring(p3+1,p4-p3-1);
                this->label8->Text=rec->Substring(p4+1,p5-p4-1);
                this->label7->Text=rec->Substring(p5+1,p6-p5-1);
        }
        else{
                this->label11->Text=" ";
                this->label11->Text=" ";
                this->label10->Text=" ";
                this->label9->Text=" ";
                this->label8->Text=" ";
                this->label7->Text=" ";
        }
sem=0;
}
```

Se observa ca p1-p5 sunt pozitiile in care se gaseste separatorul "," iar p6 este pozitia in care se termina sirul, respectiv lungimea acestuia. Pozitia este determinata folosind metoda "IndexOf". Pentru a extrage parametrul curent, stiindu-se pozitia celor doi separatori "," adica inceputul si sfarsitul subsirului ce reprezinta parametrul respectiv, s-a folosit metoda "Substring".

Rulam aplicatia si obtinem:



Vom imbunatatii aplicatia realizand aplicatia "Rs_232_v3" in care vom reprezenta grafic parametrii tranmisi. Spre deosebire de aplicatia precedenta vom avea nevoie de o serie de variabile statice plasate in #pragma region:

```
static int sem=0;
static System::Drawing::Graphics^ Desen;
static System::Drawing::Pen^ Creion_rosu ;
static System::Drawing::Pen^ Creion_albastru;
static System::Drawing::Pen^ Creion_pic;

static float w_r=2,w_a=5;
static array < double,1 >^ val_a;
static String^ v1;
static String^ v2;
static String^ v3;
static String^ v4;
static String^ v5;
```

Pe evenimentul "Load" punem:

```
               Desen = this->CreateGraphics();
               Creion_rosu=gcnew
System::Drawing::Pen(System::Drawing::Color::Red,w_r);
               Creion_albastru=gcnew
System::Drawing::Pen(System::Drawing::Color::Blue,w_a);
               Creion_pic=gcnew
System::Drawing::Pen(System::Drawing::Color(this->BackColor),w_a);
```

```
                Desen->Clear(System::Drawing::Color(this->BackColor));
                val_a = gcnew array < double,1 > (5);
                int i;
                array < System::String^,1 >^ Nume_porturi;
                Nume_porturi = serialPort1->GetPortNames();
                this->Porturi_s->Items->Clear();

                //Adaug porturile exixtente

                for(i=0;i < Nume_porturi->Length;i++)
                {
                        this->Porturi_s->Items->Add(Nume_porturi[i]);
                }
                //Pozitionarea listei pe primul element
                        this->Porturi_s->SelectedIndex = 0;
```

In cadrul evenimentului tick vom plasa:

```
if (sem==0){
        //Se trimite portului selectat textul din tx_tx.
        try
        {
                //Write the data in the text box to the open serial port

                this->serialPort1->Write("AA");
        }
        catch(...)
        {
                // In caz de erori, se inchide portul
                but_dec_Click(this, gcnew EventArgs());
        }
sem=1;
}
if (sem==1){
        String^ rec=this->serialPort1->ReadExisting();
        this->tx_rx->Clear();
        this->tx_rx->AppendText(rec);
        if (rec->Length > 0){
        int p1=rec->IndexOf(",");
        int p2=rec->IndexOf(",",p1+1);
        int p3=rec->IndexOf(",",p2+1);
        int p4=rec->IndexOf(",",p3+1);
        int p5=rec->IndexOf(",",p4+1);
        int p6=rec->Length;
        v1=rec->Substring(p1+1,p2-p1-1);
        this->label11->Text=v1;
        val_a[0]=System::Convert::ToDouble(v1);
        v2=rec->Substring(p2+1,p3-p2-1);
        this->label10->Text=v2;
        val_a[1]=System::Convert::ToDouble(v2);
        v3=rec->Substring(p3+1,p4-p3-1);
        this->label9->Text=v3;
        val_a[2]=System::Convert::ToDouble(v3);
        v4=rec->Substring(p4+1,p5-p4-1);
```

