

CERCETĂRI PRIVIND PROIECTAREA ȘI REALIZAREA APLICAȚIILOR SOFTWARE DE LEGĂTURĂ DINTRE INTERFEȚELE HARDWARE ȘI DEVICE

GÂNGU Andrei-Bogdan

Conducători științifici: Conf.dr.ing. **George ENCIU**, Asist. univ. dr. ing. **Adrian POPESCU**

REZUMAT: Visual C++ este un mediu de programare integrat pentru limbajul C++, produs de Microsoft. C++ este una dintre componentele Visual C++. Totuși, compilatorul poate procesa atât cod sursă C cât și C++ iar versiunea. Visual C++ include de asemenea o colecție mare și elaborată de unelte de dezvoltare software (software development tools) toate folosite printr-o interfață cu ferestre.

1 INTRODUCERE

Lucrarea de față își propune să illustreze modalitățile de interfațare cu structurile hardware externe. Prin limbajul C++ se creează legătura dintre interfețele hardware și device.

Folosind Visual Studio C++ am realizat conexiunea dintre două porturi COM.

2 STADIUL ACTUAL

Crearea unei interfețe grafice în Visual C++ 2010 (vezi figura 1) :

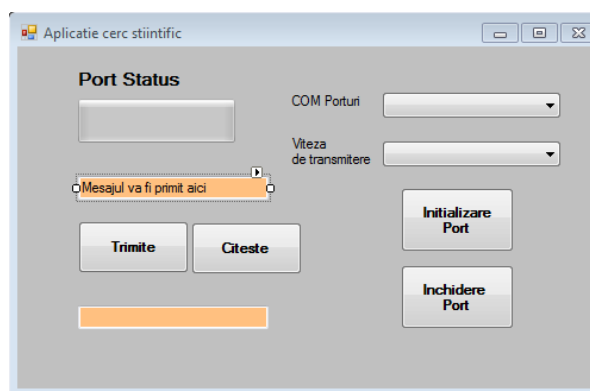


Fig. 1. Interfața grafică

Specializarea Logistică industrială, Facultatea IMST;

E-mail: bogdan.gangu@gmail.com;

2.1 Lucrul cu barele de instrumente (Toolbox)

Butoanele de pe toolbox (vezi figura 2) corespund la comenzi, la fel ca articolele dintr-un meniu. În general butoanele din toolbar sunt duplicate pentru comenzi din meniu (shortcut-uri).

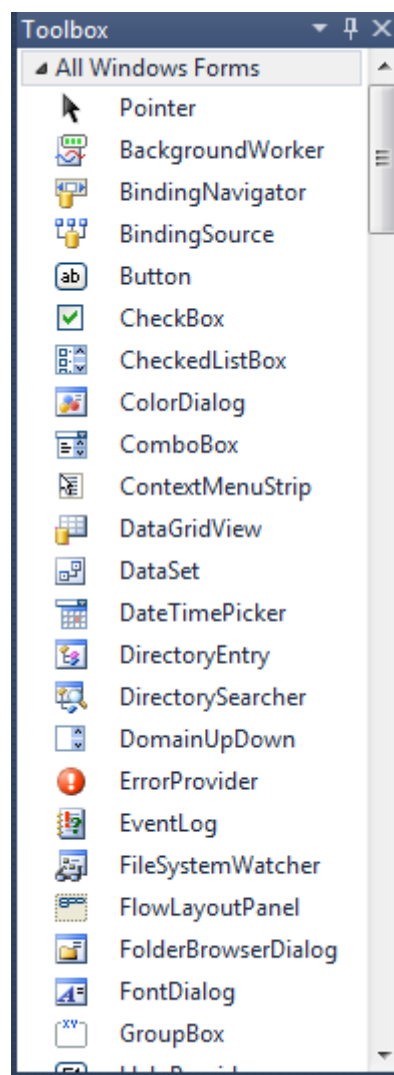


Fig. 2 . Toolbox

CERCETĂRI PRIVIND PROIECTAREA ȘI REALIZAREA APLICAȚIILOR SOFTWARE DE LEGĂTURĂ DINTRE INTERFEȚELE HARDWARE ȘI DEVICE

2.2 Crearea unei casuțe care afișează o listă a porturilor:

Din Toolbox selectam opțiunea ComboBox si apoi desenăm icoana căsuței(vezi figura 3).



