Ministerul Educației al Republicii Moldova
Universitatea Tehnică a Moldovei

RAPORT

La lucrarea de laborator nr. 1
la disciplina «MIDPS»
«Mediul integrat C++ Builder»

A efectuat: studentul gr. T-145 Ialticenco A.
A verificat: lector univ. Cojocaru S.

Chisinau 2016

Obiectivele lucrării

a) Însuşirea 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) Însuşirea modului de utilizare a componentei VCL **TTimer.** Însuşirea modului de utilizare a funcţiilor de lucru cu timpul sistem. Realizarea unor aplicaţii de gestionare a resursei timp.

c) Însuşirea modului de utilizare a componentelor VCL **TPaintBox** şi **TPanel.** Însuşirea modului de utilizare a principalelor funcţii grafice ale mediului C++BUILDER . Realizarea unor elemente pentru afişarea grafică a informaţiei (diagramă şi bargraf).

Sarcina lucrarii

1) Vor fi examinate toate componentele prezentate în indicatii teoretice;

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

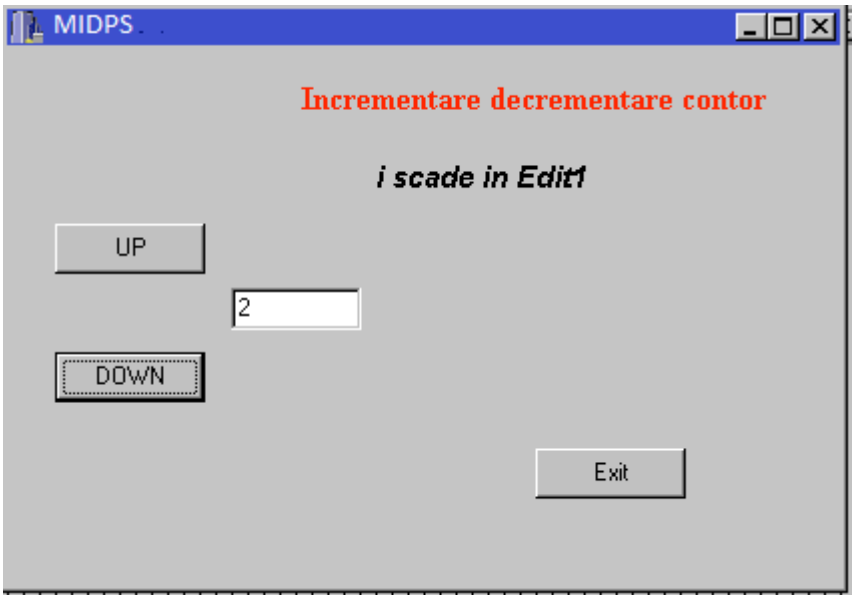


Fig. 4.1 – Realizarea 1

Se vor utiliza următoarele obiecte (în afara formei):

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

3) Se elaborează un program pentru realizarea unui cronometru.

Se vor utiliza următoarele obiecte, evidenţiate în figura 4.2:

- o formă (*Form1*) pe care sunt dispuse celelalte obiecte şi în *Caption*-ul căreia se va afişa textul **„MIDPS";**

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

Observaţii:

- 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 execuţiei programului forma va avea aspectul din figura 4.3