```
        this->label8->Text=v4;
        val_a[3]=System::Convert::ToDouble(v4);
        v5=rec->Substring(p5+1,p6-p5-1);
        val_a[4]=System::Convert::ToDouble(v5);
        this->label7->Text=v5;
        Desen->DrawLine( Creion_rosu,this->Width-130,this->Height-20,this-
>Width-130,10);
        Desen->DrawLine( Creion_rosu,this->Width-135,this->Height-43,this-
>Width,this->Height-43);
        for ( int i=0; i < 5; i++){
                Desen->DrawLine( Creion_pic,this->Width-30-i*20,this->Height-
49,this->Width-30-i*20,0);
                Desen->DrawLine( Creion_albastru,Width-30-i*20,this->Height-
47,
                this->Width-30-i*20,(this->Height-50)*(1-val_a[i]/1023));
                }
        }
        else{
                for ( int i=0; ilabel11->Text=" ";
                this->label10->Text=" ";
                this->label9->Text=" ";
                this->label8->Text=" ";
                this->label7->Text=" ";
        }

sem=0;
}
```

Rulam aplicatia si obtinem:

Pentru a extrage valorile parametrilor din cadrul sirului transmis, s-a folosit metoda **IndexOf** pentru a afla pozitiile succesive ale caracterului despartitor ",". Mult mai eleganta este folosirea metodei "Split".

Urmatoarea aplicatie "Rs_232_v4" foloseste aceasta metoda in procedura deschisa pe evenimentul Tick.

```
if (sem==0){
        //Se trimite portului selectat textul "AA".
        try
        {
                //Write the data in the text box to the open serial port

                this->serialPort1->Write("AA");
        }
        catch(...)
        {
                // In caz de erori, se inchide portul
                but_dec_Click(this, gcnew EventArgs());
        }
sem=1;
}
if (sem==1){
 String^ rec=this->serialPort1->ReadExisting();
 this->tx_rx->Clear();
 this->tx_rx->AppendText(rec);
date_c=rec->Split(delimiter);
 if (date_c[0]=="AA") {
        double
v0=System::Math::Round(System::Convert::ToDouble(date_c[5]),2);
        val_a[0]=v0;
        double
v1=System::Math::Round(System::Convert::ToDouble(date_c[4]),2);
        val_a[1]=v1;
        double
v2=System::Math::Round(System::Convert::ToDouble(date_c[3]),2);
        val_a[2]=v2;
        double
v3=System::Math::Round(System::Convert::ToDouble(date_c[2]),2);
        val_a[3]=v3;
        double
v4=System::Math::Round(System::Convert::ToDouble(date_c[1]),2);
        val_a[4]=v4;
        this->label11->Text=System::Convert::ToString(v0);
        this->label10->Text=System::Convert::ToString(v1);
        this->label9->Text=System::Convert::ToString(v2);
        this->label8->Text=System::Convert::ToString(v3);
        this->label7->Text=System::Convert::ToString(v4);
        Desen->DrawLine( Creion_rosu,this->Width-130,this->Height-20,this-
>Width-130,10);
        Desen->DrawLine( Creion_rosu,this->Width-135,this->Height-43,this-
>Width,this->Height-43);
        for ( int i=0; i < 5; i++){
        Desen->DrawLine( Creion_pic,this->Width-30-i*20,this->Height-49,this-
>Width-30-i*20,0);
        Desen->DrawLine( Creion_albastru,Width-30-i*20,this->Height-47,this-
>Width-30-i*20,
        (this->Height-50)*(1-val_a[i]/1023));
 }
}
```

```
else{
                for ( int i=0; i < 5; i++){
                        val_a[i]=0;
                }
                this->label11->Text=" ";
                this->label10->Text=" ";
                this->label9->Text=" ";
                this->label8->Text=" ";
                this->label7->Text=" ";
        }

sem=0;
}
```

Dupa cum se observa au fost definite noi variabile in zona #pragma region pentru a putea utiliza metoda "Split"

