IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 4 Issue 6, June 2017
ISSN (Online) 2348 – 7968 | Impact Factor (2016) – 5.264
www.ijiset.com

Computer-Aided Design of a Flat Floor Timber Formwork

Okere Chinenye Elizabeth

Civil Engineering department, Federal University of Technology, Owerri
P.M.B. 1526 Owerri.

Abstract
The traditional method of design which involves manual calculations is energy wasting, time consuming and not error free. A user friendly computer aided design program coded in VISUAL Basic language for the design of flat floor formwork is presented which will minimize some of these limitations and curb the complexities encountered in the manual design of formwork. There is no significant difference between the results obtained from manual design and that obtained from the computer program. The use of this program will lead to maximum functionality and productivity level of the construction industry.
Keywords*: Formwork, Timber, Computer program, Visual Basic, Computer-aided design.*

1. Introduction

Formwork is the term given to either temporary or permanent moulds into which concrete elements or similar materials are poured. It serves as a mould to produce concrete elements having a desired size and configuration [1]. It is a structure built to support and confine concrete until it hardens. It is usually erected for this purpose and then removed after the element has cured to satisfactory strength. In some cases, the forms may be left in the place to become part of the permanent structure.

Various materials can be used as forms but timber is the most common material because of its availability. Traditional timber formwork is easy to produce and erect but time consuming for larger structures. It is the most flexible type of formwork and is required for practically all jobs of formwork. Timber forms are economical for small construction jobs. They have good thermal insulation which makes them useful to be used in colder regions. They can easily be made into ant shape or size.

Several factors are put into consideration in the design of forms in order to avoid structural failures and achieve a balance of certain requirements such as containment, strength, resistance to leakage, accuracy, ease of handling, finish and reuse potential, access for concrete and economy. A rational approach to formwork design based on simplified assumptions and approximate beam formulas is sufficiently adequate when designing formwork subjected to light loads [2]. A detailed structural design may be required for extremely heavy loading or where there is unusual danger to life or property. A simplified approach for the rational design of formwork members is usually justified [3].

The traditional method of design which involves manual calculations is energy wasting, time consuming and not error free. A computer aided design program is proposed which will minimize some of these limitations and curb the complexities encountered in the manual design of formwork. A modeling approach is also proposed by Christian [4]. This will lead to maximum functionality and productivity level of the construction industry. This work presents a user friendly program coded in VISUAL Basic language for the design of flat floor formwork. The program capabilities are illustrated by means of a case study.

2. Methodology

A typical example of the design of forms needed to support a flat floor (using conventional construction framing members and shoring) is presented. A manual design approach was adopted and a computer program was written based on the step by step description of how the problem was solved. The developed program is appended.

2.1 Manual Calculations

<u>Design Data</u>
Floor thickness = 150mm
Bay dimension = 4m x 4m
Live load = 25KN/m^2
<u>Sheathing</u>
Allowable stress, f = 8.5N/mm^2
Horizontal shear, H = 1.25N/mm^2
Modulus of Elasticity, E = 10,500N/mm^2
Unit weight of concrete = 24Kn/m^3
<u>Joists</u>
Use 50 x 100mm lumber
H = 1.25N/mm^2
F = 10N/mm^2
E = 11,000N/mm^2

Stringers
Use 50 x 150mm lumber
All other data are as given in joist above

Design Load
Characteristic dead load = 150/1000 x 24 = $3.6KN/m^2$
Design load = dead load + live load
= 3.6 + 2.5 = $28.6KN/m^2$

Sheathing design
Thickness of sheathing = 25mm (Nominal size)

Actual size, h = ¾ x Nominal size = 18.75mm

(using b = 1000mm)

Section modulus, $Z = bh^2/6$ (1)

= $1000 * 18.75^2/6 = 5.86 \times 10^4 mm^3$

Second moment of inertia, $I = bh^3/12$ (2)

= $1000 * 18.75^3/12 = 5.49 \times 10^5\ mm^4$

Check for bending

$L = 3.16 * \sqrt{fz/w}$ (3)

= $3.16 * \sqrt{8.5 * 5.85 * 10^4/28.6}$ = 417.03mm

Check for shear

L = Hbh/0.9w (4)

= 1.25 * 1000 * 18.75/0.9*28.6 = 910.55mm

Check for deflection

$L = 0.74 * \sqrt[3]{EI/w}$ (5)

= $0.74 * \sqrt[3]{10500 * 5.49 * 10^5/28.6}$ = 433.87mm

Bending hereby governs the check.