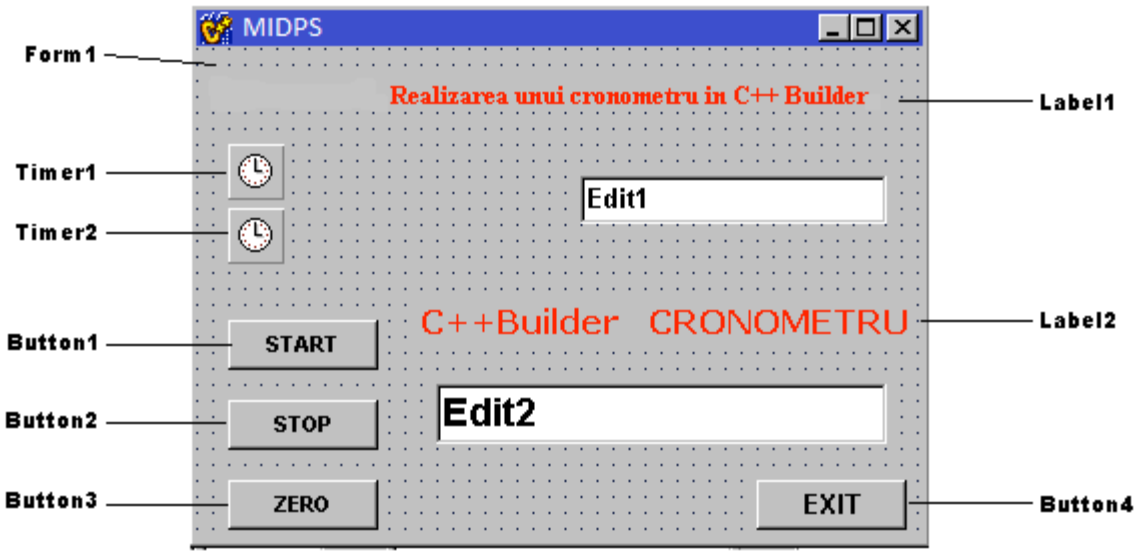


Fig 4.2 – Realizarea 2

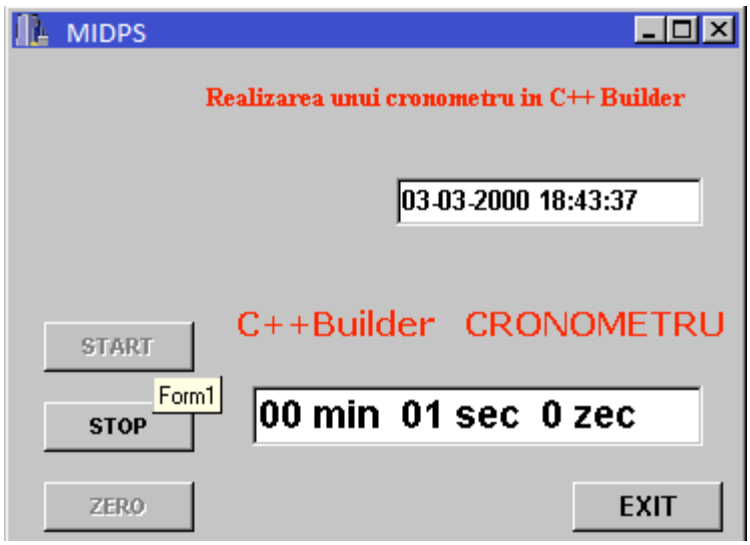


Fig.4.3 – Rezultatul aplicatiei 2

4) Se elaborează un program pentru realizarea a două elemente de afişare (bargraf şi 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 afişa textul **„MIDPS;**
- trei butoane (*Button 1, 2, 3*) cu următoarele funcţii:
  - Buton1 – activarea afişării în diagramă şi în bargraf (Caption **Start**);
  - Buton2 – oprirea afişării în diagramă şi în bargraf (Caption **Stop**);
  - Buton3 – ieşirea din program (Caption **Exit**).

