

## **Automated street lighting control**

**S. O. Igbinovia<sup>1</sup>, R. O. Osayande<sup>2</sup>, and S. Omodamwen<sup>3</sup>**

<sup>1, 2, 3</sup> Department of Electrical and Electronic Engineering,  
Faculty of Engineering, University of Benin, Benin City, Nigeria.

---

### *Abstract*

---

*The control of street lights in some of the streets outside the campus and in university of Benin, Ugbowo Campus, Benin City, using Switching control circuit interfaced with a computer to enable effective control of the street light was investigated in this work.*

*The methodology adopted in this work includes:*

- *Designing and constructing a dc power supply unit which provides +12Volts and +5Volts to power the transistorized switching logic circuits and the Multiplexers respectively,*
  - *Designing a suitable interface,*
  - *Developing a control program written in visual Basic which is linked to the hardware with the help of the dynamic link library [DLL] program,*
  - *Demonstrating the model work which had twelve 40 Watts connected in three circuits depicting the adjoining streets.*
- The sequence of address inputs and outputs functions are indicated and generated by the computer program. The switching time and delay time of ON/OFF of the circuits are dependent on weather and not time of the day. To have effective response signal of the weather we used photocells which drive the transistors into saturation and cut – off modes. The automated street lights control capable of handling up 167Amp. Current at 230Volts, and intended to eliminate human inadequacies, worked satisfactorily. The relevance of this scheme is its capability of saving electric energy consumption and reduction in man hour which translates to revenue saving. All these could results in new concept and development in our environment.*
- 

### **Nomenclature**

DLL: Dynamic link library, IC: Integrated circuit, KBPs: Kilobytes per seconds  
LPT: Link port terminal, PID: Proportional integral differential, TDM: Time division  
Multiplexing, TTL: Transistor-transistor logic, VB: Visual basic

**1.0 Introduction** Man over the decades has transformed his primitive nature to highly developed and proven technology. To eliminate the problems associated with inadequacies in man's physical senses and reliability, automated circuits are been introduced into processes to enhance systems overall efficiency. These desired automated circuits reduce stress and energy as it controls itself and are seldom dependent on human intervention in case the need arises. This project was necessitated because from our observation from the offices, most times the switching time goes far into the evenings before the operators' switches on the adjoining street lights behind electrical/electronic engineering department.

---

<sup>1</sup>e-mail: [samigbinovia2006@yahoo.com](mailto:samigbinovia2006@yahoo.com)

Since our interest in this work is automating street lighting in order to switch ON the circuits at appropriate time of the day at dawn i.e once the natural sunlight goes down after sunset, meaning ambient illuminance goes below 60 Lux, and OFF at dawn on sensing the illuminance level to be above 10Lux [1] to save electric energy consumption, which translates into operating cost savings, the adequately designed automated street lighting control facilitates the following:

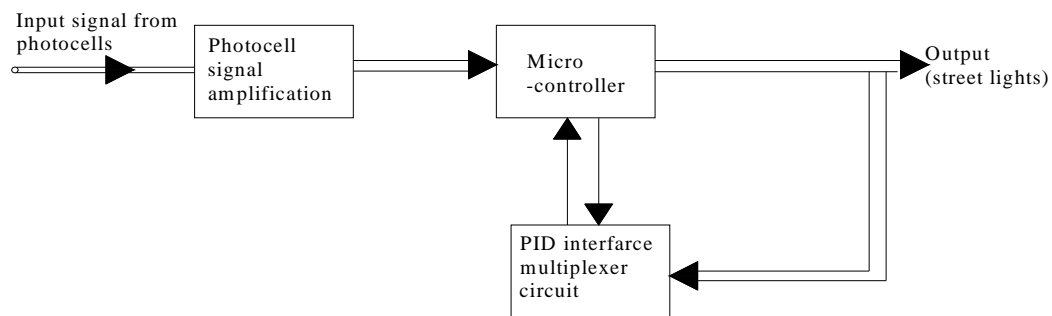
- Enhancement of vision during the night,
- Reduces mishap/ accidents, resulting in most cases amputation and lost of life,
- Security consciousness, thus encouraging free movements,
- Encourages commercial activities,
- Wade-off night marauders, burglary and robbery activities,
- Photocell sensor with a defined field of view,
- Facility for selecting time selection and good resolution,
- Saves electric energy consumption and
- Precise timeliness

The non-provision of street lights virtually along our various categories of roads in Nigeria, lack of maintenance and the manual/mode of operation of the existing ones and coupled with the insistence vandalization of the circuit, it becomes very necessary to incorporate automatic control modes in street light circuits and investigate such systems using locally developed codes in visual basic.

In section 2 of this work the operating principles of the work is presented, Section 3 presents the system design, while section 4 illustrates how the streets were modeled to demonstrate the automation. The controlled circuit results are presented in section 5, while section 6 concludes the work.

## 2.0 Operating principles

The operation of the device is based on direct sequence spread spectrum technology [2]. The designed and constructed automated street light controller is interfaced with a computer with controlled sequence of instructions which reduced the stress of moving from one location to another location of switching to turn ON/OFF the street lights. The control circuit is made up of three stages as shown in Figure 2.1.



**Figure 2.1:** Multivariable control system for the street light

The street light circuits are burden in sections, the photocells with the help of the computer sends control signals to the circuits depending on the address in binary codes that has been sent by the computer to the control section via the parallel port. The operation is controlled by sequence of instructions written in visual basic. The signals from the photocells are amplified, converted and fed into the controller. The interface is made possible via the use of DB25 parallel port. The 9 pins used in the parallel port were grouped into input and output pins in order to eliminate off-set voltages between computer parallel port and designed circuits. The parallel port which transmits 8 bits from the A/DC simultaneously has a maximum data rate of 6KBPS, which exceeds our requirement of 1KBPS [3, 4].

The output of the parallel port is TTL compatible, thus no special interfacing. The cost of operating parallel port arrangement is relatively cheap and this out-weights the disadvantage of average cost of using computer.

Operating the transistors in saturation and cut-off modes with minimum power dissipation enables the transistorized relays to make and break the particular street lights circuit. Advancement in electronic elements technology has in a long way improved the resolution, stability and sensitivity of control and computer systems.

The time required for the transistor to switch from the “OFF to ON” and “ON to OFF” stages in milliseconds respectively are [4, 5];

$$t_{on} = t_r + t_d \quad (2.1)$$

$$t_{off} = t_s + t_f \quad (2.2)$$

where

$t_r$  = rise time from 10% to 90% of final value

$t_d$  = delay time between changing state of the input

$t_s$  = storage time

$t_f$  = fall time from 90% to 10% of initial value

These values are dependent on the speed of response and the collector current response. To actualize the functionality of this circuit, CD 4051 8-channels multiplexer with only 2 address lines to reduce the state of the multiplexer to 4 while others grounded were utilize. Table 1.0 shows the base address for LPT1 and Typical computer used in this work. The table enables pins combination, the capacity and type of information's in the bus, the expected I/O data and the variable addresses.

**Table 2.1:** Base address for LPT1/Typical computer

Register	Pins	LPT1	Function
Data register	[2-9]	378[hex]	Input-output
Status	[10-13, 15]	379 [hex]	Input
Control	[1,14,16,17]	37A [hex]	Output
Ground	[18-25]		Ground

### 3.0 System design

The design of the street light control system involves the use of computer to analyse the feedback signals from the electronic circuit [hardware] and software (drives, and the main program).

#### 3.1 Design calculations

##### 3.1.1 DC power supply

The essential parameters for the power supply unit and sensory circuit design for the interface are as follow [3, 4, 5, 6].

The step-down transformer transforms the 220 main supply voltages to no-load 15V AC which is rectified to get +12V DC and + 5VDC required in the sub- circuits. The transformation ratio is given by the relationship;

$$\frac{V_p}{V_s} = \frac{220}{15} = 14.66 : 1 \quad (3.1)$$

The power transfer efficiency of the transformer is given by the relation;

$$V_p \times I_p = V_s \times I_s \times \eta \quad (3.2)$$

where,  $\eta$  = percentage efficiency of the transformer

$V_p$  = primary voltage

$V_s$  = secondary voltage

$I_p$  = primary current

$I_s$  = secondary current

The rectified pulsating peak voltage is given by the relation;

$$V_{PDV} = 15 - 2V_D = 15 - (2 \times 0.6) = 13.80V \quad (3.3)$$

where,  $V_D$  is the voltage drop across the diode.

The average output voltage from the rectifier circuit is;

$$\begin{aligned} V_{OAV} &= V_{PDV} \times 0.638 \\ &= 13.8 \times 0.638 = 8.80 \text{ Volts} \end{aligned} \quad (3.4)$$

In this wise IC 7812 and 7805 were used to achieved DC output full-load voltages of +12V and +5V in the circuitry. To calculate and select the capacitor required in filtering the ripples, equation 7 was used.

$$\begin{aligned} \text{Ripple of the output voltage} &= \frac{V_{RMS}}{V_{DC}} \times 100\% \\ &= \frac{15}{12} \times 100 = 125\% \end{aligned} \quad (3.5)$$

This implies that 25% regulation was done in filtering the pulsating DC voltage waveform to achieve the pure DC voltage.

### 3.1.2 Lamp parameters calculation

The current (I) flow across each of the 40Watts bulb is given by;

$$I = \frac{P}{V} = \frac{40}{220} = 0.182 \text{ Amps.} \quad (3.6)$$

The allowable voltage drop across the lamp resistor,  $R_L$  is;

$$V_{AVD} = IR_L = 0.182 \times 40 \Omega = 7.28V \quad (3.7)$$

The maximum power loss ( $P_L$ ) in the lamps is;

$$P_L = I^2 R_L = (0.182)^2 \times 40 = 1.33 \text{ watts} \quad (3.8)$$

Twelve (12) Lamps were used to represent the three adjoining streets in the model. Thus the full load current allowed in the circuit is 2.18Amps.

### 3.1.3 Calculation for the transistorised relay switching parameters

The base current for transistor Q24, is obtained from;

$$I_B = \frac{I_C}{\beta} = \frac{1 \times 10^{-3}}{60} = 16.67 \mu A \quad (3.9)$$

where,  $I_C$  = collector current, Amps

$\beta$  = forward gain of the device, for this case (= 60)

Applying kirchoff's current law, the collector limiting resistance is given by;

$$\begin{aligned} R_{BB} &= R_{29} + R_{36} \\ &= \frac{V_{CC} - 0.7}{I_B} = \frac{5 - 0.7}{16.67 \times 10^{-6}} = 258.00 K\Omega \end{aligned} \quad (3.10)$$

The base Resistor,  $R_B$  of transistor Q12 is  $R_{36} = R_X = 43K \Omega$ , which is chosen as  $R_C$ . Therefore the collector response current for transistor Q25 is;

$$I_C = \frac{V_{CC}}{R_C} = \frac{5}{43} = 116 \mu A \quad (3.11)$$

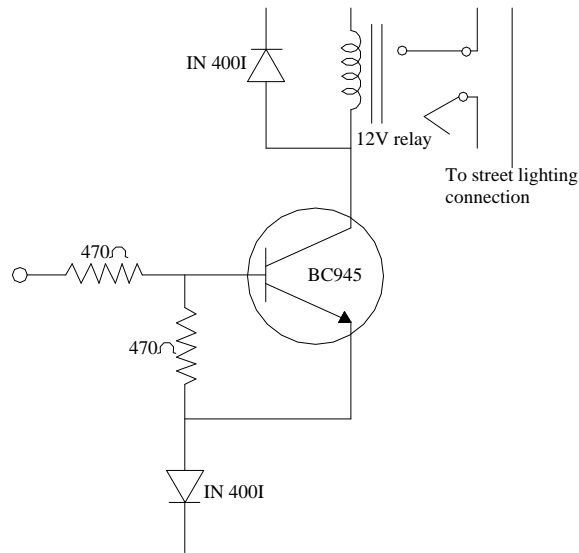
The base limiting resistor for transistor Q25 is;

$$= \frac{V_B - 0.7}{I_B} \quad (3.12)$$

$$= \frac{V_B - 0.7}{I_{C_2} / \beta} = \frac{7.23 - 0.7}{\frac{116 \times 10^{-6}}{60}} = \frac{7.16}{1.94 \times 10^{-6}} = 3.36 M\Omega$$

### 3.2 The controller circuits

This entails the design of the controller and the multiplexer circuits components shown in Figures 3.1 and 3.2 for effective switching of the circuits [3, 4, 5] respectively.



**Figure 3.1:** Control circuit

The multiplexer is used to select which of the input signals appears at the output [3]. The outputs of the multiplexers are the status signal of the bulbs at different sections of the street to the computer as depicted in Figure 3.2. The technique adopted in this work is called time division multiplexing (TDM) [3].

The multiplex operation is based on the formula;

$$2^n = N \quad (3.13)$$

where,  $n$  = number of inputs or addresses, which is two in this case

$N$  = number of outputs or states



time (dynamically). In writing this program that works with all versions of windows operating system, the following sequences were followed:

- (i) Write VC ++ code that uses Read/Write functions
- (ii) Compile it into a DLL (instead of an executable file)
- (iii) Call your functions from VB

### **3.4.2 Visual basic (VB) program development**

The program (VB) considered manual setting of the street lights, setting general ON/OFF times and enabled feed backs. The slide below the timers is used to increase or decrease the delay time in controlling the ON/OFF switching time of the street lights.

Despite its appropriate speed visual basic programming language was used due to the following reasons:

- (i) Not for the ease of learning but support for rapid application development (RAD)
- (ii) It takes no lines of sources codes to create an application frame, but simply drags, design form and manipulates (i.e. easy of upgrading)
- (iii) It is interpreted rather than compiled with VB. Conversion from the intermediary code to machine code is done with what is called visual machine.
- (iv) It is an event driven program
- (iv) It has support for abject oriented programming
- (vi) The codes are in modules and this reduces the changes of bugs.

Though the high level language has the above advantages, the following disadvantages manifest in its running:

- (i) Lack of access port
- (ii) Interpreted, hence slower than C/C++

The lack of access port in this work was overcome using infrastructure which is supported by VB.

## **4.0 Modeling of the street lights**

To demonstrate the functionality of the designed automatic control system, the road linking social sciences from faculty of engineering and some of the adjoining roads were mapped on a board using twelve 40Watts bulbs as a prototype.

## **5.0 Results**

The developed code in this work is as contained in appendix 1. The display default provides an interface for the user to send information to the circuits and also receive feedback signals from the circuits. The developed program ran perfectly well in the auto – stage. When the controlled circuit was disabled for manual operation of the street lights the scheme also worked perfectly.

## **6.0 Conclusion**

The implementation of the conceived scheme: Design and Construction of an automated model for streets lighting control was feasible. Using a computer interface and with the help of a visual basic programming, it was easy to write windows 32-bit applications which communicate with the hardware link DLL through the parallel port of the computer. This has been able to solve the problem arising from man's inefficient attitude in performing their imposed duties of street lights control. The cost of schemes of this nature depends relatively on the circuit complexity.

The automated control system also allows manual operation of the switching arrangement of the street light. Though it is a costlier scheme to operate, its operating efficiency and energy saving cost is worth implementing in controlling street lights. The most facilitating aspect of this device is that it's intelligent enough to sense the Lux level depending on the weather condition irrespective of the mains power and activate the switches.

### **Appendix 1: Visual Basic (VB) Code Listing for the Automated street lighting control**

The code for the automated street lighting control using VB programming language is show in this section. The code was developed by: S. O.Igbinovia, R. O. Osayande, and S.Omodamwen, Department of Electronics and Electctrical Engineering, Faculty of Engineering, University of Benin, Benin city, Nigeria.

### **Option Explicit**

```
'Send a value to a specified port
Private Declare Sub PortOut Lib "IO.DLL" (ByVal Port As Integer, ByVal Data As Byte)
' Returns a value from specific ports.
Private Declare Function PortIn Lib "IO.DLL" (ByVal Port As Integer) As Byte
'Delay for in Miliseconds
Private Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)
Public delayTime As Single
Private Sub cmdExit_Click()
End
End Sub
Private Sub Form_Load()
PortOut 888, 0
End Sub
Private Sub iSliderX1_OnPositionChange()
iSliderX1.ToolTipText = iSliderX1.Position
delayTime = iSliderX1.Position
End Sub
Private Sub iSliderX1_OnPositionChangeFinished()
iSliderX1.ToolTipText = iSliderX1.Position
End Sub
Private Sub iSwitchLeverX1_OnChange()
Dim i As Integer
```

```

If Not iSwitchLeverX1.Active Then
For i = 0 To 3
ledSub1(i).Active = False
Next i
PortOut 888, PortIn(888) And 253
Else
For i = 0 To 3
ledSub1(i).Active = True
Next i
PortOut 888, PortIn(888) Or 2
End If
End Sub
Private Sub iSwitchLeverX2_OnChange()
Dim i As Integer
If Not iSwitchLeverX2.Active Then
For i = 0 To 3
ledSub2(i).Active = False
Next i
PortOut 888, PortIn(888) And 251
Else
For i = 0 To 3
ledSub2(i).Active = True
Next i
PortOut 888, PortIn(888) Or 4
End If
End Sub
Private Sub iSwitchLeverX3_OnChange()
Dim i As Integer
If Not iSwitchLeverX3.Active Then
```



```

    For i = 0 To 3
        ledMain(i).Active = False
    Next i
    PortOut 888, PortIn(888) And 254
Else
    For i = 0 To 3
        ledMain(i).Active = True
    Next i
    PortOut 888, PortIn(888) Or 1
End If
End Sub
Private Sub OffTime_OnDbClick()
    Dim value As Integer
    value = Int(Val(InputBox("Enter the hour part of the Off Time", "ON TIME", "6")))
    OffTime.value = value
    Switch2.value = Int(OffTime.value) * 60 + (OffTime.value - Int(OffTime.value))
End Sub
Private Sub OnTime_OnDbClick()
    Dim value As String
    value = Int(Val(InputBox("Enter the hour part of the On Time", "ON TIME", "6")))
    OnTime.value = value
    Switch1.value = Int(OnTime.value) * 60 + (OnTime.value - Int(OnTime.value))
End Sub
Private Sub optSLTime_Click(Index As Integer)
    If optSLTime(0).value Then
        Frame2.Enabled = True
        Frame3.Enabled = False
        Timer3.Enabled = False
    Else
        Frame2.Enabled = False
        Frame3.Enabled = True
        Timer3.Enabled = True
    End If
End Sub
Private Sub Switch1_OnValueChanged()
    OnTime.value = Int(Switch1.value / 60) + ((Switch1.value Mod 60) / 100)
    If Switch1.value >= 1440 Then
        OnTime.value = 0
        Switch1.value = 0
    End If
End Sub
Private Sub Switch2_OnValueChanged()
    OffTime.value = Int(Switch2.value / 60) + ((Switch2.value Mod 60) / 100)
    If Switch2.value >= 1440 Then
        OffTime.value = 0
        Switch2.value = 0
    End If
End Sub
Public Function Read(bulb As Integer) As Integer
    'Pins 4 and 5 for selecting the Multiplexer
    'Pins 13,12,10 for reading from the Multiplexer
    Dim DataPort As Integer, StatusPort As Integer
    DataPort = Val("&H378")
    StatusPort = Val("&H379")
    Select Case bulb
        'Read from first Multiplexer through pin 13 (for SUB-1)
    Case 1
        PortOut DataPort, PortIn(DataPort) And 231 'send 00 to Mul

```

*Read = PortIn(StatusPort) And 16*

*Case 2*

*PortOut DataPort, PortIn(DataPort) Or 8 'send 01 to Mul*  
*Read = PortIn(StatusPort) And 16*

*Case 3*

*PortOut DataPort, PortIn(DataPort) And 231 'Send 00 to Mul*  
*PortOut DataPort, PortIn(DataPort) Or 16 'send 10 to Mul*  
*Read = PortIn(StatusPort) And 16*

*Case 4*

*PortOut DataPort, PortIn(DataPort) Or 24 'send 11 to Mul*  
*Read = PortIn(StatusPort) And 16*  
*'Read from second Multiplexer through pin 12 (for SUB-2)*

*Case 5*

*PortOut DataPort, PortIn(DataPort) And 231 'send 00 to Mul*  
*Read = PortIn(StatusPort) And 32*

*Case 6*

*PortOut DataPort, PortIn(DataPort) Or 8 'send 01 to Mul*  
*Read = PortIn(StatusPort) And 32*

*Case 7*

*PortOut DataPort, PortIn(DataPort) And 231 'send 00 to Mul*  
*PortOut DataPort, PortIn(DataPort) Or 16 'send 10 to Mul*  
*Read = PortIn(StatusPort) And 32*

*Case 8*

*PortOut DataPort, PortIn(DataPort) Or 24 'send 11 to Mul*  
*Read = PortIn(StatusPort) And 32*  
*'Read from third Multiplexer through pin 10 (for MAIN)*

*Case 9*

*PortOut DataPort, PortIn(DataPort) And 231 'send 00 to Mul*

*Case 10*

*PortOut DataPort, PortIn(DataPort) Or 8 'send 01 to Mul*  
*Read = PortIn(StatusPort) And 64*

*Case 11*

*PortOut DataPort, PortIn(DataPort) And 231 'send 00 to mul*  
*PortOut DataPort, PortIn(DataPort) Or 16 'send 10 to Mul*

*Read = PortIn(StatusPort) And 64*

*Case 12*

*PortOut DataPort, PortIn(DataPort) Or 24 'send 11 to Mul*  
*Read = PortIn(StatusPort) And 64*

*End Select*

*End Function*

*Private Sub Timer1\_Timer()*

*If chkFeedback.value Then*

*If iSwitchLeverX1.Active Then*

*If Read(1) = 0 Then*

*iLCDMatrixX1.PlaceText "Sub1: B1 is DEAD", 1, 0, vbRed*  
*ledSub1(0).Active = False*

*Else*

*iLCDMatrixX1.PlaceText "Sub1: B1 is GOOD", 1, 0, vbGreen*  
*ledSub1(0).Active = True*

*End If*

*If Read(2) = 0 Then*

*iLCDMatrixX1.PlaceText "Sub1: B2 is DEAD", 1, 1, vbRed*  
*ledSub1(1).Active = False*

*Else*

*iLCDMatrixX1.PlaceText "Sub1: B2 is GOOD", 1, 1, vbGreen*  
*ledSub1(1).Active = True*

*If Read(3) = 0 Then*

*iLCDMatrixX1.PlaceText "Sub1: B3 is DEAD", 1, 2, vbRed*

```

        ledSub1(2).Active = False
    Else
        iLCDMatrixX1.PlaceText "Sub1: B3 is GOOD", 1, 2, vbGreen
        ledSub1(2).Active = True
    End If
    If Read(4) = 0 Then
        iLCDMatrixX1.PlaceText "Sub1: B4 is DEAD", 1, 3, vbRed
        ledSub1(3).Active = False
    Else
        iLCDMatrixX1.PlaceText "Sub1: B4 is GOOD", 1, 3, vbGreen
        ledSub1(3).Active = True
    End If
    Else
        iLCDMatrixX1.PlaceText " ", 1, 0, vbYellow
        iLCDMatrixX1.PlaceText " ", 1, 1, vbYellow
        iLCDMatrixX1.PlaceText " ", 1, 2, vbYellow
        iLCDMatrixX1.PlaceText " ", 1, 3, vbYellow
    End If
    If iSwitchLeverX2.Active Then
        If Read(5) = 0 Then
            iLCDMatrixX1.PlaceText "Sub2: B1 is DEAD", 1, 4, vbRed
            ledSub2(0).Active = False
        Else
            iLCDMatrixX1.PlaceText "Sub2: B1 is GOOD", 1, 4, vbGreen
            ledSub2(0).Active = True
        End If
    If Read(6) = 0 Then
        iLCDMatrixX1.PlaceText "Sub2: B2 is DEAD", 1, 5, vbRed
        ledSub2(1).Active = False
        iLCDMatrixX1.PlaceText "Sub2: B2 is GOOD", 1, 5, vbGreen
        ledSub2(1).Active = True
    Else
        End If
    If Read(7) = 0 Then
        iLCDMatrixX1.PlaceText "Sub2: B3 is DEAD", 1, 6, vbRed
        ledSub2(2).Active = False
    Else
        iLCDMatrixX1.PlaceText "Sub2: B3 is GOOD", 1, 6, vbGreen
        ledSub2(2).Active = True
    End If
    If Read(8) = 0 Then
        iLCDMatrixX1.PlaceText "Sub2: B4 is DEAD", 1, 7, vbRed
        ledSub2(3).Active = False
    Else
        iLCDMatrixX1.PlaceText "Sub2: B4 is GOOD", 1, 7, vbGreen
        ledSub2(3).Active = True
    End If
    Else
        iLCDMatrixX1.PlaceText " ", 1, 4, vbYellow
        iLCDMatrixX1.PlaceText " ", 1, 5, vbYellow
        iLCDMatrixX1.PlaceText " ", 1, 6, vbYellow
        iLCDMatrixX1.PlaceText " ", 1, 7, vbYellow
    End If
    If iSwitchLeverX3.Active Then
        If Read(9) = 0 Then
            iLCDMatrixX1.PlaceText "Main: B1 is DEAD", 1, 8, vbRed
            ledMain(0).Active = False
        Else
            iLCDMatrixX1.PlaceText "Main: B1 is GOOD", 1, 8, vbGreen

```

```

        ledMain(0).Active = True
    End If
    If Read(10) = 0 Then
        iLCDMatrixX1.PlaceText "Main: B2 is DEAD", 1, 9, vbRed
        ledMain(1).Active = False
    Else
        iLCDMatrixX1.PlaceText "Main: B2 is GOOD", 1, 9, vbGreen
        ledMain(1).Active = True
    End If
    If Read(11) = 0 Then
        iLCDMatrixX1.PlaceText "Main: B3 is DEAD", 1, 10, vbRed
        ledMain(2).Active = False
    Else
        iLCDMatrixX1.PlaceText "Main: B3 is GOOD", 1, 10, vbGreen
        ledMain(2).Active = True
    End If
    If Read(12) = 0 Then
        iLCDMatrixX1.PlaceText "Main: B4 is DEAD", 1, 11, vbRed
        ledMain(3).Active = False
    Else
        iLCDMatrixX1.PlaceText "Main: B4 is GOOD", 1, 11, vbGreen
        ledMain(3).Active = True
    End If
Else
    iLCDMatrixX1.PlaceText " ", 1, 8, vbYellow
    iLCDMatrixX1.PlaceText " ", 1, 9, vbYellow
    iLCDMatrixX1.PlaceText " ", 1, 10, vbYellow
    iLCDMatrixX1.PlaceText " ", 1, 11, vbYellow
End If
Else
    Dim i As Integer
    For i = 0 To 11
        iLCDMatrixX1.PlaceText " ", 1, i, vbYellow
    Next i
End If
End Sub
Private Sub Timer2_Timer()

```

```

    Static c As Integer
        c = c + 1
    If c = 9 Then
        c = 0
        Unload Me
        Me.Show
    End If
End Sub
Private Sub Timer3_Timer()
    Dim i As Long
    Timer1.Enabled = False
    Timer2.Enabled = False
    'Check for the on times and put on the bulbs as needed
    If Int(Switch2.value / 60) = Hour(Time) And Int(Switch2.value Mod 60) = Minute(Time) Then
        Timer3.Enabled = False
        iSwitchLeverX1.Active = False
        For i = 1 To 600
            Sleep (CInt(iSliderX1.Position))
        Next i
    End If
End Sub

```

```

        iSwitchLeverX2.Active = False
        For i = 1 To 600
            Sleep (CInt(iSliderX1.Position))
            DoEvents
        Next i
        iSwitchLeverX3.Active = False
ElseIf Int(Switch1.value / 60) = Hour(Time) And Int(Switch1.value Mod 60) = Minute(Time) Then
    Timer3.Enabled = False
    iSwitchLeverX1.Active = True
    For i = 1 To 600
        Sleep (CInt(iSliderX1.Position))
        DoEvents
    Next i
    iSwitchLeverX2.Active = True
    For i = 1 To 600
        Sleep (CInt(iSliderX1.Position))
        DoEvents
    Next i
    iSwitchLeverX3.Active = True
End If
    Timer1.Enabled = True
    Timer2.Enabled = True
    Timer3.Enabled = True
End Sub

```

### **References**

- [1] <http://www.archnetco.com/english/product.ATL90.htm> "New Embedded PLC modern ATL 90 series
- [2] <http://www.naturesswitch.com/glowsign-lightbox-product.html> "Energy conservation, Energy saving, power saving, street light control, Glow sign contr.
- [3] P. Horowitz and H. Winfield (1995) The art of electronics, Cambridge Low Price Editions, International Sales Department, Cambridge University Press, The Edinburgh Building, Cambridge CB2 2RU UK, pp. 4-9, 377, 490-501.
- [4] L. Neshelsky and R. L. Boyelstad "Electronic Devices and Circuit Theory", Seventh Edition, pp.987 – 1120
- [5] S. O. Onohaebi (1998) Basic electronic, Seventh edition, pp. 7-10
- [6] B. L. Theraja, Et Al (1997) "A Textbook of Electrical Technology, New Delhi, S.Chand and company Ltd., 22<sup>nd</sup> edition.
- [7] <http://www.epanorama.com/>
- [8] <http://www.kpsec.freeuk.com/>