Therefore provide and select span reinforcement @ 410mm spacing (which is less than the minimum L values)

Spaces = longest bay dimension/ provided span length = 4000/410 = 9.76 ≈ 10 spaces

No. of Joists = spaces + 1 = 10 + 1 = 11 Joists

Therefore provide 11 joists @ 410mm spacing

Joist Design
Joist dimensions are as follows:

h = 50mm, b = 100mm

Section modulus, $Z = bh^2/6 = 1000* 50^2/6 = 4.17 \times 10^4 mm^3$

Second moment of inertia, $I = bh^3/12 = 100 * 50^3/12$ = $1.0\ 4 \times 10^6\ mm^4$

Equivalent uniform load on each joist = provided span length * design load/1000

= 410 * 28.6/1000 = $11.73KN/m^2$

Using Eqs (3), (4) and (5), the following checks were made for bending, shear and deflection respectively for the joiss.

Check for bending

$L = 3.16 * \sqrt{fz/w} = 3.16 * \sqrt{10 * 4.17 * 10^4/11.73}$ = 595.81mm

Check for shear

L = Hbh/0.9w = 1.25 * 100 * 50/0.9* 11.73 = 592.02mm

Check for deflection

$L = 0.74 * \sqrt[3]{EI/w} = 0.74 * \sqrt[3]{11000 * 1.04 * 10^6/11.73}$ = 733.85mm

Shear governs the check.

Therefore provide and select span reinforcement @ 580mm spacing

Spaces = longest bay dimension/ provided span length = 4000/580= 6.9 ≈ 7 spaces

No. of Joists = spaces + 1 = 7 + 1 = 8 stringers

Therefore provide 8 stringers @ 580mm spacing.

Stringer design
Stringer dimensions are as follows:

h = 50mm, b = 150mm

Section modulus, Z = $bh^2/6 = 150* 50^2/6 = 6.25 \times 10^4 mm^3$

Second moment of inertia, I = $bh^3/12 = 15 * 50^3/12 = 1.56 \times 10^6\ mm^4$

Equivalent uniform load on each joist = provided span length * design load/1000

= 580 * 28.6/1000 = 16.59KN/m^2

Using Eqs (3), (4) and (5), the following checks were made for bending, shear and deflection respectively for the stringer.

Check for bending

L = 3.16 * √fz/w = 3.16 * √10 * 6.25*10^4/16.59 = 613.34mm

Check for shear

L = Hbh/0.9w = 1.25 * 150 * 50/0.9* 16.59 = 627.89mm

Check for deflection

L = 0.74 * ^{3}√EI/w = 0.74 * ^{3}√ 11000 * 1.56* 10^6/16.59 = 748.38mm

Bending governs the check.

Therefore provide and select span reinforcement @ 600mm spacing

Spaces = longest bay dimension/ provided span length = 4000/6000= 6.67≈ 7 spaces

No. of shores = spaces + 1 = 7 + 1 = 8 stringers

Therefore provide 8 stringers @ 600mm spacing.

2.2 Program Development

The basic steps involved in developing the program include writing algorithm, declaring and using variables and using data types.

The algorithm, which is a step-by-step description of how the problem was going to be solved, was first written. The variables were declared and used. When the variable is first declared, it is given a null value until a value has been assigned. When a variable is declared as 'dim' style, it tells the computer that the variable is meant just for the class in which it was declared but when a variable is declared with the 'public' style, it tells the program that the variable can be accessed from any class within the program [5]. With the use of variables, the data types that should be stored in them were also specified. When the variable is expected to hold a whole number, it was declared as an 'integer' but if expected to hold a decimal number it was declared as 'double'. Where the variable is expected to hold a sequence of characters or text, it was declared as 'string'.

Different forms were added to the program. Form 1 for the design of sheathing, form 2 for the design of joists, form 3 for the design of stringers and another form for the result sheet.

3. Results And Discussion

The summary of the results from the manual design computations and computer program outputs are presented on Table 1.

Table 1: Comparison of results obtained from manual and computer computations

	MANUAL DESIGN	COMPUTER DESIGN	PERCENTAGE DIFFERENCE
SHEATING DESIGN			
Section modulus, Z (mm^3)	$5.86 * 10^4$	58,593.75	0.012
Second moment of inertia, I (mm^4)	$5.49 * 10^5$	549,316.41	0.057
Design load, w (KN/m^2)	28.6	28.6	0
Spacing for bending (mm)	417.03	417	0.007
Spacing for shear (mm)	910.55	910.55	0
Spacing for deflection (mm)	433.87	433.96	0.021
No of joists	11	11	0
Spacing of joists (mm)	410	410	0

JOIST DESIGN			
Section modulus, Z (mm^3)	$4.17 * 10^4$	41,666.67	0.080
Second moment of inertia, I (mm^4)	$1.04 * 10^6$	1,041,666.67	0.160
Design load, w (KN/m^2)	11.73	11.726	0.034
Spacing for bending (mm)	595.81	595.67	0.024
Spacing for shear (mm)	592.02	592.23	0.035
Spacing for deflection (mm)	733.85	734.33	0.065
No of stringers	8	8	0
Spacing of stringers (mm)	580	580	0
STRINGER DESIGN			
Section modulus, Z (mm^3)	$6.25 * 10^4$	62500	0
Second moment of inertia, I (mm^4)	$1.56 * 10^6$	1,562,500	0.160
Design load, w (KN/m^2)	16.59	16.99	0.4
Spacing for bending (mm)	613.34	613.34	0
Spacing for shear (mm)	627.89	627.89	0
Spacing for deflection (mm)	748.38	748.78	0.053
No of shores	8	8	0
Spacing of shores (mm)	600	600	0

A comparison of results was made. As can be observed from the table, there is no significant difference between the results obtained from manual design and that obtained from the computer program. The percentage difference ranges from 0 to 0.065 which is very insignificant.

4. Conclusion

A computer program for the design of timber formwork to support a flat floor has been developed using visual basic language. There is no significant difference between the results obtained from manual design and that obtained from the computer program. The program is user friendly, easy and inexpensive to use and yields quick and accurate results. It also limits the stress and errors which might be encountered in the manual design leading to maximum functionality. This leads to increased productivity level of the construction industry

Appendix

```
Public Class Form1

Dim breath As Double = 1000
Dim liveload As Double
Dim slabThickness As Double
Dim bending As Double = 1000
Dim shear As Double
Dim deflection As Double
Public norminalSize As Double
Public lumberSection As Double
Public moment of inertia As Double
Public sectionModulus As Double
Public deadload As Double
Public designload As Double
Public allowableStress As Double
Public  horizontalShear As Double
Public modulus ofElasticity As Double
Public lowest As String
Public s As Double
Dim noOfJoists As Double
Dim spaces As Double
Dim longerSpan As Double
Dim shorterSpan As Double
Dim spacing As Double

Private Sub Form1_Load(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
Panel2.Visible = False
Button2.Visible = false
Breath = 1000
End Sub

Private Sub Button1_Click(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles  Button1.Click
        sheatingThickness = Val(tsheating.Text)
        liveload = Val (1load.Text)
        slabThickness = Val(sthickness.Text)
        allowableStress = Val(tastress.Text)
        horizontal Shear = Val(hshear.Text)
        modulusOfElasticity = Val(melasticity.Text)
        longerSpan = Val(lspan.Text)
        shorterSpan = Val(sspan.Text)
        norminalSize = (0.75 * sheatingThickness)
        lumberSection = (breath * norminalSize)
```

```
        momentOfInertia = Math.Round (( breadth *
(Math.Pow(norminalSize, 3))) / 12), 2)
        sectionModulus =  (( breadth *
(Math.Pow(norminalSize, 2))) / 6)
        sectionModulus =math.Round
(sectionModulus, 2)
        deadLoad = ((slabThickness / 1000) * 24)
        designLoad = Math.Round ((liveLoad +
deadLoad) , 2)
        Form2.designLoads = designLoad
        Form3.designLoads = designLoad
        bending = Math.Round ((3.16 *
(Math.Sqrt((allowableStress *
        sectionModulus) / (designload)))), 2)
        shear = Math.Round((horizontalShear *
breath * norminalSize) / (0.9 * designLoad), 2)
        deflection = Math.Round((0.74 *
(Math.Pow((modulusOfElasticity *
momentOfInertia)
        / (designLoad). (1 / 3)))), 2)

        If bending < shear And bending < deflection
Then
           Lowest = "Bending"
        s = Math.Floor (bending)
        ElseIf shear < bending And shear <
deflection Then
           Lowest = "Shear"
           s = Math.Floor (shear)
        ElseIf deflection < shear And deflection <
bending Then
           Lowest = "Deflection"
          s = math.Floor (deflection)
        End If
        Dim t As Integer = Math.Round (s / 10) * 10
        t  = (t -10)
        'Form2.Lprovided = t
        Spacing = t
        Spaces = longerSpan / s
        Spaces = math.Ceiling (spaces)
        noOfJoists = spaces + 1

        populateResult ()
        End Sub

        Public Sub populateResult()
        Result.Visible = True
        Panel2.Visible = True
        Panel1.Visible = False
        Button2.Visible = True
        result.Text =  "SECTION MODULUD =  "
& sectionModulus  & "mm3   " & vbCrLf & vbCrLf
        result.Text = result.Text  + "DESIGN
LOAD =  " & designLoad &  "   KN/m2 " & vbCrLf
& vbCrLf
        result.Text = result.Text  + "BENDING =  "
& bending &  "   mm " & vbCrLf & vbCrLf
        result.Text = result.Text  + "SHEAR =  " &
shear &  "   mm " & vbCrLf & vbCrLf
        result.Text = result.Text  + "DEFLECTION
=  " & deflection &  "   mm " & vbCrLf & vbCrLf
result.Text = result.Text  + "PROVIDE=  " &
noOfjoists &  "   JOISTS AT " & spacing "  mm
SPACING"

        Form5.noOfjoists  = noOfJoists
        Form5.joistspacing  = spacing
        Form5.designLoad  = designLoad
        End Sub

        Private Sub Panel1_Paint(ByVal sender As
System.Object, ByVal e As
        System.Windows.Forms.PaintEventArgs)
Handles Panel1.Paint
        End Sub

        Private Sub Panel2_Paint(ByVal sender As
System.Object, ByVal   e  As
        System.Windows.Forms.PaintEventArgs)
        End Sub

        Private Sub Button2Click(ByVal sender As
System.Object, ByVal e As
        System.EventArgs) Handles Button2.Click
          Form2.Show()
        End Sub

        Private Sub Panel2_Paint_1 (ByVal sender
As System.Object, ByVal e As
        System.Windows.Forms.PaintEventArgs)
Handles Panel2.Paint
        End Sub
        End Class


Public Class Form2

Dim longerSpan As Double
Dim shorterSpan As Double
Public lprovided As Double
Public spaces As Double
Public noOfStringers As Double
Public designload As Double
Public result As String
Dim breath As Double
Dim heights As Double
Dim lumberSection As Double
Dim momentOfInertiaAs Double
Dim sectionModulus As Double
Dim JoistDesignLoad As Double
```

```
Dim allowableStress As Double
Dim horizontalShear As Double
Dim melasticity As Double
Dim bending As Double
Dim shear As Double
Dim deflection As Double
Dim joistThickness As Double
Dim nominalSize  As Double
Dim spacing As Double

Private Sub Form2_Load(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
Button2.Visible = false
End Sub
Private Sub Label11_Click(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles  Label1.Click
End Sub
Private Sub Button1_Click(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles  Button1.Click
        longerSpan = Val(lspan.Text)
        shorterSpan = Val(sspan.Text)
        allowableStress = Val(tastress.Text)
        melasticity = Val(melasticity.Text)
        horizontal Shear = Val(hshear.Text)
        heights  = Val(hgt.Text)
        breadth = Val(bdth.Text)
        Lprovided = Val(lp.Text)

        lumberSection = (breath * norminalSize)
        momentOfInertia = Math.Round (( breadth *
(Math.Pow(heights, 3))) / 12), 2)
        sectionModulus = Math.Round (( breadth *
(Math.Pow(heights, 2))) / 6), 2)
        joistdesignLoad = ((Lprovided / 1000) *
designLoads

        Dim u As Double
        U = (allowable Stress * sectionModulus) /
joistDesignLoad
        U = Math.Sqrt (u)

shear = Math.Round((horizontalShear * breath *
heights) / (0.9 * joistDesignLoad)), 2)
deflection = Math.Round((0.74 *
(Math.Pow((melasticity * momentOfInertia) /
joistDesignLoad). (1 / 3)))), 2)
bending = Math.Round ((3.16 *
(Math.Sqrt((allowableStress * sectionModulus) /
(designload)))), 2)

        Dim a As String
        Dim s As Double

        If bending < shear And bending < deflection
Then
            a = "Bending"
            s = bending

        ElseIf shear < bending And shear <
deflection Then
            a = "Shear"
            s = shear
        ElseIf deflection < shear And deflection <
bending Then
            a = "Deflection"
            s = deflection
        End If
        Dim t As Double
         t = Math.Round (s / 10) * 10
        t  = (t -10)
        'Form3.Lprovided = t
        spacing = t
        Spaces = longerSpan / t
        Spaces = math.Ceiling (spaces)
        noOfStringers = spaces + 1

        populateResult ()

        End Sub
        Public Sub populateResult()
        Button2.Visible = True
        results.Visible = True

results.Text =  "SECTION MODULUDS=  " &
sectionModulus  & "mm3   " & vbCrLf & vbCrLf
result.Text = result.Text  + "MOMENT OF
INERTIA =  " & momentOfInertia &  "   mm4 " &
vbCrLf & vbCrLf
result.Text = result.Text  + "DESIGN LOAD =  " &
joistDesignLoad &  "   KN/mm2 " & vbCrLf &
vbCrLf
result.Text = result.Text  + "BENDING =  " &
bending &  "   mm " & vbCrLf & vbCrLf
result.Text = result.Text  + "SHEAR =  " & shear &
"   mm " & vbCrLf & vbCrLf
result.Text = result.Text  + "DEFLECTION =  " &
deflection &  "   mm " & vbCrLf & vbCrLf
result.Text = result.Text  + "PROVIDE  " &
noOfStringers &  "   STRINGERS AT " & spacing "
mm SPACING"

        Form5.noOfStringers = noOfStringers
        Form5.stringerpacing  = stringerspacing
        End Sub
        Private Sub Button2Click(ByVal sender As
System.Object, ByVal e As
        System.EventArgs) Handles Button2.Click
```

```
 Form3.Show()
End Sub
End Class

Public Class Form3

Public lprovided As Double
Dim spaces As Double
Dim bayLength As Double
Dim bayWidth As Double
Public noOfShores As Double
 Dim breadth As Double
Dim heigths As Double
Dim lumberSection As Double
Dim momentOfInertiaAs Double
Dim sectionModulus As Double
Dim sDesignLoad As Double
Public DesignLoads As Double
Dim bending As Double
Dim shear As Double
Dim deflection As Double
Dim norminalSize As Double
Dim allowableStress As Double
Dim horizontalShear As Double
Dim modulusofElasticity As Double
Dim stringerThickness As Double
Dim spacingAs Double
Dim spacing As Double

Private Sub Form3_Load(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
End Sub

Private Sub Button1_Click(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles  Button1.Click

 .........GET THE VALUES FROM THE
TEXTBOXES………………………

 breadth = Val(bdth.Text)
heights = Val (hgt.Text)
bayLength = Val(lspan.Text)
horizontal Shear = Val(hshear.Text)
modulusOfElasticity = Val(melasticity.Text)
allowableStress = Val(tastress.Text)
'stringerThickness = Val(stck.Text)
Lprovided = Val (lp.Text)

…………CALCULATIONS……………………

norminalSize = (0.75 * sheatingThickness)
lumberSection = (breath * norminalSize)
momentOfInertia = Math.Round (( breadth *
(Math.Pow(heights 3))) / 12), 2)
sectionModulus = Math.Round ((( breadth *
(Math.Pow(heights, 2))) / 6, 2)
sDesignLoad = Math.Round ((lprovided /
1000  + designLoads) , 2)
bending = Math.Round ((3.16 *
(Math.Sqrt((allowableStress *
sectionModulus) / (sdesignload)))), 2)
shear = Math.Round((horizontalShear *
breath * heights) /
(0.9 *sDesignLoad), 2)
deflection = Math.Round((0.74 *
(Math.Pow((modulusOfElasticity *
momentOfInertia)
/ (sDesignLoad). (1 / 3)))), 2)

If bending < shear And bending < deflection
Then
 a = “Bending”
b = Bending
ElseIf shear < bending And shear <
deflection Then
a = “Shear”
b = Shear
ElseIf deflection < shear And deflection <
bending Then
a= “Deflection”
a=  Deflection
End If
b = Math.Round (b/10) * 10
b = b - 10
Spacing = b
Spaces = bayLength / b
Spaces = math.Ceiling (spaces)
noOfShores = spaces + 1
populateResult ()
End Sub

Public Sub populateResult()
Result.Visible = True
result.Text =  “SECTION MODULUS =  “ &
sectionModulus  & “mm3   “ & vbCrLf & vbCrLf
result.Text = result.Text  + “MOMENT OF
INERTIA =  “ & momentOfInertia &  “   mm4“ &
vbCrLf & vbCrLf
result.Text = result.Text  + “DESIGN LOAD =  “ &
sDesignLoad &  “   KN/mm2“ & vbCrLf & vbCrLf
result.Text = result.Text  + “BENDING =  “ &
bending &  “   mm “ & vbCrLf & vbCrLf
result.Text = result.Text  + “SHEAR =  “ & shear &
“   mm “ & vbCrLf & vbCrLf
result.Text = result.Text  + “DEFLECTION =  “ &
deflection &  “   mm “ & vbCrLf & vbCrLf
```

```
result.Text = result.Text  + “PROVIDE=  “ &
noOfShores &  “   SHORES AT “ & spacing “  mm
SPACING”
        Form5.noOfShores  = noOfShores
        Form5.shoresSacing  = spacing
        showResult ()
        End Sub
        Private Sub lspan_TextChanged(ByVal
sender As System.Object, ByVal e As
        System.EventArgs) Handles
lspan.TextChanged
        End Sub
        Private Sub showresult ()
        Form5.Show ()
        End Sub
        End Class

Public Class Form5

Public noOfJoists As Double
Public JoistSpacing As Double
Public   noOfStringers As Double
Public   stringerSpacing As Double
Public noOfShores As Double
Public shoreSpacing As Double
Public  designLoad As Double

Private Sub TextBox 3_TextChanged(ByVal sender
As System.Object, ByVal e As
System.EventArgs) Handles  TextBox
3.TextChanged
End Sub
Private Sub TextBox 1_TextChanged(ByVal sender
As System.Object, ByVal e As
System.EventArgs) Handles  TextBox
1.TextChanged
End Sub
Private Sub Form5_Load(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles  MyBase.Load

TextBox 1.Text  =  “DESIGN LOAD =  “ &  vbCrLf
& vbCrLf
TextBox 1.Text  =   TextBox 1.Text + “DESIGN
LOAD =  “ &  designLoad  &     “ Kn/mm2
TextBox 2.Text  =  “STRINGER RESULT”  =  “ &
vbCrLf & vbCrLf
TextBox 2.Text  =   TextBox 2.Text + “PROVIDE =
“ &  noOfStringers  &  STRINGERS  @    “ &
stringerSpacing  &      “ mm  SPACING”
TextBox 4.Text  =  “SHORES  RESULT”  =  “ &
vbCrLf & vbCrLf
TextBox 4.Text  =   TextBox 4.Text + “PROVIDE =
“ &  noOfShores  &  SHORES  @    “ &
shoreSpacing  &      “ mm  SPACING”
TextBox 3.Text  =  “JOIST  RESULT”  =  “ &
vbCrLf & vbCrLf
TextBox 3.Text  =   TextBox 3.Text + “PROVIDE =
“ &  noOfJoists  &  JOISTS  @    “ & joistSpacing  &
“ mm  SPACING”
        End Sub
        Private Sub  TextBox2_TextChanged(ByVal
sender As System.Object, ByVal e As
        System.EventArgs) Handles  TextBox
2.TextChanged
        End Sub
        Private Sub  TextBox4_TextChanged(ByVal
sender As System.Object, ByVal e As
        System.EventArgs) Handles  TextBox
4.TextChanged
        End Sub
        Private Sub Button1Click(ByVal sender As
System.Object, ByVal e As
        System.EventArgs) Handles Button1.Click
          Form1.Close()
         Form2.Close()
        Form3.Close()
        Form1.Show()
        Me.Close()

        End Sub
        End Class
Public Class Form6

Private Sub Form6_Load(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
 Timer1.Enabled = True
Timer1.Interval =  10000
End Sub

Private Sub Timer1_Tick(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles Timer1.Tick
        Form1.Show()
        Me.Close()
End Sub
End Class
```

Acknowledgements

The author acknowledges the following persons for their contributions: Messers Amako, I. C. and Onyeukwu, D.N.

References

[1]. M. K. Hurd, Formwork for concrete, special publication, No 4, Detroit Michigan: American Concrete Institute, ,1989

[2]. R.C. Ringwald, "Formwork Design." Journal of construction Engineering and Management. ASCE, Vol. 113, No. 4, 1985, pp 391-403

[3]. C.E. Moore, Concrete Form Construction. New York: Van Nostrand Reinhold, 1977.

[4]. J. Christain, S.U. Mir, "Use of integrated micro computer package for formwork design" .Journal of construction Engineering and Management. ASCE, Vol. 113, No 4, 1987, pp 608-610.

[5]. D.M.Wrox, Beginning with Visual Basic, New York: Plenum Publishing Corporation, 1990.

Author Biography

Okere C.E. holds a Bachelor of Engineering (B.Eng) in Civil Engineering (1998), the Master of Engineering (M.Eng) in Structural Engineering (2006) and the Doctor of Philosophy (Ph.D) in Sttructural Engineering (2012), all from the Federal University of Technology Owerri Nigeria. She is a Senior Lecturer in Civil Engineering department, Federal University of Technology, Owerri. She was the best graduating student of Civil Engineering, in 1998. She is a co-author of the paper titled "A model for optimization of compressive strength of sand-laterite blocks using Osadebe's regression theory" which won the best paper award in International Journal of Engineering and Technical Research Vol. 2, Issue 1, 2014. She has over 20 journal papers in both local and international journals. She is a member of Nigeria Society of Engineers (NSE), Association of Professional Women Engineers of Nigeria (APWEN) and Society of Women Engineers (SWE). She is also registered by the Council for the regulation of Engineering in Nigeria (COREN).