```
static int sem=0;
static System::Drawing::Graphics^ Desen;
static System::Drawing::Pen^ Creion_rosu ;
static System::Drawing::Pen^ Creion_albastru;
static System::Drawing::Pen^ Creion_pic;
static float w_r=2,w_a=5;
static array < double,1 >^ val_a;
static String^ v1;
static String^ v2;
static String^ v3;
static String^ v4;
static String^ v5;
static String^ delimStr = " ,.:\t";
static array^ delimiter = delimStr->ToCharArray( );
static array< String^,1>^ date_c = gcnew array < String^ > (7);
```

- **Utilizarea portului serial - C#**

```
namespace Multiio_v05
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        public System.Drawing.Graphics desen;
        public System.Drawing.Pen creion_blu;
        public System.Drawing.Pen creion_rosu;
        public System.Drawing.Pen creion_gri;
```

```
        public System.Drawing.SolidBrush pens_blu;
        public System.Drawing.SolidBrush pens_red;
        public System.Drawing.SolidBrush pens_back;
        public System.Drawing.SolidBrush radiera;
        public System.Drawing.Font font_nina;
        public binar binar1;
        public osciloscop grafic0;
        public osciloscop grafic1;
        public termo termo1;
        public termo termo2;
        public termo termo3;
        public termo termo4;
        public termo termo5;
        int a0_pozx = 350, a0_pozy = 50, a0_n_maxx = 300, a0_n_maxy = 200;
        Int32 a0_val, a0_val_max = 500, k, suma;
        int a1_pozx = 700, a1_pozy = 50, a1_n_maxx = 300, a1_n_maxy = 200;
        Int32 a1_val, a1_val_max = 1000;
        Int32 a2_val, a3_val, a4_val, a5_val;
        UInt64 digi;
        string txt, cda;
        int c0, c1, c2, c3, c4, c5, c6, c7, val_cmd, val_cmd_v;
        static int[] a0_valori = new int[0];
        static int[] a1_valori = new int[0];
        static String[] date_r = new String[11]; // Date receptionate
        static String[] date_c = new String[11]; // Date check - verificate
daca sunt ok
        static String delimStr = " ,.:\t";
        static Char[] delimiter = delimStr.ToCharArray();
        static String[] Nume_porturi = new String[11];

        // --------------   Osciloscop ------------------

        public class osciloscop
        {
            int x0;
            int y0;
            int w;
            int h;
            int val_max,val_max_af, val, val_v;
            int nr_max;
            System.Drawing.Graphics zona_des;
            System.Drawing.Pen creion_r = new
System.Drawing.Pen(System.Drawing.Color.Red);
            System.Drawing.Font font_ni = new System.Drawing.Font("Nina",
8);
            System.Drawing.SolidBrush pens_blu = new
System.Drawing.SolidBrush(System.Drawing.Color.Blue);
            System.Drawing.SolidBrush radiera = new
System.Drawing.SolidBrush(System.Drawing.Color.White);

            System.Drawing.Bitmap img;
            System.Drawing.Bitmap ims;

            public void setval(int[] vals, int nrv)
            {
                img = new Bitmap(nr_max, val_max, zona_des);
```

```
                int val, i, j;

                // afisare grafic sub forma de puncte

                val_v =
System.Convert.ToInt16(System.Convert.ToDouble(vals[0]) *
(System.Convert.ToDouble(h) / System.Convert.ToDouble(val_max))); //scalare
                for (i = 0; i < w; i++)
                {

                    val =
System.Convert.ToInt16(System.Convert.ToDouble(vals[i]) *
(System.Convert.ToDouble(h) / System.Convert.ToDouble(val_max))); //scalare
                    if (val_v < val)
                    {
                        for (j = val_v; j <= val; j++)
                            img.SetPixel(i, j, System.Drawing.Color.Red);
                    }
                    else
                    {
                        for (j = val; j <= val_v; j++)
                            img.SetPixel(i, j, System.Drawing.Color.Red);

                    }
                    val_v = val;
                }
                zona_des.DrawImage(ims, x0, y0);
                zona_des.DrawImage(img, x0, y0);