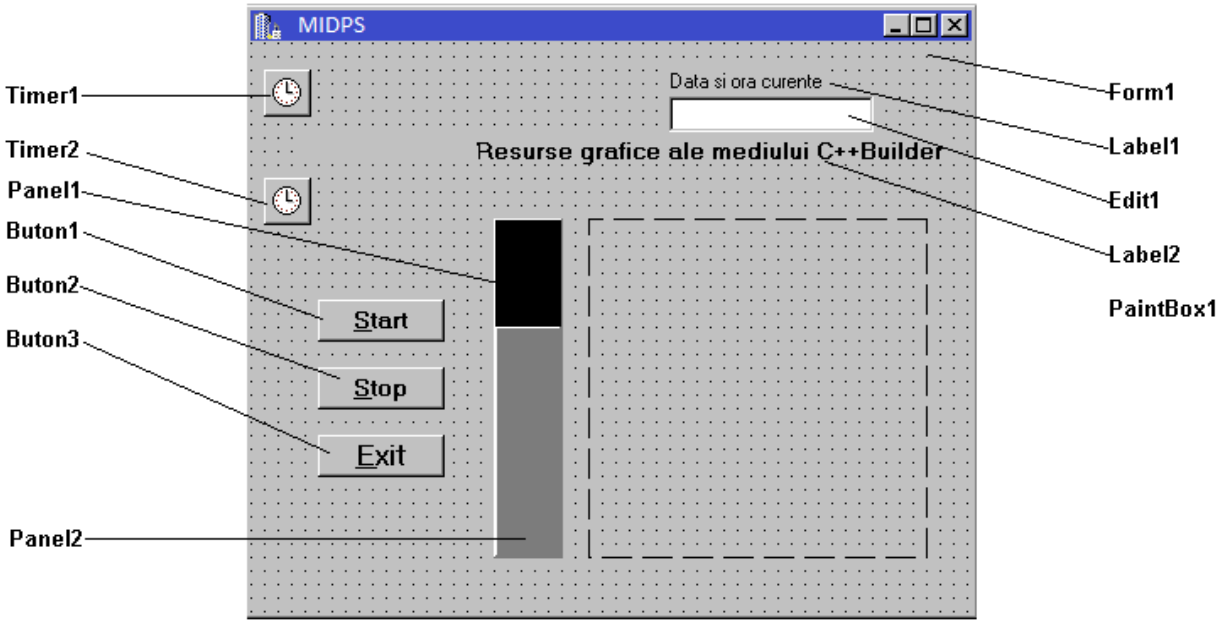


Fig 4.3 – Aplicatia 3

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

Observaţii:

- 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 afişa în cele două elemente grafice se obţine cu funcţia *random* după care numărul generat se va converti în pixeli ţinându-se cont de înălţimea comună a graficului şi bargrafului
- pentru realizarea bargrafului se vor utiliza două obiecte de tip TPanel de culori diferite care se vor suparpune;

- pentru desenarea graficului se vor utiliza funcţiile *MoveTo, LineTo* iar pentru avansul acestuia funcţia *CopyRect.*

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

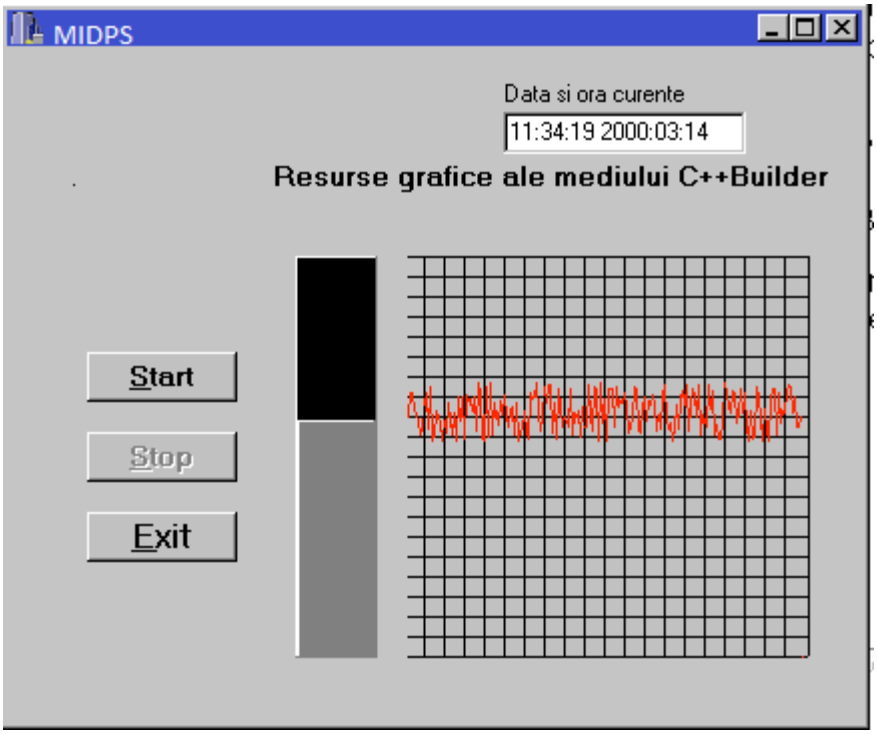
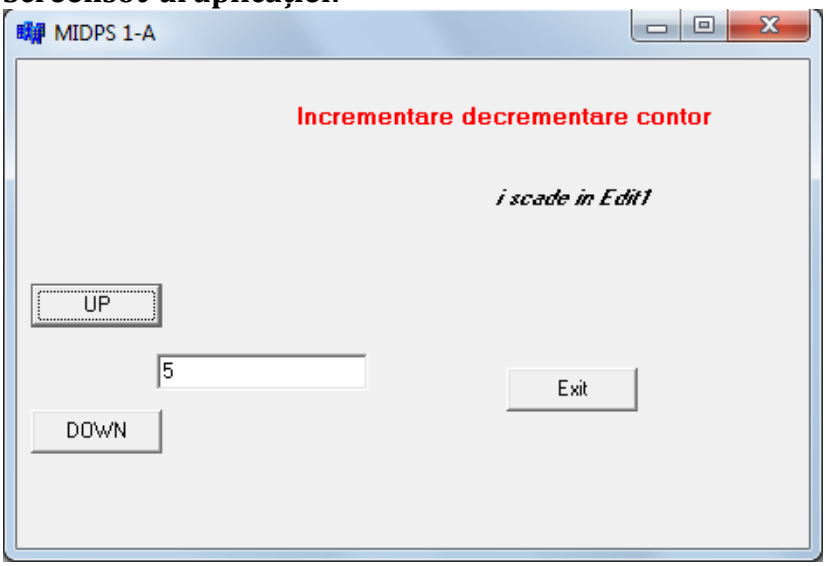