Fig. 3 .Lista porturilor

Apoi se realizeaza inca o casuta in care afișăm Viteza de transmitere (vezi figura 4) .Aceasta casuta afișeaza viteza optima 9600.

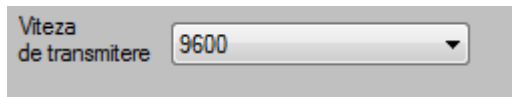


Fig. 4. Rata de transfer

2.3 Crearea unui buton:

Din Toolbox selectăm opțiunea button (vezi figura 5) și apoi desenam icoana butonului.

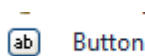


Fig. 5.Optiunea button

Acest buton (vezi figura 6) va inițializa portul selectat.Apoi dând dublu click va afisa codul zunde putem scrie algoritmul (vezi figura 7) .



Fig. 6.Buton de inițializare

```
//init buton
private: void button1_Click(System::Object^ sender, System::EventArgs^ e) {
    this->textBox1->Text=String::Empty;
    if(this->comboBox1->Text==String::Empty || this->comboBox2->Text==String::Em
        this->textBox1->Text="Selecteaza setarile portului";
    else {
        try{
            // make sure port isn't open
            if(!this->serialPort1->IsOpen){
                this->serialPort1->PortName=this->comboBox1->Text;
                //this->textBox1->Text=this->comboBox1->Text;
                this->serialPort1->BaudRate=Int32::Parse(this->comboBox2->Text);
                //this->textBox1->Text=this->comboBox2->Text;
                this->textBox2->Text="Scrie mesajul aici!";
                //open serial port
                this->serialPort1->Open();
                this->progressBar1->Value=100;
            }
            else{
                this->textBox2->Text="Portul nu este deschis";
            }
        } catch(UnauthorizedAccessException){
            this->textBox2->Text="Acces neautorizat";
        }
    }
}
```

Fig. 7.Algoritmul de initializare si pentru erori

2.4 Inserarea unei componente:

Selectăm opțiunea SerialPort (vezi figura 8) și il desenăm pe fereastra de lucru (vezi figura 9) .

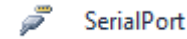


Fig. 8.Optiunea din Toolbox

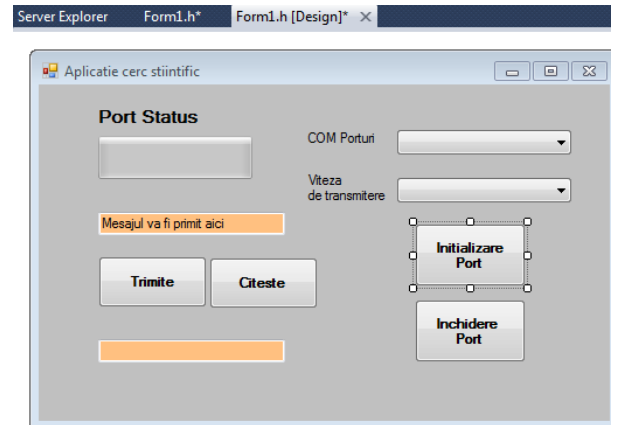


Fig. 9.Componenta desenată

Apoi scriem in cod algoritmul (vezi figura 10) pentru a gasi porturi disponibile.

```
}
#pragma endregion
// find available ports
private: void findPorts(void)
{
    // get port names
    array<Object^>^ objectArray = SerialPort::GetPortNames();
    // add string array to combobox
    this->comboBox1->Items->AddRange( objectArray );
}
}
```

Fig. 10.Algoritmul de căutare

2.5 Crearea butonului de inchiderea portului

Folosim opțiunea anterioară, button.