                //zona_des.FillRectangle(radiera, x0, y0 + h, w + 20, 20);
                for (i = 0; i <= w; i += 50)
                {
                    val = System.Convert.ToInt16(System.Convert.ToDouble(i)
* (System.Convert.ToDouble(nr_max) / System.Convert.ToDouble(w))); //scalare
                    zona_des.DrawString(val.ToString(), font_ni, pens_blu,
x0 + i, y0 + h);
                }
                //zona_des.FillRectangle(radiera, x0 - 20, y0 - 10, 20, h +
20);
                for (i = 0; i <= h; i += 50)
                {
                    val = System.Convert.ToInt16(System.Convert.ToDouble(i)
* (System.Convert.ToDouble(val_max_af) / System.Convert.ToDouble(h)));
//scalare
                    zona_des.DrawString(val.ToString(), font_ni, pens_blu,
x0 - 20, y0 + h - i - 10);
                }

            }
            public osciloscop(System.Drawing.Graphics desen, int pozx, int
pozy, int n_maxx, int n_maxy, int vmaxa)
            {
                x0 = pozx;
                y0 = pozy;
                w = n_maxx;
                h = n_maxy;
```

```
                nr_max = n_maxx;
                val_max = n_maxy;
                val_max_af = vmaxa;
                zona_des = desen;
                int i, j;
                img = new Bitmap(nr_max, n_maxy, zona_des);
                ims = new Bitmap(nr_max, n_maxy, zona_des);
                // sterg imaginea

                for (j = 0; j < val_max; j++)
                {
                    for (i = 0; i < nr_max; i++)
                    {
                        ims.SetPixel(i, j, System.Drawing.Color.WhiteSmoke);
                    }
                }
                // grid
                for (j = 0; j < val_max; j++)
                {

                    // grid orizontal


                    if (j % 10 == 0)
                    {
                        for (i = 0; i < nr_max; i++)
                        {
                            if (j % 50 == 0)
                                ims.SetPixel(i, j,
System.Drawing.Color.Gray);
                            else
                                ims.SetPixel(i, j,
System.Drawing.Color.LightGray);
                        }
                    }
                    else
                    {

                        // grid orizontal vertical

                        for (i = 0; i < nr_max; i++)
                        {
                            if (i % 10 == 0)
                            {
                                if (i % 50 == 0)
                                    ims.SetPixel(i, j,
System.Drawing.Color.Gray);
                                else
                                    ims.SetPixel(i, j,
System.Drawing.Color.LightGray);
                            }
                        }
                    }
                }

                //chenar
```

```
                for (i = 0; i < n_maxx; i++)
                {
                    ims.SetPixel(i, 0, System.Drawing.Color.Blue);
                    ims.SetPixel(i, val_max - 1, System.Drawing.Color.Blue);
                }
                for (j = 0; j < val_max; j++)
                {
                    ims.SetPixel(0, j, System.Drawing.Color.Blue);
                    ims.SetPixel(nr_max - 1, j, System.Drawing.Color.Blue);
                }

            }

        }

        // -------------- Clasa instrument ------------------
        public class binar
        {
            int x0;
            int y0;
            int w;
            int h;
            public void setval(int nrb, UInt64 n, System.Drawing.Graphics
zona_des, System.Drawing.Pen creion, System.Drawing.SolidBrush
pens_albastra, System.Drawing.SolidBrush radiera)
            {

                int wb = w / (3 * nrb);
                int hb = h / 3;
                int x = x0 + w - 3 * wb;
                int y = y0 + hb;
                int i;
                //zona_des.DrawRectangle(creion, x0, y0, w, h);
                for (i = nrb - 1; i >= 0; i--)
                {
                    System.UInt64 bit = ((n >> (nrb - i - 1)) & 1);
                    zona_des.DrawRectangle(creion, x - 1, y - 1, wb + 1, hb
+ 1);
                    if (bit == 1)
                        zona_des.FillRectangle(pens_albastra, x, y, wb, hb);
                    else
                        zona_des.FillRectangle(radiera, x, y, wb, hb);

                    x -= 3 * wb;

