

Ministerul Educatiei
Universitatea Tehnica a Moldovei
FCIM

RAPORT

La Lucrarea de laborator №1

La disciplina «MIDPS»

Tema: «**MEDIUL INTEGRAT C++
BUILDER**»

A verificat:

lect. Cojanu Irina

A efectuat:

st.gr. TI-145 Vitcovich Igor

Obiectivele lucrării

a) Însușirea modului de utilizare a celor mai importante componente ale mediului integrat C++ BUILDER . Realizarea unui program simplu care utilizează componente de tip *TButton*, *TEdit*, *Tlabel*, *RadioButton* etc.

b) Însușirea modului de utilizare a componentei VCL **TTimer**. Însușirea modului de utilizare a funcțiilor de lucru cu timpul sistem. Realizarea unor aplicații de gestionare a resursei timp.

c) Însușirea modului de utilizare a componentelor VCL **TPaintBox** și **TPanel**. Însușirea modului de utilizare a principalelor funcții grafice ale mediului C++BUILDER . Realizarea unor elemente pentru afișarea grafică a informației (diagramă și bargraf).

Sarcina Lucrarii

1) Se modifică programul din *Project1.cpp* astfel încât să se obțină forma cu obiecte din figura 4.1 ;

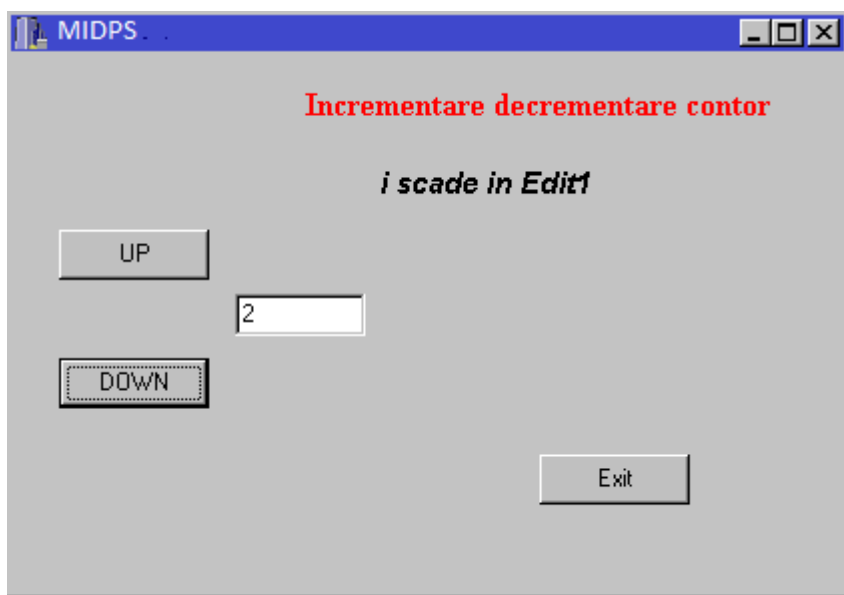


Fig. 4.1 – Realizarea 1

Se vor utiliza următoarele obiecte (în afara formei):

- două butoane (Button 1 și 2) pentru incrementarea (UP) respectiv decrementarea (DOWN) a unei variabile întregi **i** ;
- un buton (Button 3) pentru ieșirea din program (Exit);
- o casetă de editare (Edit1) unde se va afișa valoarea variabilei **i**;
- două etichete (Label1 și 2) pentru afișarea textului „**Incrementare decrementare contor.**”
Respectiv a **sensului de variație a variabilei i din caseta Edit1**;
- în caption-ul formei se va afișa textul „**MIDPS 1- A**”;
- fiecare obiect va avea hint-ul activ completat corespunzător .

2)) Se elaborează un program pentru realizarea unui cronometru.

Se vor utiliza următoarele obiecte, evidențiate în figura 4.2:

- o formă (*Form1*) pe care sunt dispuse celelalte obiecte și în *Caption*-ul căreia se va afișa textul „MIDPS”;

- patru butoane (*Button 1, 2, 3, 4*) cu următoarele funcții:
 - Button1 – pornirea cronometrului(*Caption Start*);
 - Button2 – oprirea cronometrului(*Caption Stop*);
 - Button3 – inițializarea cronometrului(*Caption Zero*);
 - Button4 – ieșirea din program (*Caption Exit*).
- două timere (*Timer1* și *Timer2*) cu următoarele funcții
 - *Timer1* (*Interval=1000 ms*) utilizat la afișarea timpului curent;
 - *Timer2* (*Interval=100 ms*) utilizat pentru cronometru;
- două casete de editare (*Edit1* și *Edit2*) utilizate pentru :
 - *Edit1* - afisarea datei și orei curente;
 - *Edit2* - afișarea timpului cronometrat;
- două etichete (*Label1* și *Label2*) cu *Caption*-ul conform figurii 2.4

Observații:

- din primele trei butoane, la un un moment dat va fi activ unul singur;
- fiecare obiect va avea *hint*-ul activ completat corespunzător;