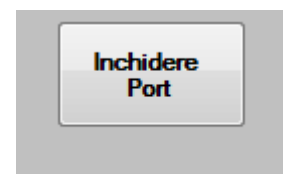


Fig. 11.Buton inchidere port

Apoi dam dublu click pe buton si scriem algoritmul (figura 12) .

```

    }
    //inchidere| buton
private: System::Void button2_Click(System::Object^ sender, System::EventArgs^ e) {
    //close serialPort
    this->serialPort1->Close();
    // update progress bar
    this->progressBar1->Value=0;
    // Enable read button
    this->button4->Enabled = true;
    // Enable the init button
    this->button1->Enabled = true;
}
    }
    
```

Fig.12.Algoritmul de inchidere port

2.6 Crearea unei casute de progres:

Din Toolbox alegem optiunea ProgressBar (figura 13)apoi o desenam pe fereastra de lucru.

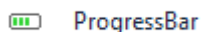


Fig.13.Optiunea pentru status

Aceasta optiune (figura 14) arata daca portul este deschis sau exista o eroare.

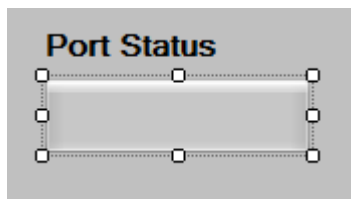


Fig. 14.Statusul portului

Dupa ce se inițializează portul,căsuța (vezi figura 15) se va face verde.

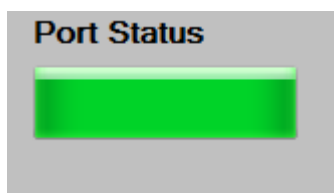


Fig. 15.Port inițializat

2.7 Crearea unei căsuțe de afișare:

Din Toolbox alegem optiune Label si o desenăm pe fereastra de lucru (figura 16).



Fig. 16.Casuța de afișare

Apoi introducem un text de afisare (figura 17).

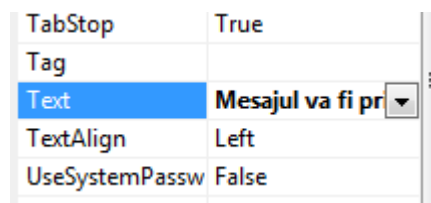


Fig. 17.Țditare text

In figura 18 este ilustrată informația despre casuță.

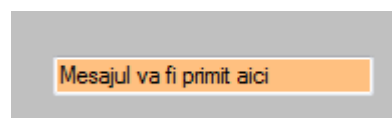


Fig. 18.Locația mesajului

Apoi introducem o casuta pentru a scrie mesajul (figura 19) urmand pasii anteriori.

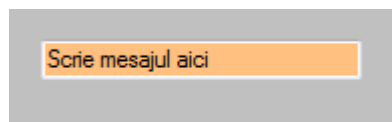


Fig. 19.Scriere mesaj

2.8 Crearea butonului de trimitere (figura 20):

Folosim opțiunea button si il desenăm pe fereastra de lucru.



Fig. 20.Buton trimitere

Dăm dublu click și scriem algoritmul (figura 21) in cod.

```

//Butonul de trimitere|
private: System::Void button3_Click(System::Object^ sender, System::EventArgs^ e) {
    // add sender name
    String^ name = this->serialPort1->PortName;
    // grab text and store in send buffer
    String^ message = this->textBox2->Text;
    // write to serial
    if(this->serialPort1->IsOpen)
        //this->_serialPort->WriteLine(String::Format("<0>: {1}",name,message));
        this->serialPort1->WriteLine(message);
    else
        this->textBox2->Text="Portul nu este deschis";
}
    
```

Fig. 21. Algoritmul de trimitere

CERCETĂRI PRIVIND PROIECTAREA ȘI REALIZAREA APLICAȚIILOR SOFTWARE DE LEGĂTURĂ DINTRE INTERFEȚELE HARDWARE ȘI DEVICE

2.9 Crearea unui buton de citire a mesajului:

Folosind opțiunea button din Toolbox desenăm butonul (figura 22) pe fereastra de lucru.