Fig.4.4- Rezultatul aplicatiei 3

Partea I

Screensot-ul aplicaţiei:



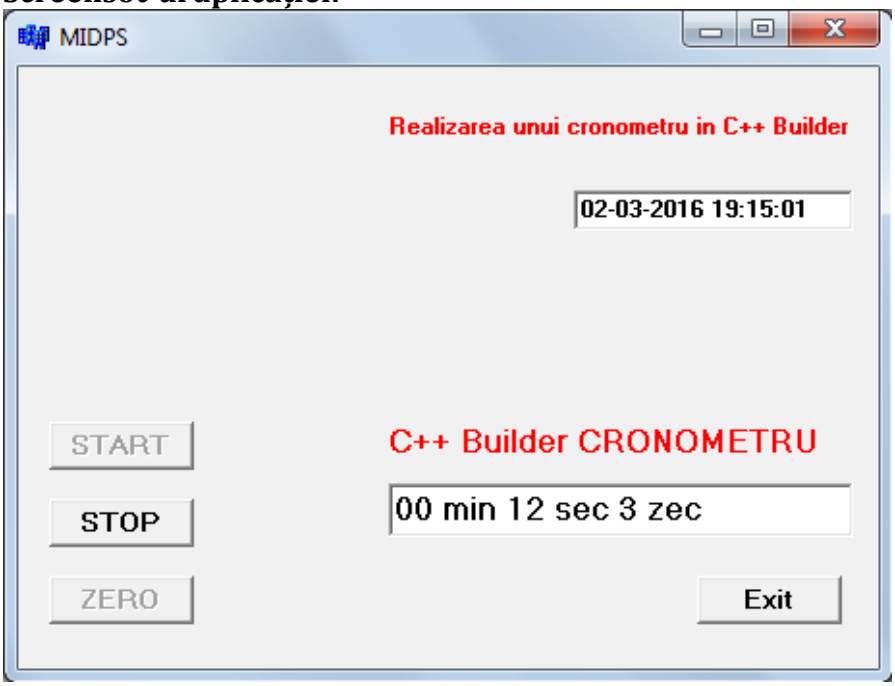
Codul sursa:

```
//---------------------------------------------------------------------------

#include <vcl.h>
#pragma hdrstop

#include "Unit1.h"
//---------------------------------------------------------------------------
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
int i;
//---------------------------------------------------------------------------
__fastcall TForm1::TForm1(TComponent* Owner)
        : TForm(Owner)
{
}
//---------------------------------------------------------------------------

void __fastcall TForm1::Button3Click(TObject *Sender)
{
Close();
}
//---------------------------------------------------------------------------

void __fastcall TForm1::FormCreate(TObject *Sender)
{
i = 0;
```

```
Edit1->Text = i;

}

//---------------------------------------------------------------------------

void __fastcall TForm1::Button1Click(TObject *Sender)

{

Edit1->Text = ++i;

}

//---------------------------------------------------------------------------

void __fastcall TForm1::Button2Click(TObject *Sender)

{

Edit1->Text = --i;

}

//---------------------------------------------------------------------------
```