În timpul execuției programului forma va avea aspectul din figura 4.3

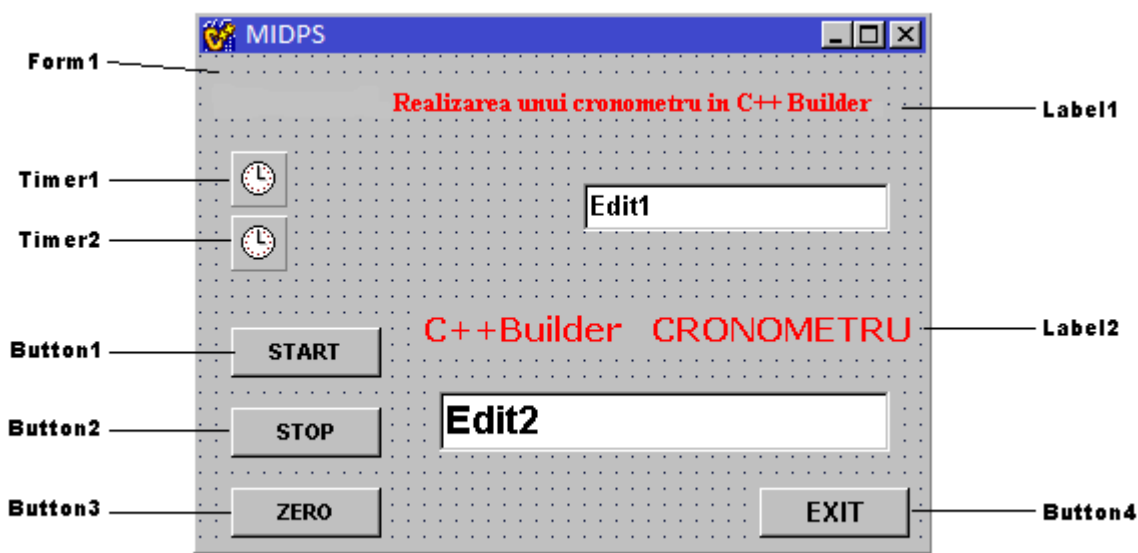


Fig 4.2 – Realizarea 2

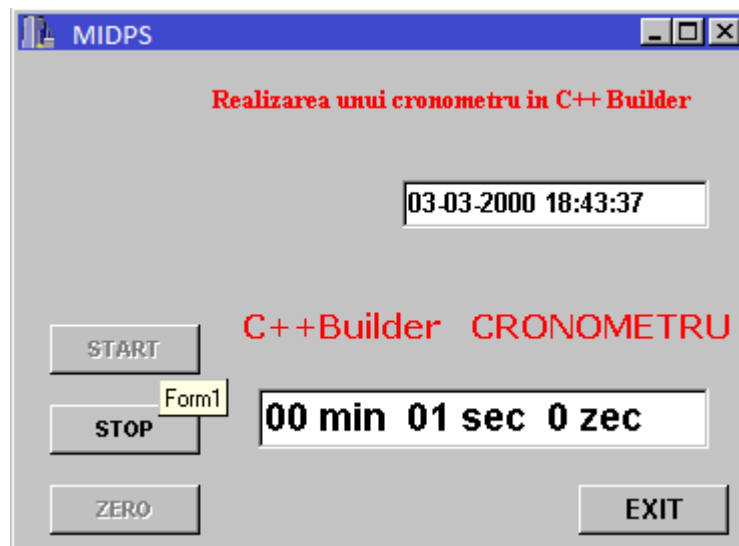


Fig.4.3 – Rezultatul aplicatiei 2

Se elaborează un program pentru realizarea a două elemente de afișare (bargraf și diagramă cu avans continuu) pentru care forma arată ca în figura 4.4 pe care sunt dispuse următoarele obiecte:

- o formă (*Form1*) în *Caption*-ul căreia se va afișa textul „**MIDPS**”;
- trei butoane (*Button 1, 2, 3*) cu următoarele funcții:
 - Buton1 – activarea afișării în diagramă și în bargraf (*Caption Start*);
 - Buton2 – oprirea afișării în diagramă și în bargraf (*Caption Stop*);
 - Buton3 – ieșirea din program (*Caption Exit*).

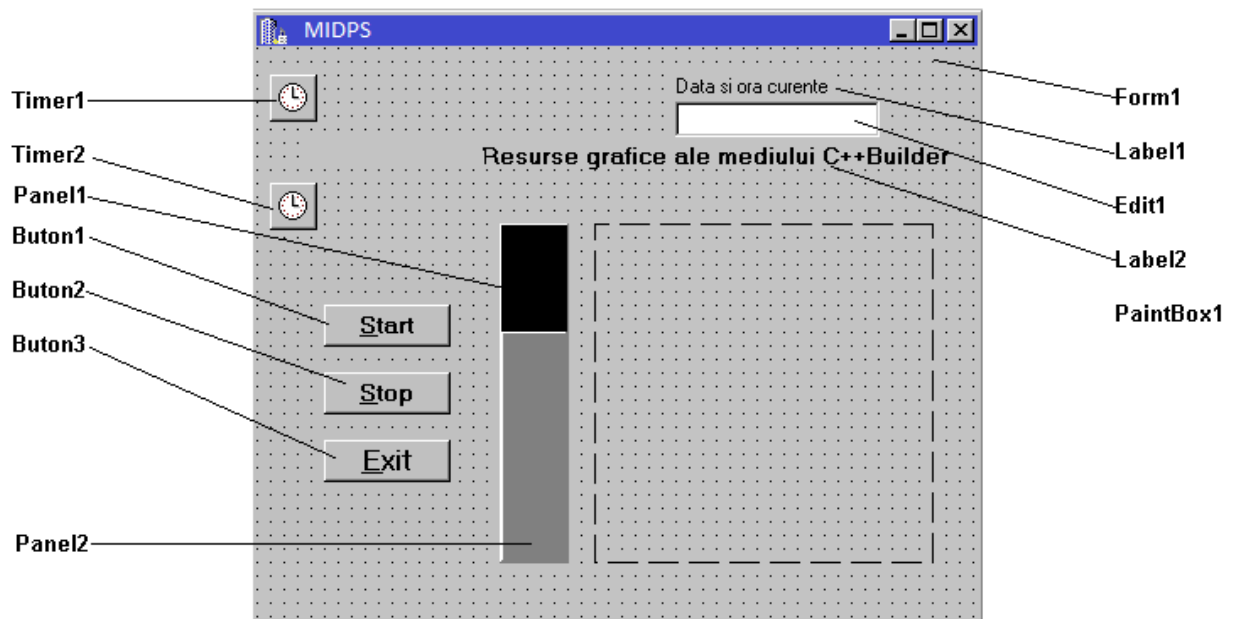


Fig 4.3 – Aplicatia 3

- două timere (*Timer1* și *Timer2*) cu următoarele funcții
 - *Timer1* (*Interval=1000 ms*) utilizat la afișarea timpului curent;
 - *Timer2* (*Interval=500 ms*) pentru intervalul de afișare în diagramă și în bargraf;
- o casetă de editare (*Edit1*) utilizată pentru afișarea datei și orei curente;
- două etichete (*Label1* și *Label2*) cu Caption-ul conform figurii 4.4

Observații:

- din primele două butoane, la un un moment dat va fi activ unul singur;
- fiecare obiect va avea *hint*-ul activ completat corespunzător;
- valoarea numerică ce se va afișa în cele două elemente grafice se obține cu funcția *random* după care numărul generat se va converti în pixeli ținându-se cont de înălțimea comună a graficului și bargrafului
- pentru realizarea bargrafului se vor utiliza două obiecte de tip *TPanel* de culori diferite care se vor suprapune;
- pentru desenarea graficului se vor utiliza funcțiile *MoveTo*, *LineTo* iar pentru avansul acestuia funcția *CopyRect*.

În timpul execuției programului forma va avea aspectul din figura 4.4.

Indiplinirea sarcinilor

1-A :

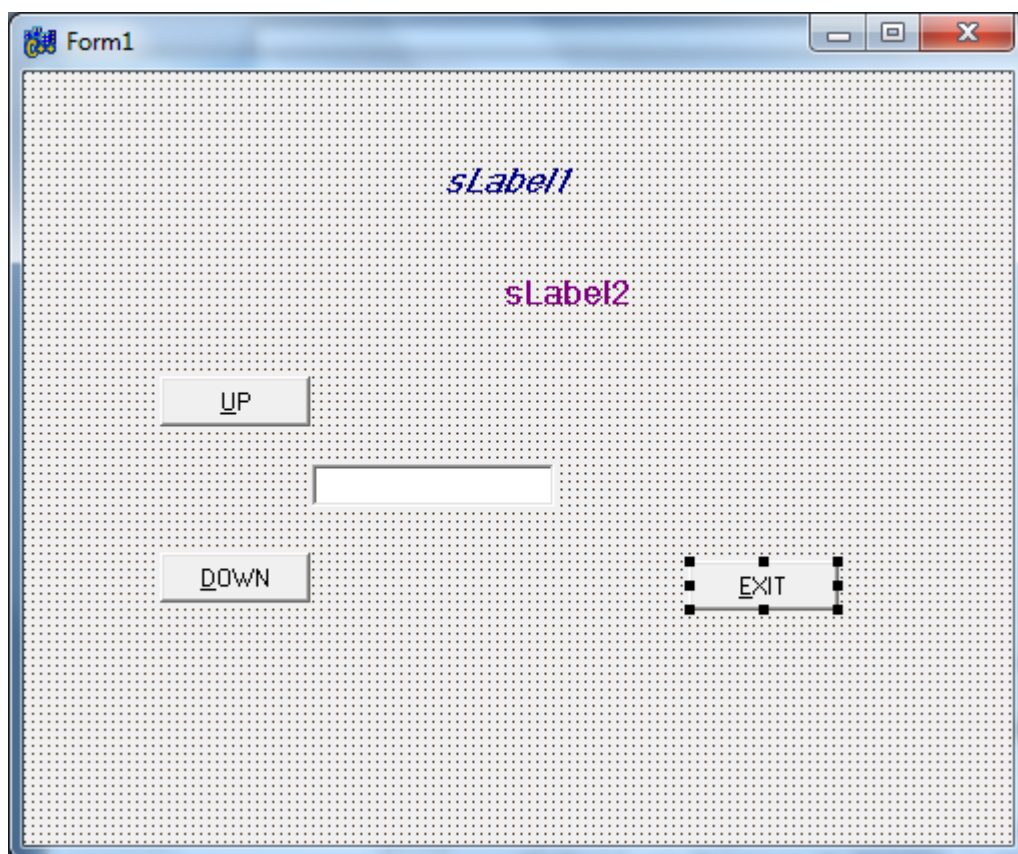


Fig 1.1 – Realizarea punctului 1-A

Programul realizeaza increment/decrement prefics prin apasarea butoanelor.

Listing programului 1-A

```
#include <vcl.h>
#pragma hdrstop

#include "Unit1.h"
//-----
#pragma package(smart_init)
#pragma link "sButton"
#pragma link "sEdit"
#pragma link "sLabel"
#pragma link "sMemo"
#pragma resource "*.dfm"
TForm1 *Form1;
//-----
int i = 0;
__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
    sLabel1 -> Caption = "Increment/decrement";
    sLabel2 -> Caption = "MIDPS 1-A";
}
//-----
//Butonul DOWN
void __fastcall TForm1::sButton2Click(TObject *Sender)
{
    sEdit1-> Text = --i;
}
//-----
//Butonul UP
void __fastcall TForm1::sButton1Click(TObject *Sender)
{
    sEdit1-> Text = ++i;
}
//-----
//Butonul EXIT
void __fastcall TForm1::sButton3Click(TObject *Sender)
{
    Close();
}
//-----
```

În rezultat primim un fișier executabil, care realizează increment și decrement.

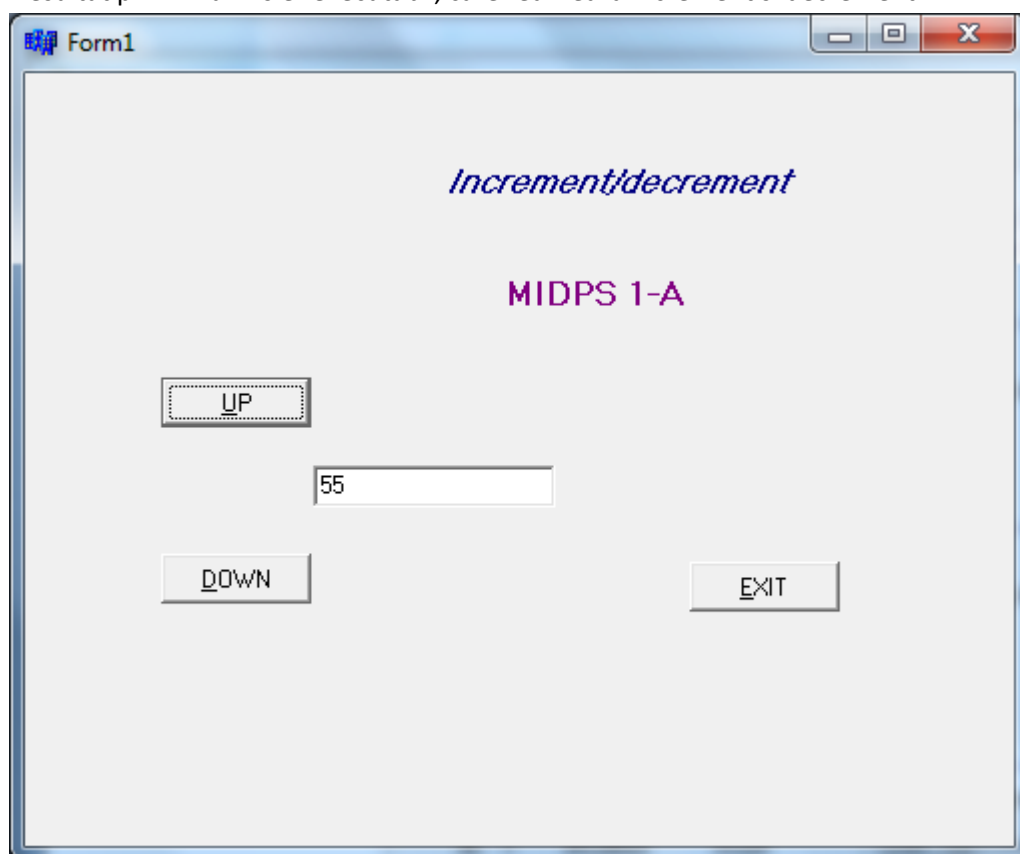


Figura 1.2 – Exemplu execuției programului 1-A

1-B :

În constructorul Formă programului 1-B este realizată în felul următor :

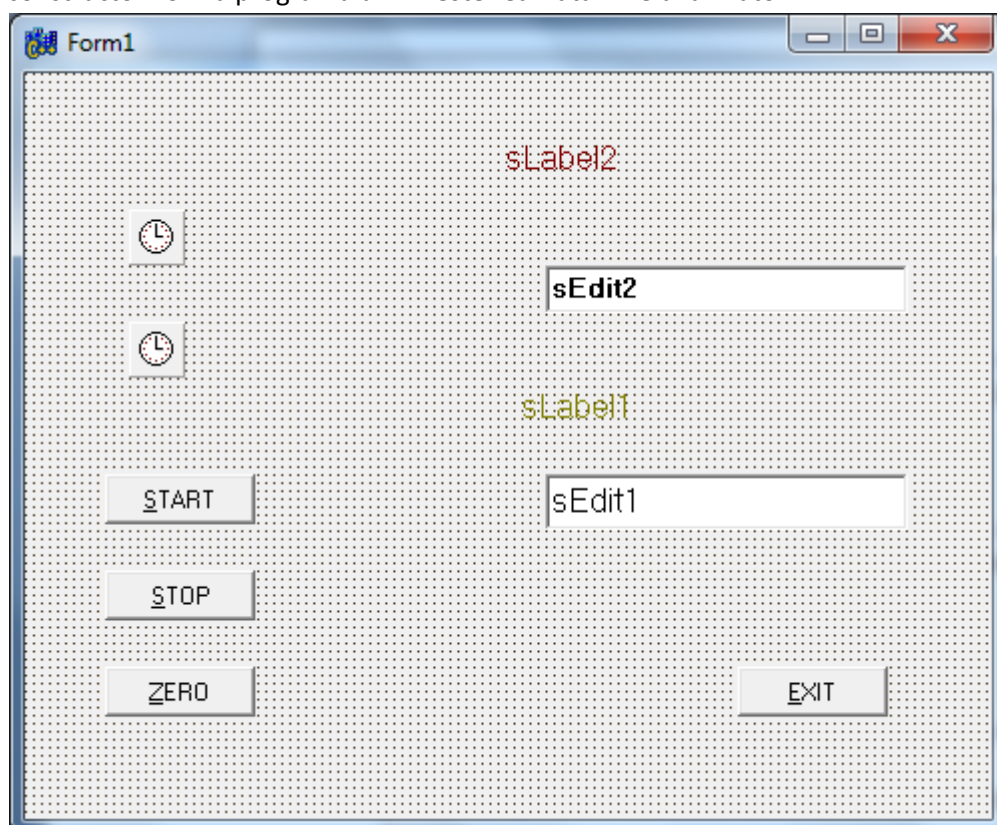


Figura 2.1 – Realizarea punctului 1-B

Listing programului 1-B

```
//-----
#include <stdio.h>
#include <vcl.h>
#pragma hdrstop
#include <dos.h>
#include "Unit4.h"
//-----
#pragma package(smart_init)
#pragma link "sButton"
#pragma link "sEdit"
#pragma link "sLabel"
#pragma resource "*.dfm"
TForm1 *Form1;
int i = 0;
int minutes = 0;
struct date d; //Exemplarul structure date
struct time t; //Exemplaru structure time
//-----
__fastcall TForm1::TForm1(TComponent* Owner)
: TForm(Owner)
{
    sEdit1 -> Text = "";
    sEdit2 -> Text = "";
    Timer1 -> Enabled = false;
    sButton1 -> Enabled = false;
    sButton2 -> Enabled = true; //Puteam apasa numai pe butonul START
    sButton3 -> Enabled = false;
    sLabel1 -> Caption = "Cronometru"; //Text in Label 1
    sLabel2 -> Caption = "MIDPS lab. 1-B"; //Text in Label 2
}
//-----

void __fastcall TForm1::Timer1Timer(TObject *Sender)
{
    i++;
    char buf[20]; //Cream buffer to fill with help of sprintf
    Timer1 -> Tag = i; //Variabila cronometrului
    if( i == 600 ){
        i = 0; //Pentru a proiecta timpul
        minutes++; //In felul obisnuit
    }
    sprintf(buf,"%d min. %d sec. %d ms.",
        minutes, Timer1->Tag/10,
        Timer1->Tag%10);
```

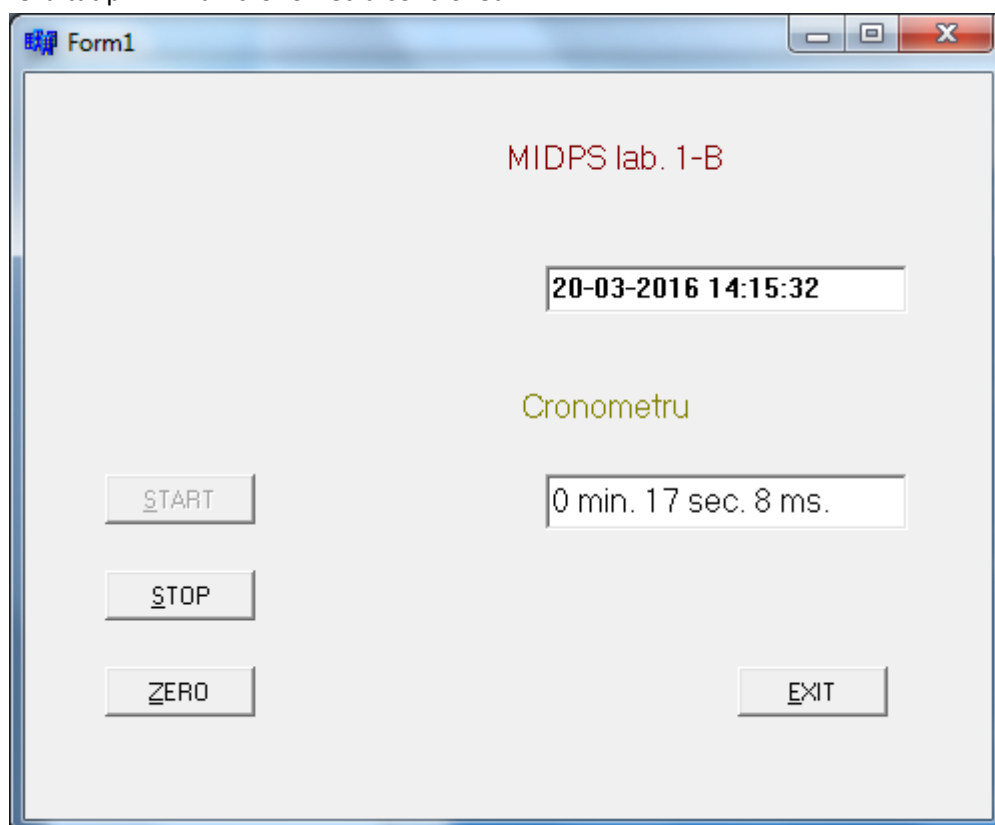


```

        sEdit1 -> Text = buf;
    }
    //-----
    //Apasarea pe Butonul EXIT
    void __fastcall TForm1::sButton4Click(TObject *Sender)
    {
        Close();
    }
    //-----
    void __fastcall TForm1::Timer2Timer(TObject *Sender)
    {
        char buff[20];
        getdate(&d); //Primim data cu ajutorul funtiei getdate
        gettime(&t); // Primim timpul cu ajutorul funtiei getdate
        sprintf(buff,"%02d-%02d-%4d %02d:%02d:%02d",d.da_day,d.da_mon,d.da_year,
        t.ti_hour,t.ti_min,t.ti_sec);
        sEdit2 -> Text = (AnsiString) buff;
    }
    //-----
    //Apasarea pe butonul START
    void __fastcall TForm1::sButton2Click(TObject *Sender)
    {
        sButton1 -> Enabled = true; // STOP devine enable
        sButton2 -> Enabled = false; // START devine inactive
        sButton3 -> Enabled = true; //ZERO devine enable
        Timer1 -> Enabled = true; //launch timerului
    }
    //-----
    //Apasarea pe butonul STOP
    void __fastcall TForm1::sButton1Click(TObject *Sender)
    {
        sButton1 -> Enabled = false;
        sButton2 -> Enabled = true; //START enable
        sButton3 -> Enabled = true; //ZERO enable
        Timer1 -> Enabled = false; //STOP timerului
    }
    //-----
    //Apasarea pe butonul ZERO face egal cu zero timpul cronometrului
    void __fastcall TForm1::sButton3Click(TObject *Sender)
    {
        i = 0;
        minutes = 0;
    }
    //-----

```

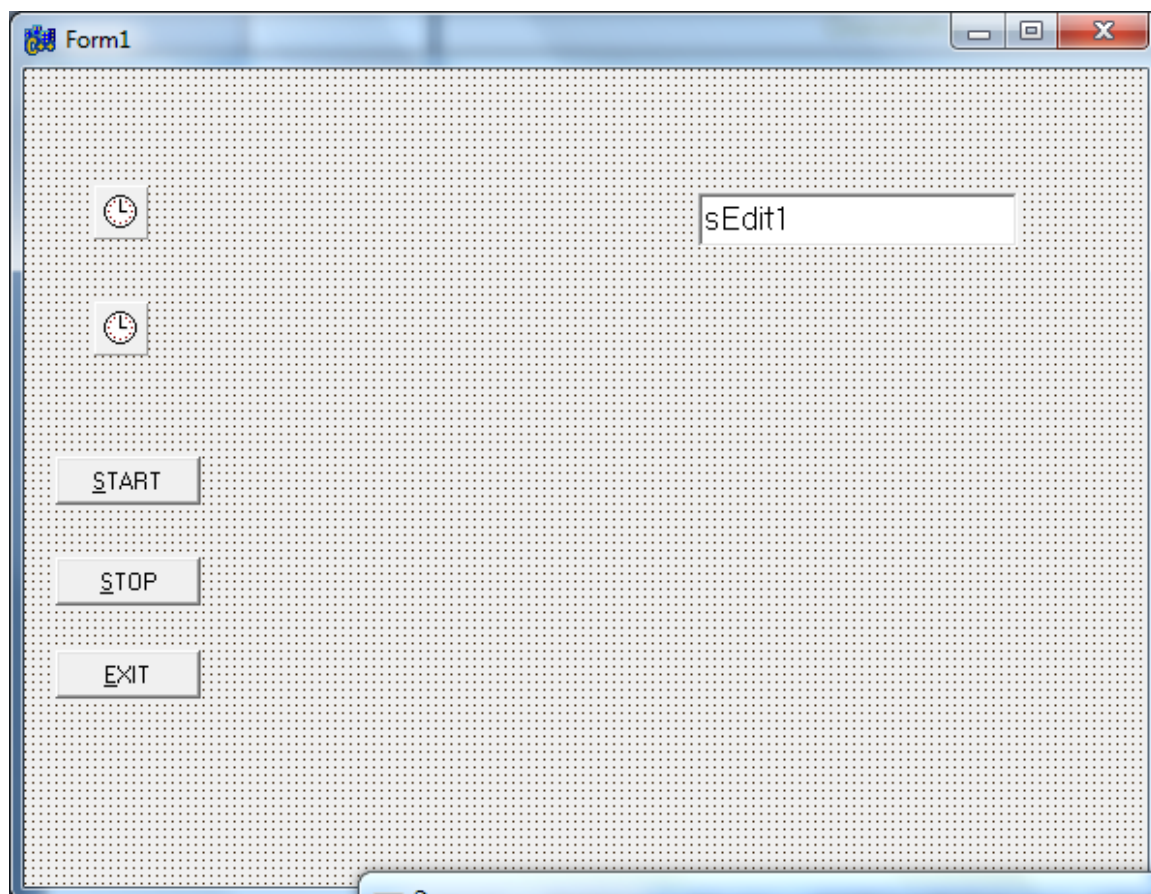
În rezultat primim un cronometru controlled .



The screenshot shows a Windows application window titled "Form1". The window has a standard Windows XP-style title bar with minimize, maximize, and close buttons. The main content area is light gray. At the top center, the text "MIDPS lab. 1-B" is displayed in red. Below this, a text box contains the date and time "20-03-2016 14:15:32". Further down, the word "Cronometru" is displayed in green. Below "Cronometru", another text box shows the time "0 min. 17 sec. 8 ms.". On the left side, there are three buttons stacked vertically: "START", "STOP", and "ZERO". On the right side, there is a single button labeled "EXIT".

Figura 2.2 – Exemplu de executia programului 1-B

1-C



The screenshot shows a Windows application window titled "Form1" in the Visual Studio constructor. The window has a standard Windows XP-style title bar. The main content area is a light gray grid. On the left side, there are three clock icons stacked vertically. Below the clock icons, there are three buttons stacked vertically: "START", "STOP", and "EXIT". On the right side, there is a text box labeled "sEdit1".

Figura 3.1 – Forma 3-C in constructor

In punctul C diagrama si barometrul vor fi « desenate » numai dupa apasarea butonului START.

Listing programului 1-B

```
#include <stdio.h>
#include <dos.h>
#include <vcl.h>
#pragma hdrstop
#include "Unit5.h"
//-----
#pragma package(smart_init)
#pragma link "sEdit"
#pragma link "sButton"
#pragma resource "*.dfm"
TForm1 *Form1;
struct date d;
struct time t;
int x, y;
randomize();
//-----
__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
    START -> Enabled = true;
    STOP -> Enabled = false;
    Timer2 -> Enabled = false;
}
//-----
//Cu ajutorul timerului 1 proiectam data si timpul ca si in punctul 1-B
void __fastcall TForm1::Timer1Timer(TObject *Sender)
{
    char buff[20];
    getdate(&d);
    gettime(&t);
    sprintf(buff,"%02d-%02d-%4d %02d:%02d:%02d",d.da_day,d.da_mon,d.da_year,
t.ti_hour,t.ti_min,t.ti_sec);
    sEdit1 -> Text = (AnsiString) buff;
}
//-----
void __fastcall TForm1::EXITClick(TObject *Sender)
{
    Close();
}
//-----
//Apasarea pe butonul START
void __fastcall TForm1::STARTClick(TObject *Sender)
{
    Canvas -> Pen -> Width = 1;
```

```

Canvas -> Pen -> Color = clGray;
Canvas -> Brush -> Color = clGray;
Canvas -> Rectangle(120,100, 170, 400); //Desenam rectangle background barometrului
Canvas -> Brush -> Color = clWhite;
Canvas -> Pen -> Color = clBlack;
START -> Enabled = false;
STOP -> Enabled = true;
Canvas -> Pen -> Style = psDash;
Canvas -> Rectangle(200,100,500,400); //Rectangle pentru diagrama
Canvas -> Pen -> Style = psSolid;
//Cream grid
for(int i = 200; i < 500; i += 10){
    Canvas -> MoveTo(i,100);
    Canvas -> LineTo(i,400);
}
for(int i = 100; i < 400; i += 10){
    Canvas -> MoveTo(200,i);
    Canvas -> LineTo(500,i);
}
//Un line bold in mijloc
Canvas -> MoveTo(500,250);
Canvas -> Pen -> Width = 3;
Canvas -> LineTo(200,250);
Canvas -> Pen -> Width = 2;
//Primele ordinate
x = 210;
y = 398;
Timer2 -> Enabled = true;
}
//-----
//Apasarea pe butonul STOP
void __fastcall TForm1::STOPClick(TObject *Sender)
{
    START -> Enabled = true;
    STOP -> Enabled = false;
    Timer2 -> Enabled = false;
}
//-----
void __fastcall TForm1::Timer2Timer(TObject *Sender)
{
    //Desenam un rectangle negru pe background sur
    //Pina frontier
    //Pentru a realize barametru
    if(y >= 100){
        Canvas -> Pen -> Width = 1;
        Canvas -> Pen -> Color = clBlack;
        Canvas -> Rectangle(120, 400, 170, y);
    }
}

```

```

        y -= rand() % 20; //Micsoram ordinata y cu o valoarea randoma
    }
    if (x != 500){
        Canvas -> Pen -> Width = 2;
        Canvas -> Pen -> Color = clRed;
        Canvas -> LineTo(x, rand() % 300 + 100); //Conectam 2 puncte, unul din care este random
        x += 10; //inmultim x
    }
}

```

In rezultat primim un program, care dupa apasare butonului START deseneaza: 1) un grid pe care dupa fiecare 500ms se apare o noua ramura digramei; 2)un barometrul care inmulteste cu o valoarea randoma dupa 500ms. Diagrama si barometrul termina inmultirea linga frontier. Desinarea se termina prin apasarea butonului STOP si deseneaza din inceput prin apasarea butonului START.

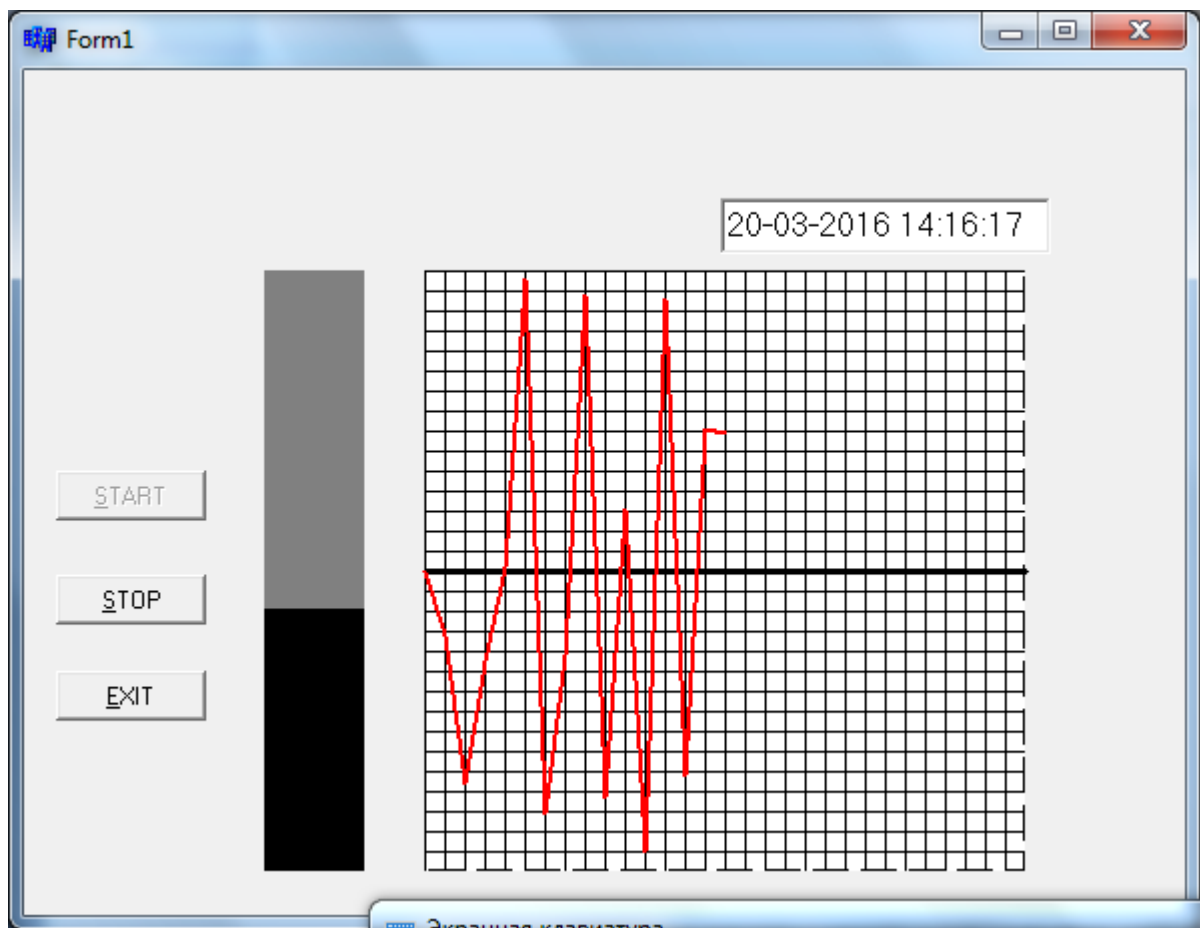


Figura 3.2 – Exemplu de executia programului 1-C

Concluziile

În cadrul acestei lucrări de laborator am studiat principiile și modul de utilizare a celor mai importante componente ale IDE Borland C++ Builder. C++ Builder are un plus important – cu ajutorul lui este posibil de a crea un fișier executabil, care poate fi transmis fără ceva adăugător. Așa ceva nu ne oferă chiar și Visual Studio. Builder conține un construtor cu ajutorul căruia putem să adăugăm pe forma alte componente. Obiectul Canvas ne ușurează lucrarea cu obiecte grafice. Iar cu ajutorul “Object Inspector” putem să schimbăm unele din opțiuni componentelor, fără a intra în cod.