                }
            }
            public void init_binar(int pozx, int pozy, int lat, int inalt)
            {
                x0 = pozx;
                y0 = pozy;
                w = lat;
                h = inalt;
            }
        }
```

```
        public class termo
        {
            int x0;
            int y0;
            int w;
            int h;
            int val_max;
            public void desenez(System.Drawing.Graphics zona_des,
System.Drawing.Pen creion_a, System.Drawing.Pen creion_gr,
System.Drawing.SolidBrush pens_r, System.Drawing.Font font_ni)
            {
                zona_des.DrawRectangle(creion_a, x0, y0, w, h);
                for (int j = 0; j <= h; j += 5)// desenez gradatii
                {
                    if (j % 25 == 0)
                    {
                        zona_des.DrawLine(creion_gr, x0 + w + 2, y0 + j, x0
+ w + 12, y0 + j);
                        zona_des.DrawString(System.Convert.ToString(val_max
- j * val_max / h), font_ni, pens_r, x0 + w + 20, y0 + j - 7);
                    }
                    else
                    {
                        zona_des.DrawLine(creion_gr, x0 + w + 2, y0 + j, x0
+ w + 7, y0 + j);
                    }
                }



            }
            public void sterg(System.Drawing.Graphics zona_des,
System.Drawing.SolidBrush rad)
            {
                zona_des.FillRectangle(rad, x0 + 1, y0 + 1, w - 1, h - 1);
            }

            public void setval(float val, System.Drawing.Graphics zona_des,
System.Drawing.SolidBrush pens_r, System.Drawing.SolidBrush pens_b)
            {
                val = System.Convert.ToInt16(System.Convert.ToDouble(val) *
(System.Convert.ToDouble(h) / System.Convert.ToDouble(val_max))); //scalare
                zona_des.FillRectangle(pens_b, x0 + 1, y0 + 1, w - 1, h -
1);
                zona_des.FillRectangle(pens_r, x0 + 1, y0 + h - val, w - 1,
val);

            }
            public termo(int pozx, int pozy, int lat, int inalt, int vmax)
            {
                x0 = pozx;
                y0 = pozy;
                w = lat;
                h = inalt;
                val_max = vmax;
```

```
            }
        }
        private void Form1_Load(object sender, EventArgs e)
        {
            date_r = "0,0,0,0,0,0,0,0,0,0,0".Split(delimiter);
            date_c = "0,0,0,0,0,0,0,0,0,0,0".Split(delimiter);
            desen = this.CreateGraphics();
            creion_blu = new System.Drawing.Pen(System.Drawing.Color.Blue);
            creion_rosu = new System.Drawing.Pen(System.Drawing.Color.Red);
            creion_gri = new
System.Drawing.Pen(System.Drawing.Color.LightGray);
            pens_blu = new
System.Drawing.SolidBrush(System.Drawing.Color.Blue);
            pens_red = new
System.Drawing.SolidBrush(System.Drawing.Color.Red);
            pens_back = new System.Drawing.SolidBrush(this.BackColor);
            font_nina = new System.Drawing.Font("Nina", 8);
            binar1 = new binar();
            binar1.init_binar(0, 275, 330, 30);
            Array.Resize(ref a0_valori, a0_n_maxx + 1);
            grafic0 = new osciloscop(desen, a0_pozx, a0_pozy, a0_n_maxx,
a0_n_maxy, a0_val_max);
            Array.Resize(ref a1_valori, a1_n_maxx + 1);
            grafic1 = new osciloscop(desen, a1_pozx, a1_pozy, a1_n_maxx,
a1_n_maxy, a1_val_max);
            termo1 = new termo(10, 50, 10, 200, 1023);
            termo2 = new termo(65, 50, 10, 200, 1023);
            termo3 = new termo(120, 50, 10, 200, 1023);
            termo4 = new termo(175, 50, 10, 200, 1023);
            termo5 = new termo(230, 50, 10, 200, 1023);
            //int i, j;
            // Listez porturile seriale
            Nume_porturi = System.IO.Ports.SerialPort.GetPortNames();
            this.Porturi_s.Items.Clear();

