Software-Based Scientific Calculator Using Visual Basic.Net

Isizoh A. N.[1]

Dept. of Electronic and Computer Engineering,
Nnamdi Azikiwe University, Awka, Nigeria.

Anazia A.E.[2]

Dept. of Electrical Engineering,
Nnamdi Azikiwe University, Awka, Nigeria.

Okide S. O.[3]

Dept. of Computer Science,
Nnamdi Azikiwe University, Awka, Nigeria.

Okwaraoka C.A.P.[4]

Dept. of Electrical/Electronics Engineering,
Federal Polytechnic , Nekede, Imo State, Nigeria

Abstract

This Paper presents the development of a software-based scientific calculator using Visual Basic.Net (VB.Net). Visual Studio 2008 package was used and it contains the VB.Net framework. The calculator design was done in the visual "FORM" environment of the VB .Net, while the control program was written in "Form1.vb" by double clicking the buttons on the scientific calculator. The program was run and it was error-free. The Scientific calculator was implemented in a Windows Operating System (OS) because Visual Basic .Net is a Windows-based software.

Keywords: Visual Basic .NET, scientific calculator, label and toolbox.

1. Introduction

A Calculator is a small (often pocket-sized), usually inexpensive electronic device used to perform the basic operations of arithmetic. Modern calculators are more portable than most computers, though most Palmtops are comparable in size to handheld calculators [1].

This software-based scientific calculator was developed using Visual Studio. Visual Studio is an Integrated Development Environment (IDE), providing a single interface for any number of languages, including Visual Basic. The purpose of an IDE is to combine the editing, debugging, and compiling components of software development into a single interface for a programmer [2].

Visual Basic (VB) starts from Visual Basic 1.0 up to Visual basic.Net (VB.Net). The .Net is from VB 7.0 upwards, and represents the entire range of VB technology and concepts that form a platform on which you can develop applications [3].

Visual Studio 2008 package was used for this work. It contains the VB.Net framework which was used to develop the software-based scientific calculator. The calculator was developed in the "FORM" environment of the Visual Studio; while the control program was written by double clicking the buttons in the "Form 1.vb.

There are two major classes of operating tools for software development using Visual Basic .Net [4]. They are the Toolbox and the Properties. Toolbox is grouped into classes, namely; All Windows Form, Common Controls, Containers, Menus & Toolbars, Data, Components, Printing, Dialogs, WPF Interoperability, Reporting, and General. While Properties have Accessibility, Appearance, Behavior, Data, Design, Focus, Layout, Misc, and Window Style [5].

2. The System Development

To develop this calculator, open Visual Studio 2008, and create the new design in the 'FORM' environment. Note the needed information, like the names that will be given to the various buttons and the Label for information display. In order to open the environment for system development, the following steps are to be taken:

1) Launch your Visual Basic .NET or Visual Studio software. After loading, the software will open.

2) At the bottom of the screen, there are two buttons: "Open project" and "Create project". To get started, click on the "Create project" button.

The Visual Basic.NET design time environment will open. It will look like figure 1.

One thing with VB.Net is that when using it, the work must be saved first before designing the program.

3) Type the file name which is used for saving the work, and click on ok. The "FORM" environment will come out. This "FORM" is the design interface.
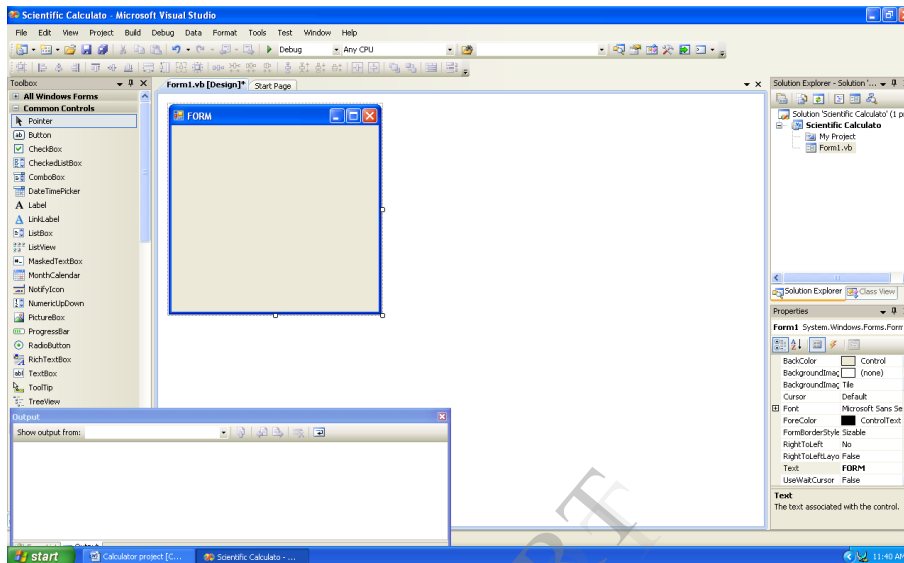