Fig. 22. Buton citire

Apoi dăm dublu click pe buton și intrăm în cod și scriem algoritmul (figura 23).

```
private: //Buton de citire
System::Void button4_Click(System::Object^ sender, System::EventArgs^ e)
{
    // check if port is ready for reading
    if(this->serialPort1->IsOpen){
        // Reset the text in the result label.
        this->textBox1->Text = String::Empty;

        // this will read manually
        try{
            this->textBox1->Text=this->serialPort1->ReadLine();
        }
        catch(TimeoutException^){
            this->textBox1->Text="Fara conexiune";
        }
        // Disable the init button
        // the asynchronous operation is done.
        this->button1->Enabled = false;
    }
    else
        // give error warning
        this->textBox1->Text="Portul nu este deschis";
}
}
```

Fig. 23. Algoritmul de citire

2.10 Erori de executare ale aplicației software:

Dacă nu este deschis niciun port sau nu este selectată rata de transfer va apărea mesajul (figura 24) "Selectează setările portului" când vom inițializa portul.

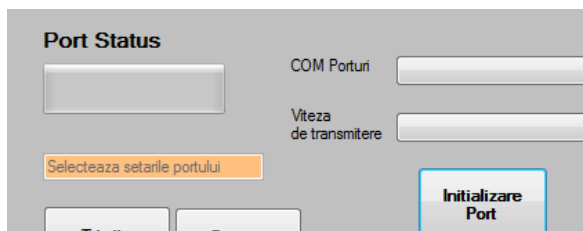


Fig. 24. Eroare inițializare

Dacă apăsăm butonul trimite fără să inițializăm portul va apărea mesajul (figura 25) "Portul nu este deschis".

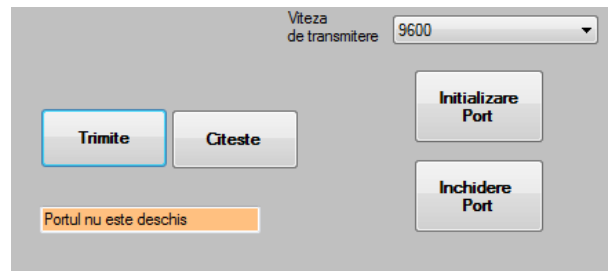


Fig. 25. Eroare trimitere

2.11 Crearea porturilor virtuale:

Deoarece nu am niciun serial port conectat în laptop am creat două porturi COM cu aplicația Configure Serial Port (figura 26).

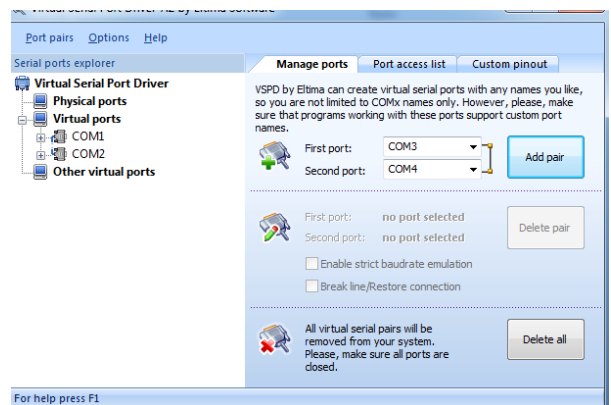


Fig. 26. Creare port virtual

3 CONCLUZII

În momentul de față am realizat modalitatea de comunicare dintre două COM-uri. Aceste COM-uri au fost realizate virtual cu aplicația Virtual Serial Port.

Pe viitor aș vrea să fac legătura dintre interfața grafică și bazele de date. Să creez o modalitate ușoară de interfatare cu bazele de date.

4 BIBLIOGRAFIE

[1] Autor, Liviu Negrescu, Limbajele C și C++ pentru începători;

[2] <https://www.youtube.com/watch?v=icQEK4HmLqU>. Accesat la data: 05.03.2015.

[3] <https://www.youtube.com/watch?v=cxgKTWTHX1w>. Accesat la data: 6.04.2015.