Partea III

Screensot-ul aplicației:



Codul sursa:

```
//---------------------------------------------------------------------------
```

```
#include <vcl.h>
#include <stdio.h>
#pragma hdrstop
#include "Unit1.h"
//---------------------------------------------------------------------------
#pragma package(smart_init)
#pragma resource "*.dfm"
#include "dos.h"
TForm1 *Form1;
struct time t;
struct date d;
int min, sec, zec;
//---------------------------------------------------------------------------
__fastcall TForm1::TForm1(TComponent* Owner)
      : TForm(Owner)
{
}
//---------------------------------------------------------------------------

void __fastcall TForm1::Timer1Timer(TObject *Sender)
{
char buf[20];
getdate(&d);
gettime(&t);
sprintf(buf,"%02d-%02d-%4d %02d:%02d:%02d",d.da_day,d.da_mon,d.da_year,
t.ti_hour,t.ti_min,t.ti_sec);
Edit1->Text=(AnsiString)buf;
```

```
}
//---------------------------------------------------------------------------
void __fastcall TForm1::Button1Click(TObject *Sender)
{
Timer2->Enabled = true;
Button2->Enabled = true;
Button1->Enabled = false;
Button3->Enabled = false;
}
//---------------------------------------------------------------------------
void __fastcall TForm1::Timer2Timer(TObject *Sender)
{
zec += 1;
if (zec >= 10){
zec = 0;
sec ++;
}
if (sec>=60){
sec = 0;
min ++;
}
char buf[20];
sprintf(buf,"%02d min %02d sec %d zec",min,sec, zec);
Edit2->Text=(AnsiString)buf;
}
//---------------------------------------------------------------------------
void __fastcall TForm1::Button2Click(TObject *Sender)
```

```
{
Timer2->Enabled = false;
Button1->Enabled = true;
Button3->Enabled = true;
Button2->Enabled = false;
}
//---------------------------------------------------------------------------
void __fastcall TForm1::Button3Click(TObject *Sender)
{
min = sec = zec = 0;
char buf[20];
sprintf(buf,"%02d min %02d sec %d zec",min,sec, zec);
Edit2->Text=(AnsiString)buf;
}
//---------------------------------------------------------------------------
void __fastcall TForm1::Button4Click(TObject *Sender)
{
Close();
}
//---------------------------------------------------------------------------

void __fastcall TForm1::FormCreate(TObject *Sender)
{
char buf[20];
getdate(&d);
gettime(&t);
sprintf(buf,"%02d-%02d-%4d %02d:%02d:%02d",d.da_day,d.da_mon,d.da_year,
```

t.ti_hour,t.ti_min,t.ti_sec);

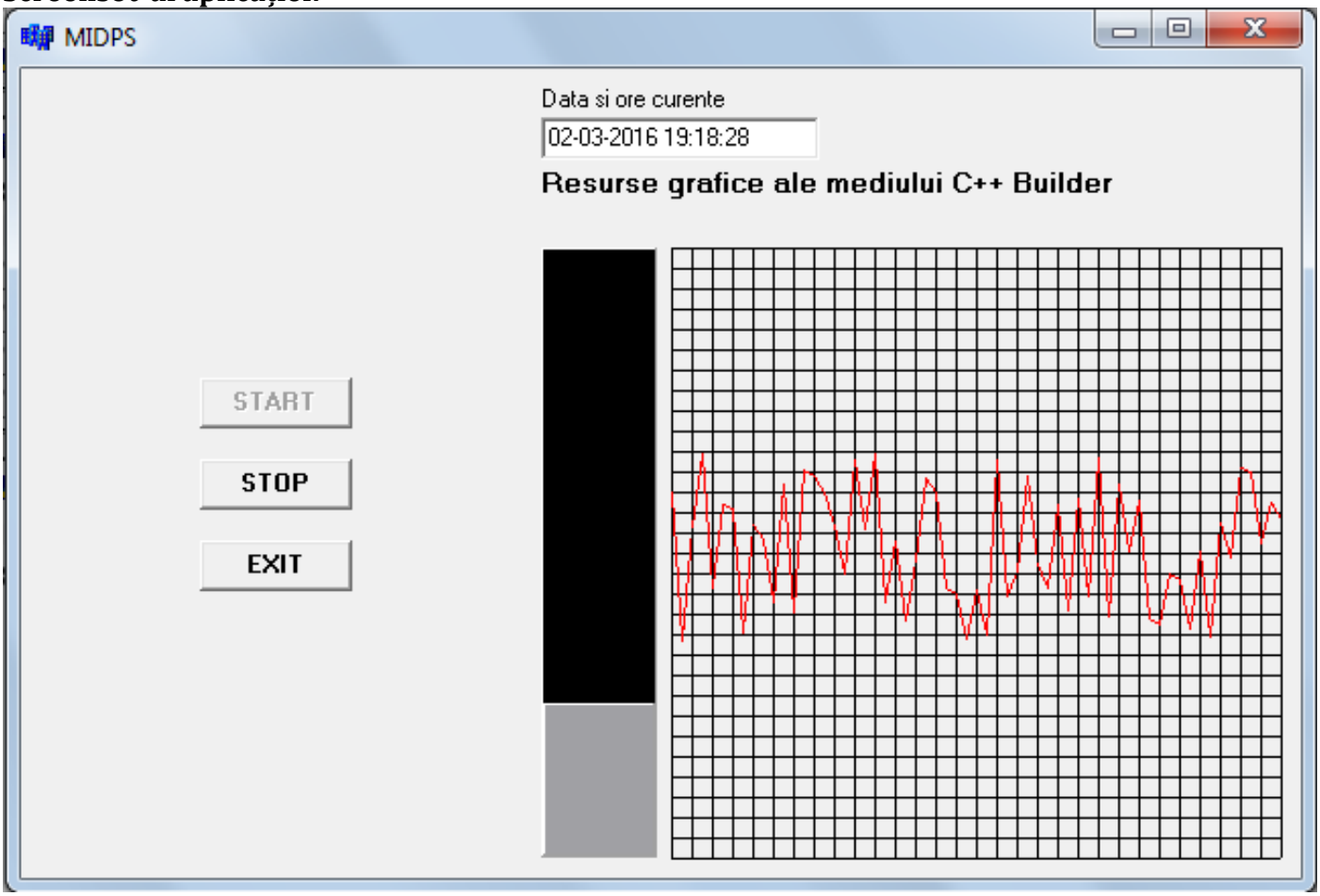
Edit1->Text=(AnsiString)buf;

}

//---

Partea III

Screensot-ul aplicației:



Codul sursa:

//---

#include <vcl.h>

#pragma hdrstop

#include <stdio.h>

#include "Unit1.h"