Fig. 1. A Visual Studio design interface containing the "FORM"

4) Start the proper design by positioning the Label for display and the buttons as keys. These are known as Controls, and are kept in the Toolbox for ease of use. The Toolbox can be found on the left hand side of the screen.

To display all the tools, move the mouse over the Toolbox icon. There are seven categories of tools available. The Toolbox to be working with first is the Common Controls Toolbox. To see the tools, click on the plus symbol next to Common Controls.

2.1. Common Controls in VB.Net Used for the System development

Button: This is used to raise an event when a user clicks on it. It is for writing most programs. Here, it was used in the design of the 'number buttons' like 1, 2, 3, 4, etc and 'mathematical functions buttons' like /, -, +, x, etc.

TextBox: It enables the user to enter text, and provides multiline editing and password character masking [6]. Cursor can be placed here. It was not used in this work.

Label: It provides run-time information or descriptive text for a control. It is used for writing or labelling items in the form environment. If this control is used, data on it cannot be tampered because it has no cursor inside it, unlike the Textbox [7]. It was used in this work for designing the 'screen area' for the display of the numbers pressed and the mathematical operation together with the result of any calculation.

2.2. Design procedure for the Scientific calculator

1) Drag the needed buttons from the ToolBox and drop them inside the "FORM".

2) Copy and paste the buttons to the corresponding number of numeric buttons and Math functions buttons that are needed.

On the right hand side of the design environment, there are Properties box. This is the area that controls the design "FORM". The Properties Box of the Visual Studio is shown in figure 2.

Click on each button and at the Properties box, change the names of the buttons in the "Text" Properties to 0,1,2,3,4,5,6,7,8,9,sin,/,+,-. log, cos, tan, sqr, etc.
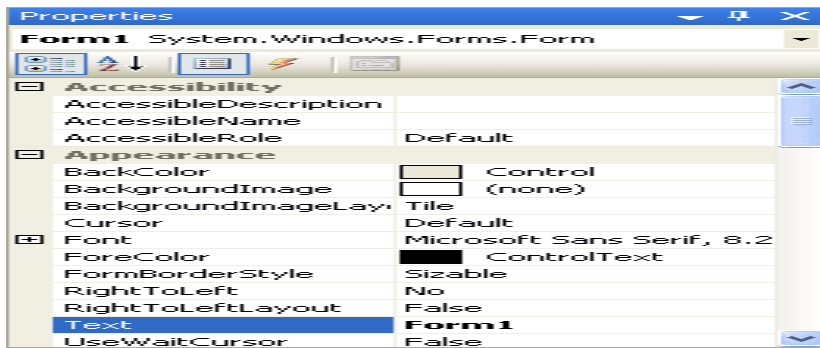
Fig. 2. The Properties box of the Visual Studio

3) Also click on the Math function buttons and likewise change their names at the "Text" Properties to +, -, *, /, etc.

4) Drag "Label" from the Toolbox and it drop inside the FORM. This is used for the display of information.

5) When any of the buttons is double clicked, it takes one to the code area from where one writes the code that defines the function of that button.