            //Adaug porturile exixtente

            for (int i = 0; i < Nume_porturi.Length; i++)
            {
                this.Porturi_s.Items.Add(Nume_porturi[i]);
            }

            //Pozitionarea listei pe primul element
            this.Porturi_s.SelectedIndex = 0;
            this.label1.Text = "Conectati-va la portul serial!";



        }

        private void timer1_Tick(object sender, EventArgs e)
        {
            k++;
            if (k > 123456789)
                k = 0;
            this.label2.Text = k.ToString();
```

```
            if (this.checkBox8.Checked)
                c0 = 1;
            else
                c0 = 0;
            if (this.checkBox7.Checked)
                c1 = 1;
            else
                c1 = 0;
            if (this.checkBox6.Checked)
                c2 = 1;
            else
                c2 = 0;
            if (this.checkBox5.Checked)
                c3 = 1;
            else
                c3 = 0;
            if (this.checkBox4.Checked)
                c4 = 1;
            else
                c4 = 0;
            if (this.checkBox3.Checked)
                c5 = 1;
            else
                c5 = 0;
            if (this.checkBox2.Checked)
                c6 = 1;
            else
                c6 = 0;
            if (this.checkBox1.Checked)
                c7 = 1;
            else
                c7 = 0;

            val_cmd = 128 * c7 + 64 * c6 + 32 * c5 + 16 * c4 + 8 * c3 + 4 *
c2 + 2 * c1 + c0;

            this.label16.Text = System.Convert.ToString(val_cmd);
            int transl = 0;
            int amplif = a0_n_maxy;
            int zero = a0_n_maxy - 1;

            if (this.serialPort1.IsOpen)
            {
                if (val_cmd == val_cmd_v)  //  nu am val_cmd noua deci
lanxsez comanda "AA"
                {
                    cda = "AA" + val_cmd.ToString();
                    this.serialPort1.Write(cda);
                    txt = "";
                    txt = this.serialPort1.ReadExisting();
                    if (txt.Length > 0)
                    {
                        //this.label1.Text = txt;
                        //Despachetare date
                        date_r = txt.Split(delimiter);
```

```
                        if (date_r.Length == 8) //Verific daca am primit 8
valori
                        {
                            //Pentru comanda M1, verific daca sum ade
control e corecta
                            //suma = Convert.ToInt16(date_r[0]) +
Convert.ToInt16(date_r[1]) + Convert.ToInt16(date_r[2]) +
Convert.ToInt16(date_r[3]) + Convert.ToInt16(date_r[4]) +
Convert.ToInt16(date_r[5]) + Convert.ToInt16(date_r[6]) +
Convert.ToInt16(date_r[7]);
                            //if (suma % 1024 == Convert.ToInt16(date_r[8]))
                            //{
                            this.label1.Text = txt;
                            date_c = txt.Split(delimiter);
                            //}
                        }
                        // Afisare A1
                        try
                        {
                            a0_val = System.Convert.ToInt16(date_c[2]);
                        }
                        catch (System.FormatException)
                        {
                            this.label1.Text = "Format necorespunzator!";
                        }

                        catch (System.IndexOutOfRangeException)
                        {
                            this.label1.Text = "Format necorespunzator!";
                        }
                        // Trasare grafic
                        int f = System.Convert.ToInt32(transl + zero -
amplif * System.Convert.ToDouble(a0_val) / a0_val_max);
                        if (f > a0_n_maxy)
                            f = a0_n_maxy - 1;
                        if (f < 0)
                            f = 0;

                        for (int i =0;i< a0_n_maxx - 1; i++)
                        {
                            a0_valori[i] = a0_valori[i + 1];
                        }
                        a0_valori[a0_n_maxx - 1] = f;

                        //desen.DrawImage(im, a0_pozx, a0_pozy);
                        grafic0.setval(a0_valori, a0_n_maxx);