#include "dos.h"

```
//---------------------------------------------------------------------------
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
struct time t;
struct date d;
TRect rect;
int nextY, next2Y, nextP;
bool first;
//---------------------------------------------------------------------------
__fastcall TForm1::TForm1(TComponent* Owner)
        : TForm(Owner)
{
}
//---------------------------------------------------------------------------

void __fastcall TForm1::Button3Click(TObject *Sender)
{
Close();
}
//---------------------------------------------------------------------------

void __fastcall TForm1::Button1Click(TObject *Sender)
{
Button1->Enabled = false;
Button2->Enabled = true;
Timer2->Enabled = true;
```

```
first = true;
rect.Left = 0;
rect.Right = 300;
rect.top = 0;
rect.Bottom = 300;
}
//---------------------------------------------------------------------------

void __fastcall TForm1::Button2Click(TObject *Sender)
{
Button1->Enabled = true;
Button2->Enabled = false;
Timer2->Enabled = false;
}
//---------------------------------------------------------------------------
void __fastcall TForm1::Timer1Timer(TObject *Sender)
{
char buf[20];
getdate(&d);
gettime(&t);
sprintf(buf,"%02d-%02d-%4d %02d:%02d:%02d",d.da_day,d.da_mon,d.da_year,
t.ti_hour,t.ti_min,t.ti_sec);
Edit1->Text=(AnsiString)buf;
}
//---------------------------------------------------------------------------
void cells(){
```

```
}
void __fastcall TForm1::Timer2Timer(TObject *Sender)
{
if (first){
nextP =  abs(rand()%100);
nextY = abs(rand()%100);
PaintBox1->Canvas->Pen->Color = clBlack;
for (int i =0; i<31; i++){
PaintBox1->Canvas->MoveTo(0,i*10);
PaintBox1->Canvas->LineTo(300,i*10);
}
for (int i =0; i<31; i++){
PaintBox1->Canvas->MoveTo(i*10,0);
PaintBox1->Canvas->LineTo(i*10,300);
}
first = false;
}
TRect rect2;
rect2.left = 0;
rect2.right = 300;
rect2.top = 0;
rect2.bottom = 300;
rect.left = -10;
rect.right = 290;
TRect rect3;
rect3.left = 290; rect3.right = 300; rect3.top = 0; rect3.bottom = 300;
PaintBox1->Canvas->FillRect(rect3);
```

```
PaintBox1->Canvas->MoveTo(290,0);
PaintBox1->Canvas->LineTo(290,300);
for (int i =0; i<31; i++){
PaintBox1->Canvas->MoveTo(290,i*10);
PaintBox1->Canvas->LineTo(300,i*10);
}
PaintBox1->Canvas->MoveTo(290,nextY+100);
PaintBox1->Canvas->Pen->Color = clRed;
nextY = next2Y;
PaintBox1->Canvas->LineTo(295,100+nextP);

PaintBox1->Canvas->LineTo(300,100+nextY);
PaintBox1->Canvas->CopyRect(rect,PaintBox1->Canvas, rect2);
PaintBox1->Canvas->Pen->Color = clBlack;

rect3.left = 290; rect3.right = 300; rect3.top = 0; rect3.bottom = 300;
PaintBox1->Canvas->FillRect(rect3);
PaintBox1->Canvas->MoveTo(290,0);
PaintBox1->Canvas->LineTo(290,300);

for (int i =0; i<31; i++){
PaintBox1->Canvas->MoveTo(290,i*10);
PaintBox1->Canvas->LineTo(300,i*10);
}

nextP =  abs(rand()%100);
PaintBox1->Canvas->Pen->Color = clRed;
```

```
next2Y = abs(rand()%100);

PaintBox1->Canvas->MoveTo(290,nextY+100);

PaintBox1->Canvas->LineTo(295,100+nextP);

PaintBox1->Canvas->LineTo(300,100+next2Y);

PaintBox1->Canvas->Pen->Color = clBlack;

PaintBox1->Canvas->MoveTo(300,0);

PaintBox1->Canvas->LineTo(300,300);

Panel2->Height = Panel1->Height*(nextP/100.0);

//PaintBox1->Canvas->FillRect(rect);

}

//---------------------------------------------------------------------------


void __fastcall TForm1::FormCreate(TObject *Sender)

{

char buf[20];

getdate(&d);

gettime(&t);

sprintf(buf,"%02d-%02d-%4d %02d:%02d:%02d",d.da_day,d.da_mon,d.da_year,

t.ti_hour,t.ti_min,t.ti_sec);

Edit1->Text=(AnsiString)buf;

}

//---------------------------------------------------------------------------
```

Concluziile

În cadrul acestei lucrări de laborator am studiat principiile şi modul de utilizare a celor mai importante componente ale mediului integrat C++ Builder aşa ca TButton, TEdit, TTimer, TPaintBox şi etc. Bazând pe cunoştinţele obţinute am reuşit sa realizez trei aplicaţii Windows cu interfaţa grafica, funcţionalul cărora include nu doar interacţiunea grafică dintre utilizator şi aplicaţia cît şi gestionarea resursei de timp şi afişarea grafică

a informației (prin intermediul diagramelor și bargrafelor). Experiența ți cunoștințele obținute pe parcursul îndeplinirii lucrării de laborator vor fi utile în viitor și pot fi aplicate pentru realizarea proiectelor diferite.