The layout design is called the form 1.vb (design), shown in figure 3.
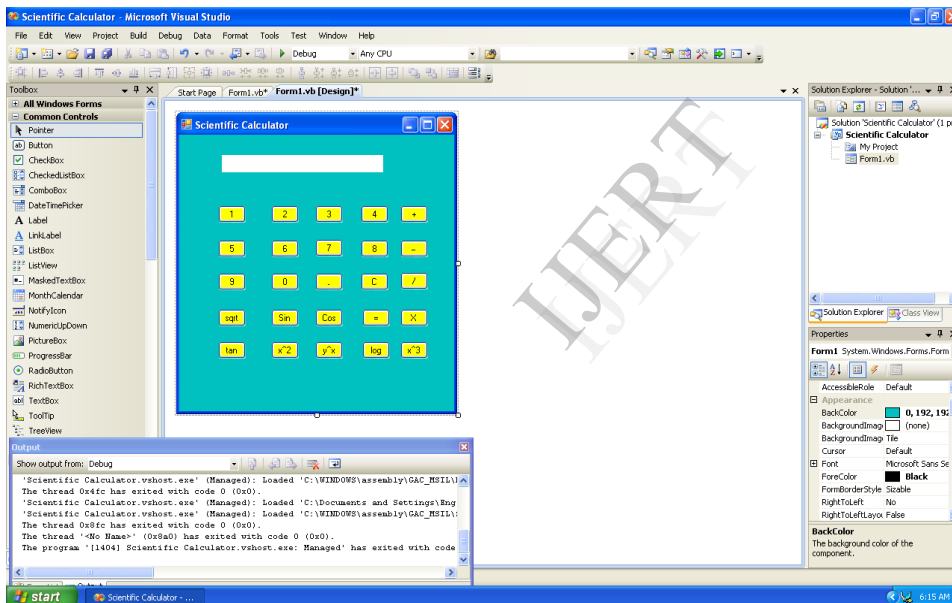

Fig. 3. The designed view of the scientific calculator

Double clicking any of the buttons or keys, opens the code window for programming.

3. System Implementation

Implementation of this visual calculator is not difficult. This system must be implemented in a computer with Windows Operating System, because VB.Net is a Windows-based software.

Just install this Scientific calculator software in a computer with Windows Operating System. Run the program in the code environment by debugging. When this is done, the developed software will be displayed. It works like ordinary physical calculator used for basic calculations. Punch the number buttons you want and the mathematical function you want to carry out, punch the "=" button to give you the result. This result will be displayed on the "Display" label, as shown in figure 4.
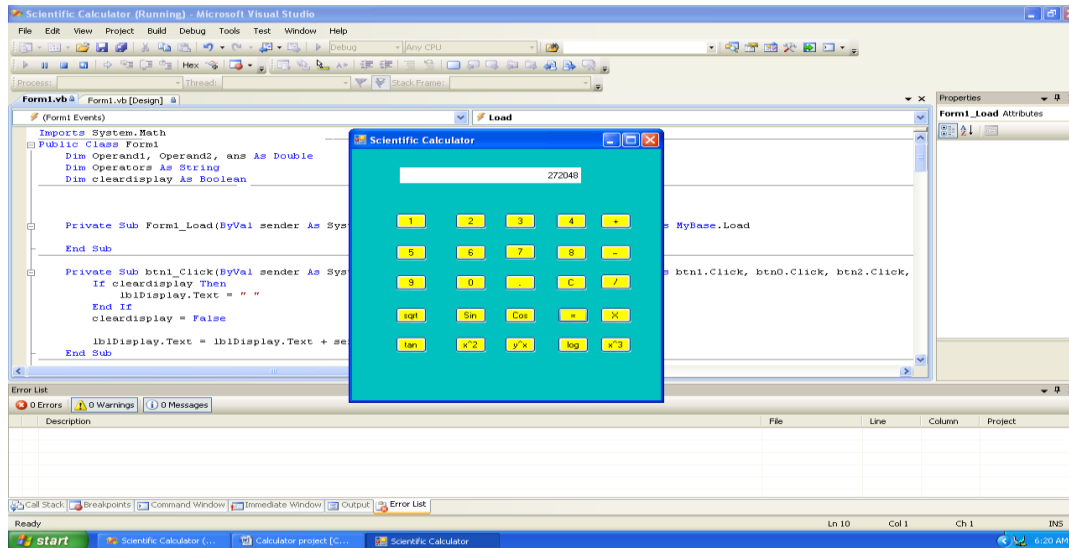
Fig. 4. The software-based Scientific Calculator

4. Conclusion

With the introduction of the .Net framework and VB.Net language, software development has been greatly facilitated with easy use of objects (forms, buttons, boxes, etc) which can be inserted and then their functions specified by coding.

This paper has succeeded in giving an insight into the Microsoft's .Net framework and that of Visual Basic Programming language; and the use of the language in the software development of a Software-based Scientific calculator. Other software-based calculating systems like Visual loan calculator, weather forecast analysing software, etc, can as well be designed using VB.Net.

5. The Developed Software (Program)

```
Imports System.Math
Public Class Form1
    Dim Operand1, Operand2, ans As Double
    Dim Operators As String
    Dim cleardisplay As Boolean
    Private Sub Form1_Load(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
    End Sub
    Private Sub btn1_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles btn1.Click, btn0.Click, btn2.Click,
btn3.Click, btn4.Click, btn5.Click, btn6.Click,
btn7.Click, btn8.Click, btn9.Click
        If cleardisplay Then
            lblDisplay.Text = " "
        End If
        cleardisplay = False
        lblDisplay.Text = lblDisplay.Text + sender.text
    End Sub
    Private Sub btnCancel_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles btnCancel.Click
        lblDisplay.Text = " "
    End Sub
    Private Sub btnplus_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles btnplus.Click
        Operand1 = Val(lblDisplay.Text)
        Operators = "+"
        cleardisplay = True
    End Sub
    Private Sub btnminus_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles btnminus.Click
        Operand1 = Val(lblDisplay.Text)
        Operators = "-"
        cleardisplay = True
    End Sub
    Private Sub btndiv_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles btndiv.Click
        Operand1 = Val(lblDisplay.Text)
        Operators = "/"
        cleardisplay = True
    End Sub
    Private Sub btnmul_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles btnmul.Click
        Operand1 = Val(lblDisplay.Text)
```

```vb
      Operators = "*"
      cleardisplay = True
   End Sub
   Private Sub btnequal_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles btnequal.Click
      Operand2 = Val(lblDisplay.Text)
      Select Case Operators
        Case "+"
          ans = Operand1 + Operand2
        Case "-"
          ans = Operand1 - Operand2
        Case "/"
          ans = Operand1 / Operand2
        Case "*"
          ans = Operand1 * Operand2
        Case "^"
          ans = Operand1 ^ Operand2
      End Select
      lblDisplay.Text = ans
   End Sub
   Private Sub btnsqr_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles btnsqr.Click
      Operand1 = Val(lblDisplay.Text)
      ans = Operand1 ^ 2
      lblDisplay.Text = ans
   End Sub
   Private Sub btnpower_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles btnpower.Click
      Operand1 = Val(lblDisplay.Text)
      Operators = "^"
      cleardisplay = True
   End Sub
   Private Sub btnqube_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles btnqube.Click
      Operand1 = Val(lblDisplay.Text)
      ans = Operand1 ^ 3
      lblDisplay.Text = ans
   End Sub
   Private Sub btnsqrt_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles btnsqrt.Click
      Operand1 = Val(lblDisplay.Text)
      ans = Operand1 ^ 0.5
      lblDisplay.Text = ans
   End Sub
   Private Sub btndot_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles btndot.Click
      If lblDisplay.Text.IndexOf(".") > 0 Then
        Exit Sub
      Else
        lblDisplay.Text    =    lblDisplay.Text    +
sender.text
      End If
   End Sub
   Private Sub btnsin_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles btnsin.Click
      Operand1 = Val(lblDisplay.Text)
      ans = Math.Sin(Operand1)
      lblDisplay.Text = ans
   End Sub
   Private Sub btnCos_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles btnCos.Click
      Operand1 = Val(lblDisplay.Text)
      ans = Math.Cos(Operand1)
      lblDisplay.Text = ans
   End Sub
   Private Sub btnTan_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles btnTan.Click
      Operand1 = Val(lblDisplay.Text)
      ans = Math.Tan(Operand1)
      lblDisplay.Text = ans
   End Sub
   Private Sub btnlog_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles btnlog.Click
      Operand1 = Val(lblDisplay.Text)
      ans = Math.Log10(Operand1)
      lblDisplay.Text = ans
   End Sub
End Class
```

6. References

[1] Watkins Perkins, "Programming in the .NET Environment", Addison Wesley, London, 2008.

[2] Stroo Eric and Stuart J. Stuple, "Microsoft Visual Basic Programmers Guide", Microsoft Press, USA, 2009.

[3] Hutson Michael, "Introduction to Visual Basic .Net", John Wiley and Sons, New York, 2008.

[4] C. S. French, "Computer Science", Ashford Colour Press, Gosport, Hants, Great Britain, 2009,

[5] Wright Peter, "Beginning Visual Basic .Net", Wrox Press Ltd, Canada, 2009.

[6] Ochi – Okorie A. S., "Computer Fundamentals: Introduction and Utilization", Solid Rock Computer Press, Enugu, Nigeria, 2008.

[7] Craig Utley, "A Programmer's Introduction to Visual Basic .Net", Sams Publishing, Indianapolis, USA, 2007.