                        // Afisare A0
                        try
                        {
                            a1_val = System.Convert.ToInt16(date_c[1]);
                        }
                        catch (System.FormatException)
                        {
                            this.label1.Text = "Format necorespunzator!";
                        }
```

```
                        catch (System.IndexOutOfRangeException)
                        {
                            this.label1.Text = "Format necorespunzator!";
                        }
                        // Trasare grafic

                        f = System.Convert.ToInt32(transl + zero - amplif *
System.Convert.ToDouble(a1_val) / a1_val_max);
                        if (f > a1_n_maxy)
                            f = a1_n_maxy - 1;
                        if (f < 0)
                            f = 0;
                        for (int i = 0; i < a1_n_maxx - 1; i++)
                        {
                            a1_valori[i] = a1_valori[i + 1];
                        }
                        a1_valori[a1_n_maxx - 1] = f;

                        //desen.DrawImage(im, a1_pozx, a1_pozy);
                        grafic1.setval(a1_valori, a1_n_maxx);

                        //Afisare valori analogice

                        try
                        {
                            a2_val = System.Convert.ToInt16(date_c[5]);
                            a3_val = System.Convert.ToInt16(date_c[4]);
                            a4_val = System.Convert.ToInt16(date_c[3]);
                            a5_val = System.Convert.ToInt16(date_c[2]);
                            this.label4.Text = date_c[5].ToString();
                            this.label6.Text = date_c[4].ToString();
                            this.label8.Text = date_c[3].ToString();
                            this.label10.Text = date_c[2].ToString();
                            this.label12.Text = date_c[1].ToString();
                            this.label14.Text = date_c[0].ToString();
                            digi = System.Convert.ToUInt64(date_c[7]);
                            this.label26.Text =
System.Convert.ToString(digi);
                        }
                        catch (System.FormatException)
                        {
                            this.label1.Text = "Format necorespunzator!";
                        }

                        // Afisare DI sub forma binara

                        binar1.setval(8, digi, desen, creion_rosu, pens_blu,
pens_back);

                        //Afisare termo1
                        termo1.desenez(desen, creion_blu, creion_gri,
pens_red, font_nina);
                        termo1.setval(System.Convert.ToInt16(date_c[5]),
desen, pens_blu, pens_back);
```

```
                        termo2.desenez(desen, creion_blu, creion_gri,
pens_red, font_nina);
                        termo2.setval(System.Convert.ToInt16(date_c[4]),
desen, pens_blu, pens_back);
                        termo3.desenez(desen, creion_blu, creion_gri,
pens_red, font_nina);
                        termo3.setval(System.Convert.ToInt16(date_c[3]),
desen, pens_blu, pens_back);
                        termo4.desenez(desen, creion_blu, creion_gri,
pens_red, font_nina);
                        termo4.setval(System.Convert.ToInt16(date_c[2]),
desen, pens_blu, pens_back);
                        termo5.desenez(desen, creion_blu, creion_gri,
pens_red, font_nina);
                        termo5.setval(System.Convert.ToInt16(date_c[1]),
desen, pens_blu, pens_back);
                    }
                    else
                    {
                        this.label1.Text = "Nu vin date!";
                    }
                }
                else   //trebuie sa dau comanda "O,val_cmd"
                {
                    this.serialPort1.Write("O" +
System.Convert.ToString(val_cmd));
                    val_cmd_v = val_cmd;
                }
            }


        }

        private void but_con_Click(object sender, EventArgs e)
        {
            if (!this.serialPort1.IsOpen)
            {
                this.serialPort1.PortName =
System.Convert.ToString(this.Porturi_s.Items[this.Porturi_s.SelectedIndex]);
                this.serialPort1.Open();
                this.label1.Text = "Portul a fost deschis";
            }
            else
            {
                this.label1.Text = "Portul este deja deschis deschis";
            }
            this.serialPort1.Write("AA");
        }

        private void but_dec_Click(object sender, EventArgs e)
        {
            this.serialPort1.Close();
            this.label1.Text = "Portul este deconectat!";
        }
    }
}
```

Rulam aplicatia si obtinem: