

DLite API Manual

Contents

1-1 REAL-TIME MESSAGING USING DLITE API.....	3
1-2 XDLITE OCX METHOD AND EVENTS	4
PROPERTY:	4
METHOD:	4
EVENT:	5
1-3 LEDSIM OCX METHOD AND EVENTS.....	15
PROPERTY:	15
METHOD:	15
1-4 ACTIVEX COMPONENT DEPLOYMENT.....	17
USING ACTIVEX.OCX:	19
SAMPLES PROGRAMMING PATTERNS:	22
VISUAL BASIC PROGRAMMING:	24
<i>VB Sample #1 Real-time message</i>	<i>24</i>
<i>VB Sample #2 Divide Panel</i>	<i>27</i>
<i>VB Sample #3 Sequence Display</i>	<i>29</i>
<i>VB Sample #4 Panel Color Effect</i>	<i>30</i>
<i>VB Sample #5 Interactive Marquee</i>	<i>31</i>
<i>VB Sample #6 Drawing sample</i>	<i>33</i>
<i>VB Sample #7 Upload sample</i>	<i>35</i>
<i>VB Sample #8 QC sample</i>	<i>36</i>
ABOUT ESOFTSYSTEM CORP.....	38

1-1 Real-time messaging using DLite API

DLite is based on ICPDAS embedded remote controller technology. It provides real-time control ability via Ethernet. DLite provides an OCX control to access real-time function of DLite LED display.

User could use OCX to design more flexible and powerful application with DLite API.

We illustrate one Visual Basic sample here to show you how to use these API functions. The sample was under SDK/Sample folder of DLite source CD-ROM folder.

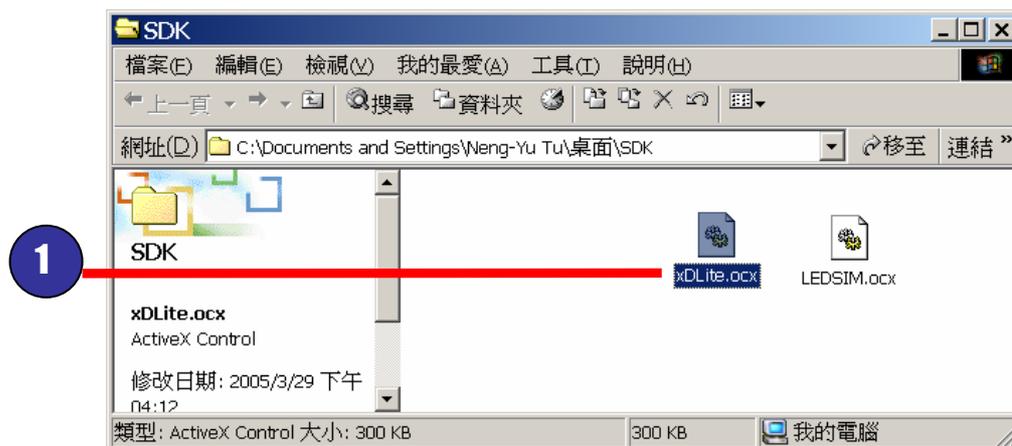


Fig 1-1 The xDLite OCX is under SDK directory

There is another ocx for the LED simulator called **LEDSIM.OCX**, You must register this ocx if you want to preview the script message effects.

1-2 xDLite OCX Method and Events

xDLite OCX provides events callback for developer to control and detect EKAN LED display, these are:

Property:

- **EkanIP:** Connection IP address
- **EkanPort:** Ekan connection port is 7000
- **Password:** The password need to connect to EKAN

Method:

- **Connect:** User provides IP address to connect remote LED display.
- **Disconnect:** for disconnect the connection with LED display.
- **SendMessage:** Used for send text message for real-time display.
- **SendMessageS:** Send small font message
- **SendMessageB5:** Send chinese font message
- **DisplayBMP:** Trigger BMP on LED display
- **DisplatDate:** Display Date on LED display
- **DisplayTime:** Display Time on LED system
- **FillRect:** Fill rectangular area for specific color
- **ClearRect:** Clear rectangular area
- **DrawLine:** Draw line on display
 - **Pixel:** Draw pixel on LED

- **Upload:** Upload bmp or script files to LED.
- **Download:** Download files from LED display.
- **Reset:** Clear all data on LED.
- **Clear:** Clear all text in the display screen.

Event:

- **Disconnected:** triggered after disconnect method.

Following table listed API method arguments:

Property & Method	Argument	Syntax
EkanIP	String like "192.168.0.100"	xDLite1.EkanIP="192.168.0.100"
EkanPort	Number like 7000	xDLite1.Port=7000
EkanPassword	String like "Admin"	xDLite1.Password="admin"
Connect	IP address as string	xDLite1.Connect
Disconnect	None	xDLite1.Disconnect
SendMessage	<p>Need 5 arguments to send the message.</p> <p>PosX and PosY are the coordinate of the Text message, start from LED display upper left corner. Both are Integer type variable.</p> <p>fgClr and bgClr are the foreground color and background color of the displayed text. Both are Integer type variable. Color code index is:</p> <p>0: Black 1: Green 2: Red 3: Orange</p> <p>szMsg is the text string needed to be displayed.</p>	<p>xDLite1.SendMessage 0,0,1,0,"Hello World!" Return > 0 success</p>

<p>SendMessageS</p>	<p>Need 5 arguments to send the small font message. PosX and PosY are the coordinate of the Text message, start from LED display upper left corner. Both are Integer type variable.</p> <p>fgClr and bgClr are the foreground color and background color of the displayed text. Both are Integer type variable. Color code index is:</p> <p>0: Black 1: Green 2: Red 3: Orange</p> <p>szMsg is the text string needed to be displayed.</p>	<p>xDLite1.SendMessageS 0,0,1,0,"Hello World!"</p> <p>Return > 0 success</p>
---------------------	--	---

<p>SendMessageB5</p>	<p>Need 5 arguments to send the small font message. PosX and PosY are the coordinate of the Text message, start from LED display upper left corner. Both are Integer type variable.</p> <p>fgClr and bgClr are the foreground color and background color of the displayed text. Both are Integer type variable. Color code index is:</p> <p>0: Black 1: Green 2: Red 3: Orange</p> <p>szMsg is the text string needed to be displayed.</p>	<p>xDLite1.SendMessageB5 0,0,1,0,"Chinese Char Here"</p> <p>Return > 0 success</p>
----------------------	--	---

<p>DisplayBMP</p>	<p>Need 5 arguments to send the small font message. PosX and PosY are the coordinate of the Text message, start from LED display upper left corner. Both are Integer type variable.</p> <p>fgClr and bgClr are the foreground color and background color of the displayed text. Both are Integer type variable. Color code index is:</p> <p>0: Black 1: Green 2: Red 3: Orange</p> <p>szMsg is the text string needed to be displayed.</p>	<p>xDLite1.DisplayBMP 0,0,1,0,"1.BMP"</p> <p>Return > 0 success</p>
-------------------	--	--

DisplayDate	<p>Need 4 arguments to send the small font message.</p> <p>PosX and PosY are the coordinate of the Text message, start from LED display upper left corner. Both are Integer type variable.</p> <p>fgClr and bgClr are the foreground color and background color of the displayed text. Both are Integer type variable. Color code index is:</p> <p>0: Black 1: Green 2: Red 3: Orange</p>	xDLite1.DisplayDate 0,0,1,0
-------------	---	-----------------------------

<p>DisplayTime</p>	<p>Need 4 arguments to send the small font message. PosX and PosY are the coordinate of the Text message, start from LED display upper left corner. Both are Integer type variable.</p> <p>fgClr and bgClr are the foreground color and background color of the displayed text. Both are Integer type variable. Color code index is:</p> <p>0: Black 1: Green 2: Red 3: Orange</p>	<p>xDLite1.DisplayTime 0,0,1,0</p>
<p>DrawLine</p>	<p>Needed 5 arguments to send the small font message. PosX1 and PosY1 and PosX2 and PosY2, start from LED display upper left corner. Both are Integer type variable. Color</p> <p>0: Black 1: Green 2: Red 3: Orange</p>	<p>xDLite1.Line 0,0,10,10, 3</p>

Pixel	<p>Needed 3 arguments to send the small font message. PosX1 and PosY1 , start from LED display upper left corner. Both are Integer type variable. Color</p> <p>0: Black 1: Green 2: Red 3: Orange</p>	xDLite1.Pixel 0,0, 3
FillRect	<p>Fill rectangular area with specify color</p> <p>Need 5 arguments to send the small font message. PosX1 and PosY1 and PosX2 and PosY2, start from LED display upper left corner. Both are Integer type variable. Color</p> <p>0: Black 1: Green 2: Red 3: Orange</p>	xDLite1.FillRect 0,0,10,10,3

ClearRect	Clear specify rectangular area Need 4 arguments to send the small font message. PosX1 and PosY1 and PosX2 and PosY2, start from LED display upper left corner	xDLite1.ClearRect 0,0,10,10
Upload	Full upload string with file name	xDLite1.uplaod ("c:\ekan.ils")
Download	Full download path for files download. All the file on DLite LED will be downloaded.	xDLite1.download ("c:\test")
Reset	Delete all files on EKAN memory	xDLite1.Reset
Clear	Used for clear the DLite message on LED display.	xDLite1.Clear

Table 1-1 OCX supported method

Following table listed these events that API method might trigger:

Event	When it Happened?	Pass in parameter
Disconnected	After broken the connection with LED device	None

Table 1-2 xDLite.ocx supported events

1-3 LEDSIM OCX Method and Events

LEDSIM OCX provides 2 methods and property for developer to simulate EKAN LED display, these are:

Property:

- **CellWidth:** Display cells' width
- **CellHeight:** Display cells' height
- **CellSpace:** The space between cell and cell
- **LEDModules:** The LED panel numbers, range from 1 to 4
- **CurDir:** The ekan.ils file and BMP file directory

Method:

- **Start:** for start emulator play.
- **Stop:** for stop emulator play.

Here is complete sample for LED emulator; you can find the sample on the SDKSample folder under SDK directory.

Option Explicit

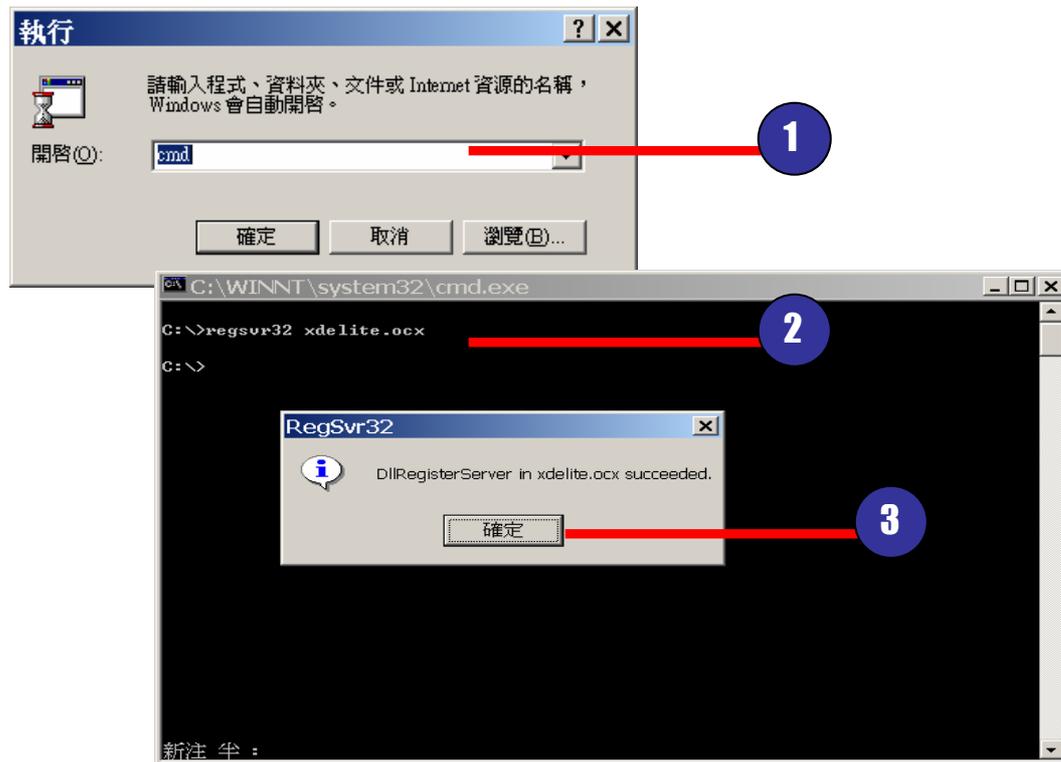
```
Private Sub cmdStart_Click()  
LEDSIM1.CellHeight = Int(txtHeight.Text)  
LEDSIM1.CellWidth = Int(txtWidth.Text)  
LEDSIM1.CellSpace = Int(txtSpace.Text)  
LEDSIM1.CurDir = txtDir.Text  
LEDSIM1.LEDModules = Int(txtModules.Text)  
LEDSIM1.Start  
End Sub
```

```
Private Sub cmdStop_Click()  
LEDSIM1.Stop  
End Sub
```

```
Private Sub Form_Load()  
LEDSIM1.CellHeight = Int(txtHeight.Text)  
LEDSIM1.CellWidth = Int(txtWidth.Text)  
LEDSIM1.CellSpace = Int(txtSpace.Text)  
LEDSIM1.LEDModules = Int(txtModules.Text)  
txtDir.Text = App.Path & "\Demo\QC"  
End Sub
```

1-4 ActiveX component deployment

Before you start using xDLite OCX or LEDSIM.ocx, you must register this OCX to your windows registry first. Please copy the ocx to the program directory, and enter windows command mode, type: regsvr32 xDLite.ocx



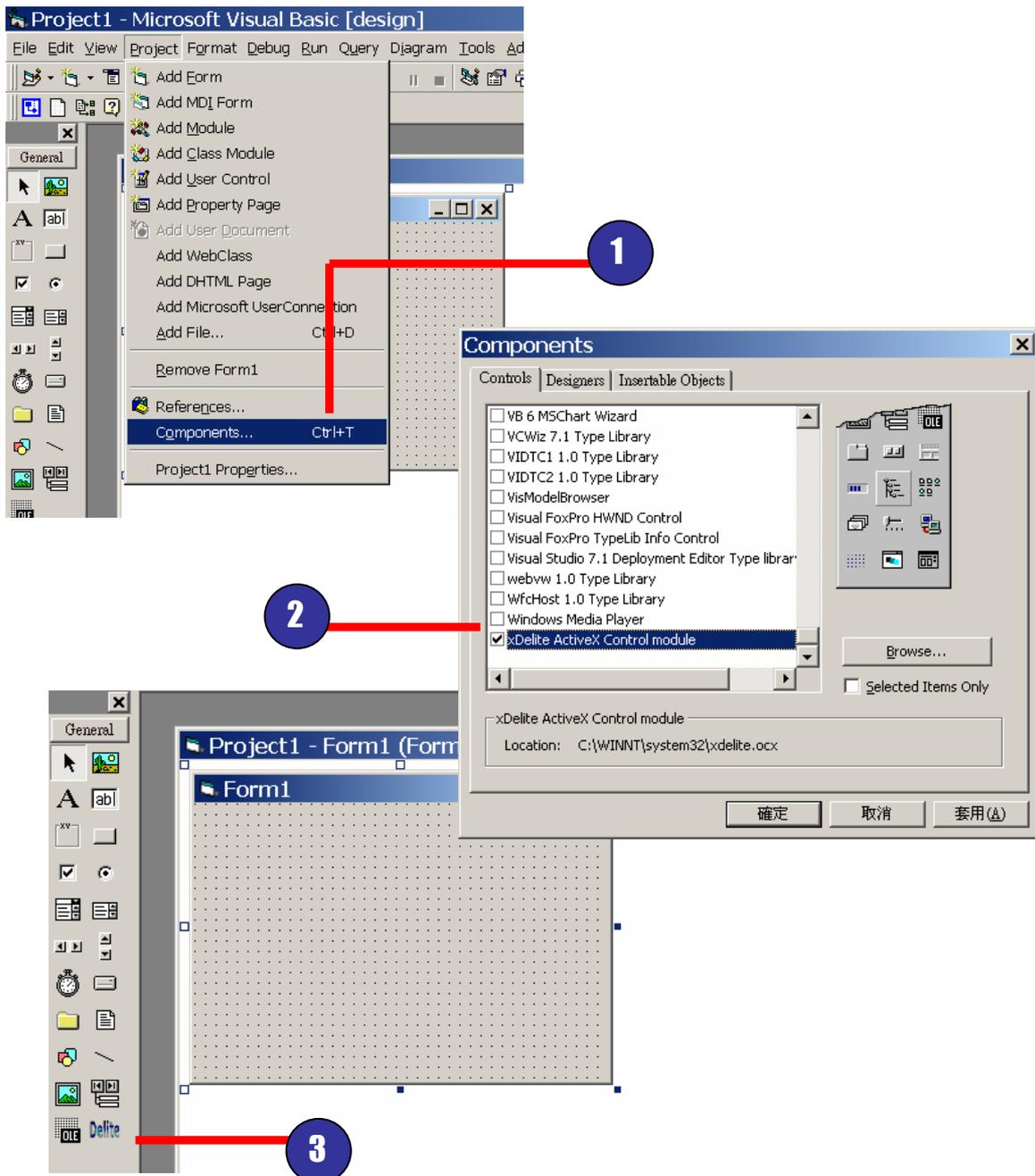
Step 1: Enter the command mode, move to the directory that xDLite.ocx exist and LEDSIM.ocx exists

Step 2: type **regsvr32 xDLite.ocx**

Step 3: type **regsvr32 LEDSIM.ocx**

Step 4: Windows message dialog box shows succeed

After OCX resisted to Windows, you could included it in visual basic develop environment. Please make sure your ocx version is up to date. Please visit eSoftsystem web site at www.esoftsystem.com for further information.



Step 1: Referencing xDLite OCX and LEDSIM.ocx

Step 2: After referencing, you can see it at Visual Basic component tool bar.

Step 3: Using the component into Visual Basic developing environment.

Another quick way to register component, is using the QuickStart sample setup under sample folder, it will automatically add the component into system registry.

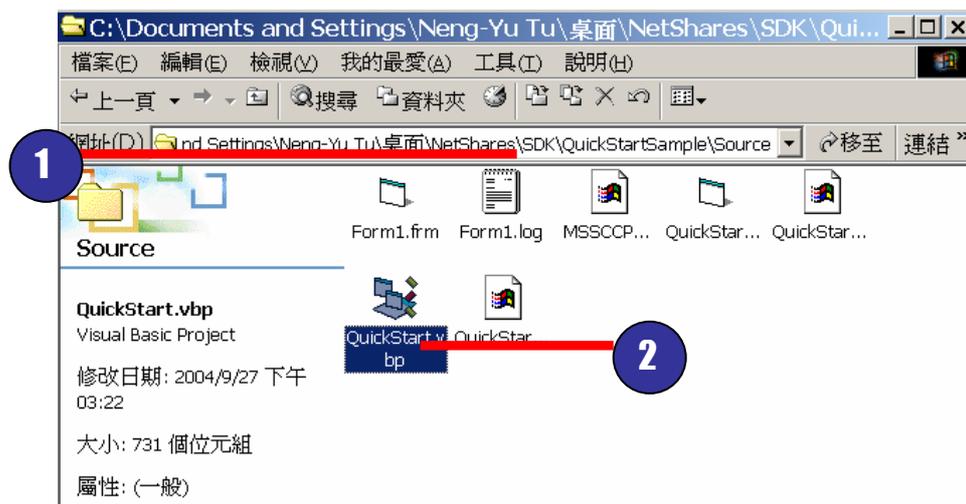
1-5 API Samples

We provide a sample here, which using the Visual Basic development environment.

Using ActiveX.ocx:

In the Visual Basic samples, we will demonstration the basic function of the OCX, and make a simple program to send message to EKAN LED display.

You can direct click on the SETUP.exe under **SDK/QuickStartSample** folder to see the full example.

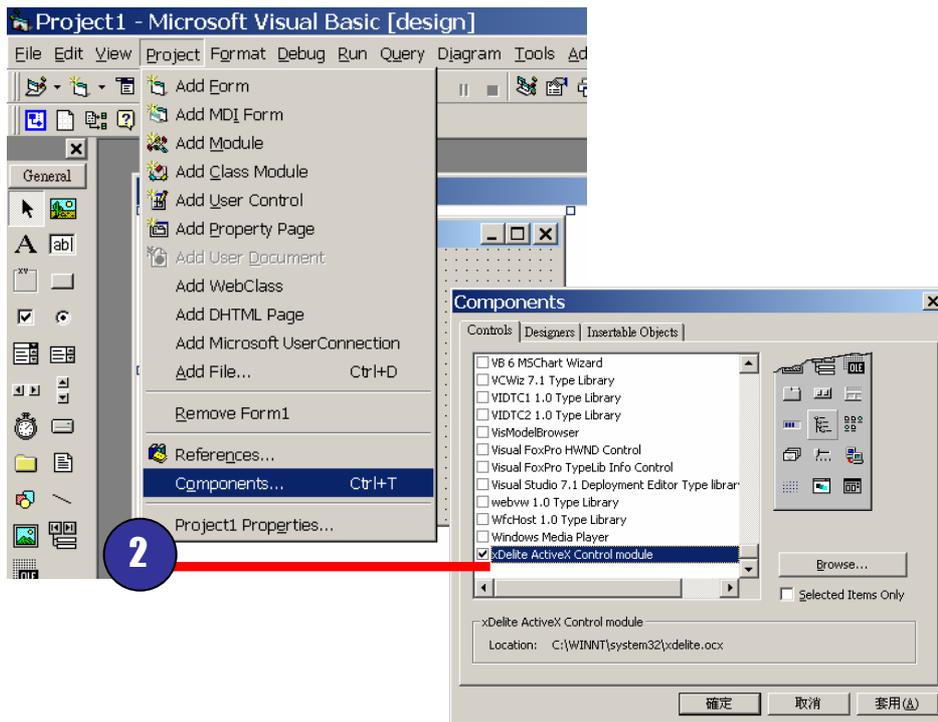


The QuickStart.vbp is the project file

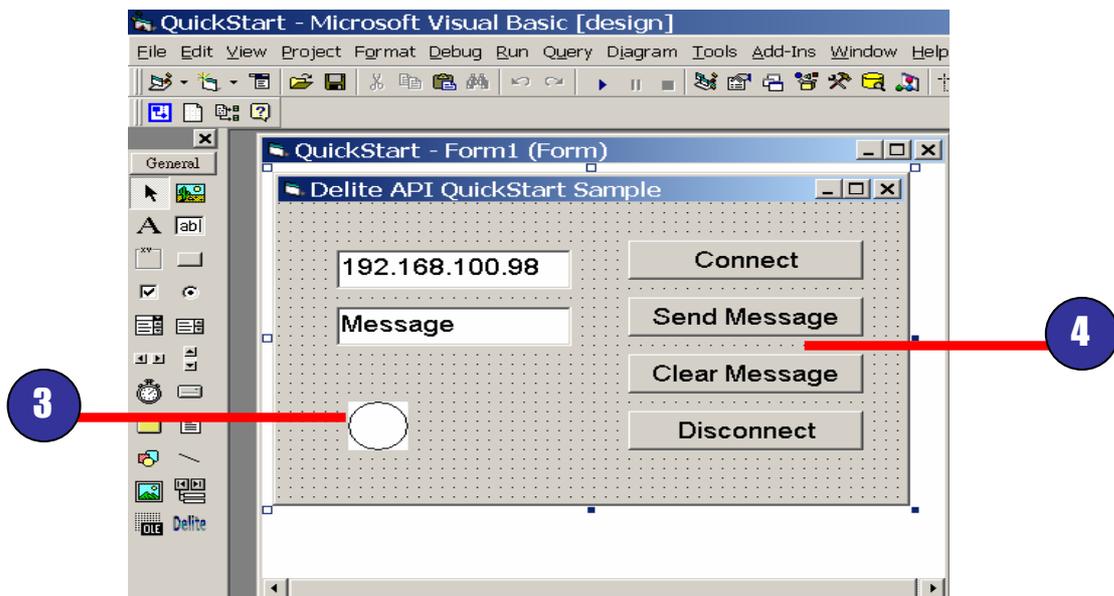
Step 1: Start the Visual Basic program, start a new standard .exe file.

Step 2: Add reference to the xDLite.ocx

Step 3: Add the xDLite.ocx to the form, and make the name of the ocx is xDLite1



Step 4: Add 4 command buttons and 2 textbox into form, please refer to following table:



Type	Name	Caption or Text property
Textbox	txtIP	192.168.0.99
Textbox	txtMessage	Message
Command Button	cmdConnect	Connect
Command Button	cmdDisconnect	Disconnect
Command Button	cmdSendMessage	Send Message
Command Button	cmdClear	Clear Message

Then, Add those following code into command button click event.

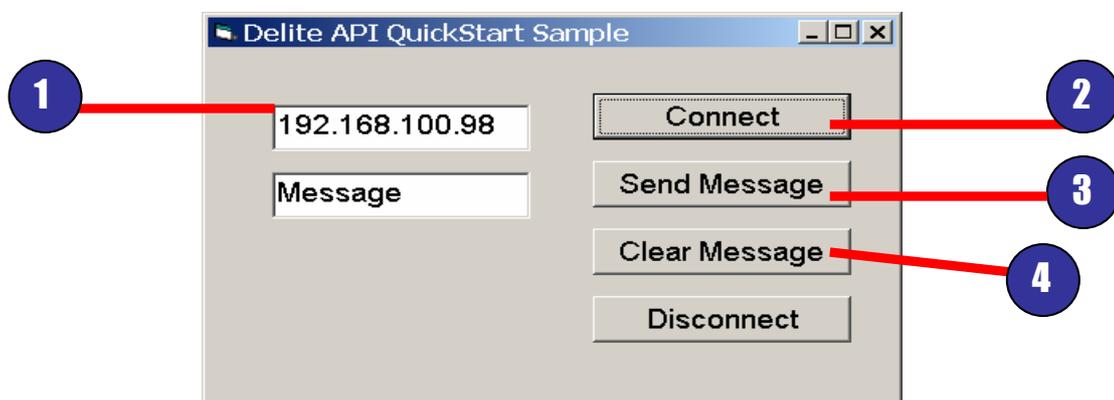
```
Private Sub cmdConnect_Click()
xDLite1.Connect txtIP.Text
End Sub

Private Sub cmdDisconnect_Click()
xDLite1.Disconnect
End Sub

Private Sub cmdSendMessage_Click()
xDLite1.SendMessage 0, 0, 1, 0, txtMessage.Text
End Sub

Private Sub cmdClear_Click()
xDLite1.Clear
End Sub
```

Finally Press **[F5]** to execute the sample:



Step 1: Input IP address of LED Display (Default is 192.168.0.99)

Step 2: Press [**Connect**] button to connect LED Display, and wait for "online" message shows on LED.

Step 3: Press [**Send Message**] to send the Message on text box, you could see "Message" on the LED display by default.

Step 4: Press [**Clear Message**] to clear message on LED display.

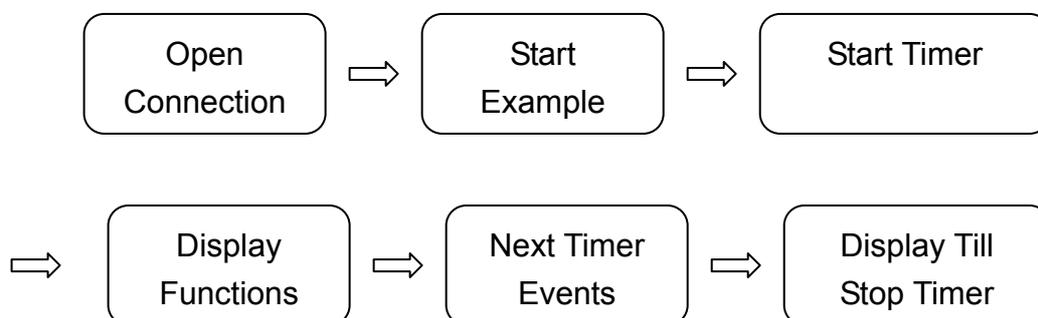
Step 5: Press [**Disconnect**] to disconnect from DLite LED display.

For more samples, please visit the [**SDK/VBSamples**] folder for more information.

Samples Programming Patterns:

There is 1 visual basic sample under VBSample folder, this sample include 5 basic techniques to control real-time message, Which includes the real-time blink, divide panel, sequential display, multiple colors text, and marquee to demonstrate the real-time messaging capability of DLite APIs.

These samples' using pattern as fellows:



5 samples have different function:

#1 is Real-time message sample, demonstrates 4 real-time message methods. Send message, clear message, blink for initial timer, stop to stop timer.

#2 Second sample demonstrated divide panel technique by using location argument of send message method.

#3 Third sample combine with select case and timer, to control the message display sequence, and using the panel location technique to display different type of message in one sample.

#4 Fourth sample using timer and panel location technique to emulate animation. And you can see how to crate animation by send real-time locating message.

#5 Fifth sample are more complex sample, the real-time marquee using location technique, event handling, and conditional select case with timer. You can create more complex effects by mixing more timers and location technique.

#6 Sixth sample is for remote drawing, user could use the remote drawing function to draw graphic on LED display.

#7 This sample is demonstrate how to upload multiple files into LED display.

#8 The QC sample uses the Bitmap file in example 7, shows how to make a small quality check bulletin by single DLite display unit.

In these examples, we used a lot of overlapping and superimposing technique to create complex screen effect.

For example, if we using **xDLite1.SendMessage 0,0,1,0,"AAA"** , we will see green **"AAA"** on LED display. If we don't clear the message on LED display then send second message **xDLite1.SendMessage 8,0,1,0,"B"** we will see **"ABA"** on the screen, and the "B" replaced "A". If we use **xDLite1.SendMessage 8,0,2,0,"B"** , we will get **"ABA"** on the display. With precise design, you might able to design complex graphic effects with simple SendMessage function.

Visual Basic Programming:

VB Sample #1 Real-time message

We will explain what 1st sample does in this section, all samples are located under **SDK\VBSamples** Folder:

Following is connection handling section:

```
Rem #Region xDLite Connection Process Template Start=====

Rem===This segment make the connection
Private Sub cmdConnect_Click()
If bIConnected = False Then
xDLite1.Connect txtIP.Text
cmdConnect.Enabled = False
Else
MsgBox ("Already Connected !")
End If
End Sub

Rem===This section handle disconnect request
```

```
Private Sub cmdDisconnect_Click()
Call SwitchOffAllTimer(True)
xDLite1.Disconnect
cmdConnect.Enabled = True
End Sub

Rem===This send the clear message command to the LED display

Private Sub cmdClear_Click()
xDLite1.Clear
End Sub

Rem===Below this remark is Callback event handle by API

Private Sub XDLite1_ConnectFail(ByVal nErr As Long)
MsgBox "Connection Error, Number is " & nErr, vbExclamation,
"Connection Fail"
bIConnected = False
bIReadyForMessage = False
cmdConnect.Enabled = True
End Sub

Private Sub XDLite1_ConnectReady()
bIConnected = True
bIReadyForMessage = True
xDLite1.Clear
cmdConnect.Enabled = False
xDLite1.SendMessage 0, 0, 3, 0, " Welcome!"
End Sub

Private Sub XDLite1_Disconnected()
bIConnected = False
bIReadyForMessage = False
cmdConnect.Enabled = True
End Sub

Private Sub XDLite1_ErrorResponse(ByVal nErr As Long)

IblSysDisplay.Caption = "Error Message " & nErr

End Sub

Rem #Region xDLite Connection Process Template End=====
```

These events are handled by DLite OCX, user can use these events to maintain connection status and handling connection error exception.

Following code section is the most important part in blinking sample. Timer will be triggered every 500 ms, same time have a flag bTimerFlag to switch which command will be executed. First time the program will be executed the line with bTimerFlag=True, then call a subroutine PaddingText to Pad the text to the screen width, here we use 12 characters.

And using cboMTFColor.ListIndex as foreground color and background color numbers (Range from 0 to 3, corresponds color is Black, Green, Red, Orange).

By default, you will see Green font color with black background (equal to: xDLite1.SendMessage 0,0,1,0,"Message"). After message sent, set the bTimerFlag to False.

When next 500 ms timer will be trigger with xDLite.Clear, and you will see black screen.

By using circulating the message text and black screen, you will see blink effect on LED.

```
Rem #Region Message Blink Sample Start=====
```

```
Private Sub TimerBlink_Timer()
```

```
If bTimerFlag = True Then
```

```
strMessage1 = PaddingText(txtMessage.Text, 12)
```

```
xDLite1.SendMessage 0, 0, cboMTFColor.ListIndex,  
cboMTBColor.ListIndex, strMessage1
```

```
Rem SendMessage function have 5 argument, 1st and 2nd is X coordinate  
Rem and Y coordinate of LED text, start from upper left corner. 3rd and  
4th
```

*Rem Argument is the color code for display, range from 0-3 and
Rem corresponds color is black, green, red, orange. Last parameter is
Rem the message string itself.*

```
bTimerFlag = False

Else

xDLite1.Clear
bTimerFlag = True

End If

End Sub
Rem #Region Message Blink Sample End=====
```

VB Sample #2 Divide Panel

Second sample demonstrated the segment panel display ability by using DLite API position argument.

The program's concept is using SendMessage function's first 2 argument to position text in different start position. For example, if you using xDLite1.SendMessage 48,0,1,0,"Hello" then you will see a "Hello" displayed right in the middle of the LED display.

In this sample, we divide the display into 2 parts, left part of LED display static text "No:" and right part of LED display green number. By using the timer, you will see the right part of display keep changing and left part of display maintain it display status. It's useful technique when you want to mix display static text and dynamic content in the same panel and same time.

Now we look at the sample:

```
Rem #Region Panel Message Sample Start=====
```

```
Private Sub cmdPanelMsg_Click()

xDLite1.SendMessage 0, 0, 2, 0, txtPanel1.Text
xDLite1.SendMessage 66, 0, cmbFColor.ListIndex, cmbBColor.ListIndex,
txtPanel2.Text

End Sub

Private Sub cmdPanelStart_Click()

TimerPanel.Enabled = True
blTimerPanel = True
TimerPanel.Interval = 1000

End Sub

Private Sub cmdPanelStop_Click()
TimerPanel.Enabled = False
End Sub

Private Sub TimerPanel_Timer()
xDLite1.SendMessage 66, 0, cmbFColor.ListIndex, cmbBColor.ListIndex,
CStr(intPanelCounter)
intPanelCounter = intPanelCounter + 1
txtPanel2.Text = intPanelCounter
End Sub

Rem #Region Panel Message Sample End =====
```

When panel message is clicked, we send 2 messages immediately:

- xDLite1.SendMessage 0, 0, 2, 0, txtPanel1.Text
- xDLite1.SendMessage 66, 0, cmbFColor.ListIndex,
cmbBColor.ListIndex, txtPanel2.Text

First send the txtPanel1.Text start from the upper left corner, second one send the message from the X coordinate equal 66 and Y coordinate is 0. And we use intPanelCounter as counter. Number will keep increasing till it's stopped.

VB Sample #3 Sequence Display

Sequence display is emulating a theater time schedule display. The program's timer only has one function call, and the code was inside a Visual Basic module named "SeqDisplay".

When timer is being triggered designated time interval, it will call SeqDisp(ByRef intSeqCount As Integer) with intSeqCount. intSeqCount control the program message display workflow. Following is part of the code segment:

```
Public Sub SeqDisp(ByRef intSeqCount As Integer)

Rem /*using intSeqCount to decide which textbox data will be displayed*/

Select Case intSeqCount

Case 1

Rem /*Stop timer to ensure process will be executed*/

frmTCPTest.TimerSeqDisplay.Enabled = False

frmTCPTest.xDLite1.Clear
frmTCPTest.xDLite1.SendMessage 0, 0, 3, 0,
frmTCPTest.txtSeqDisp11.Text

Rem /*Increasing count to display next textbox content*/

intSeqCount = 2

Rem /*Resume timer for next select case*/

frmTCPTest.TimerSeqDisplay.Enabled = True
```

You can see the sequence:

- Timer called, it will switch case to intSeqCount
- Stop the timer
- Display the textbox message
- Increasing the Sequence number
- Resume timer to next select case.

Pretty straight forward sample, but you could find we mix the 3 basic elements in this sample, they are:

- Timer
- Panel Segment
- Switch Case flow control

We could see more complex application in next 2 examples.

VB Sample #4 Panel Color Effect

This sample is pretty complex, mixing all the technique from example 1 to 3 and plus 2 timer to get more dynamic control ability. This sample consist 2 timer, first timer is display the color text of eSoftsystem, second timer is the real-time direction arrow. Program's main part is in PanelColorFunction Module.

First timer's function just make sure each character's spacing is 8 dots, and using all possible color in the same panel. The code as fellow:

```
Public Sub PanelColorTest(ByRef intPanelColorCount As Integer)

Select Case intPanelColorCount

Case 1

    frmTCPTest.TimerPanelColor.Enabled = False
    frmTCPTest.xDLite1.SendMessage 0, 0, 1, 0,
    Mid(frmTCPTest.txtPanelColor.Text, 1, 1)
```

```
intPanelColorCount = intPanelColorCount + 1  
frmTCPTest.TimerPanelColor.Enabled = True
```

We use Visual Basic Mid function to select the character from the textbox's text string.

Second part is using intPanelColorCount2 and intPanelColorCount3 to calculating spacing, and using Mid function to select arrow character.

The most technique part is 2nd timer has a "If" statement, check the checkbox is checked, and make sure the display text's consistence.

```
Private Sub TimerPanelColor_Timer()  
Rem call the "eSoftsystem"  
Call PanelColorTest(intPanelColorCount)  
  
End Sub  
  
Private Sub TimerPanelColor2_Timer()  
Rem decide the arrow direction  
  
If (intPanelColorCount2 = 1 And chkPanelColor3.Value = 1) Then  
Call PanelColorTest3(intPanelColorCount3)  
Else  
Call PanelColorTest2(intPanelColorCount2)  
End If  
  
End Sub
```

VB Sample #5 Interactive Marquee

Marquee is a pretty useful sample if you want to create a dynamic marquee effect.

Because we want to make sure if background inconsistency problem won't happen (like "AB  "). We call a small function padding text here to avoid this problem.

In the Timer, we just call corresponds scrolling function as list in the combo box, then add or minus the X, Y coordinate number as request. Same time, we catch the textbox change event, and padding the changed text if need. The program workflow listed as fellow:

- InitTimer
- StartTimer
- Detect what effect is selected right now
- Call corresponding scrolling function
- Pause timer for process
- SendMessage based on calculation
- Resume timer
- Detect next scrolling effects
- Loop till timer is stopped

The DLite LED display width is 16 dots high and 96 dots wide, so the screen actual size is like following illustration:

	X=0, Y=-16	
X=-96, Y=0	LED Display Area	X=96, Y=0
	X=0, Y=16	

You can see the X, Y coordinate limit we used in the example, and we use 16 and 96 to control the text display.

```
Public Function ScrollLeft(ByRef PosX As Integer, ByRef PosY As Integer,
    ByVal PanelHeight As Integer, ByVal PanelWidth As Integer, ByVal
    MessageMQ As String)
```

```
frmTCPTest.TimerMQ.Enabled = False

If PosX <> -PanelWidth Then
PosX = PosX - 1
Else
PosX = PanelWidth
End If

frmTCPTest.xDLite1.SendMessage PosX, PosY,
frmTCPTest.cboMQFColor.ListIndex, frmTCPTest.cboMQBColor.ListIndex,
MessageMQ

frmTCPTest.TimerMQ.Enabled = True

End Function
```

When this function is called, it will minus it's X coordinate and till the X reach the panel width limit, here we set -96.

X=-96,Y=0	LED Display Area	

All other scrolling effects are based on the same algorithm. You can try more interesting effects (like bounce, partial scrolling effects like jackpot) if you want.

VB Sample #6 Drawing sample

DLite ocx also provides some graphic function, let user could remotely draw graphic on the LED display. User could use those function to draw excel chart, or company logo on the LED display.

Here is a sample that demonstrates the drawing pixel function on the

LED display. This program using a Visual basic label component's mouse down and mouse up event.

When user click mouse button, will trigger mouse down event, and the mouse pointer cursors' X position and Y position will be mapped into remote LED display. Here using (X / 80) and (Y / 80)'s integer value to be the actual value to be send into LED display. The code could be divided into 3 parts.

- Network connection handling
- Mouse down event handling
- Mouse move event handling

The network connection part is the same with other sample in previous section. You could refer earlier sample for network connection programming.

When user mouse down event happened, then following code was executed.

```
bIMouseDown = True  
Call DrawPoint(X, Y)
```

The bIMouseDown is a Boolean flag that used for record the mouse button status. The DrawPoint function is actual painting function, used for draw local windows form label or remote LED display single pixel.

```
Public Sub DrawPoint(X, Y)  
If (Int(X / 80) < 96 And Int(Y / 80) < 16) Then  
    If bIConnected Then  
        If idx = 0 Then  
            Call DrawClear  
        End If  
  
        If idx > 2000 Then  
            idx = 0  
            Call DrawClear  
            lblPointCount.Caption = Str(idx)  
        End If  
    End If  
End Sub
```

```
        idx = idx + 1
        lblPointCount.Caption = Str(idx)
        Circle (X, Y), 30, RGB(255, 0, 0)
        XD Lite1.Pixel Int(X / 80), Int(Y / 80),
Int(cboDrawColor.ListIndex)
    End If
End If
End If
End Sub
```

The point number of this sample program was limited in 2000 points, so the draw function will check the point number first. If the number of points is over 2000 points, then the label and the LED display will be refreshed.

If the point total number is less than 2000 points, then using circle function to draw a red circle on the local label. Then the sample is using the xDLite's pixel function to draw a remote point on LED display. The pixel color setting is come from the drop down combo box.

VB Sample #7 Upload sample

DLite API provides upload file and download file function; you could upload necessary script and bmp file into LED display. DLite API also provides **RESET** function that could help you to reset LED display.

This sample upload 4 bmp files and 1 script file to LED display. The program will check upload process, in order to make sure each file is successfully uploaded.

When you upload the files, you must give specific file name with file directory. Following is the most important part for upload download sample.

```
Private Sub cmdUpload_Click()
    If blConnected Then
        'XD Lite1.Reset
```

```
Dim rtn1, rtn2, rtn3, rtn4, rtn5 As Integer
    rtn1 = XDLite1.UploadFile(txtScript.Text)
    If rtn1 Then rtn2 = XDLite1.UploadFile(txtFile1.Text)
    If rtn2 Then rtn3 = XDLite1.UploadFile(txtFile2.Text)
    If rtn3 Then rtn4 = XDLite1.UploadFile(txtFile3.Text)
    If rtn4 Then rtn5 = XDLite1.UploadFile(txtFile4.Text)
    If rtn5 Then
        XDLite1.Clear
        XDLite1.SendMessageS 0, 0, 3, 0, "Upload ok"
        XDLite1.SendMessageS 0, 9, 1, 0, "Demo Ready"
        txtSystemStatus.Text = "Upload Success" & vbCrLf &
txtSystemStatus.Text
    End If
End If
End Sub
```

In the sample, when upload button was clicked, the system will check the connection status first, if connection exist, then use UploadFile method to upload file. The rtn value > 0 means upload previous action is successful.

This code sample also adds a **Clear Display** button used for clean LED display. Because LED will not automatically play script until the last data on LED display cleaned. This is also very important step for remote data update. You should always clean the LED display after upload your newest script or BMP file.

VB Sample #8 QC sample

The QC sample use Sample #7 Upload/Download sample's files, so you must upload 4 bmp files in previous sample, then you could see the correct sample effect.

This sample shows how to remote trigger BMP files, and overlap text message on BMP files. Each of the command button send 3 different command. First command is trigger a Bitmap file on LED, 2nd and 3rd line are for the small text display.

```
frmQC.XDLite1.DisplayBMP 0, 0, 1, 0, frmQC.txtQclItemBmp.Text
frmQC.XDLite1.SendMessageS 34, 0, 3, 0, frmQC.txtQclItem.Text
frmQC.XDLite1.SendMessageS 34, 9, 3, 0, frmQC.txtQclItemVersion.Text
```

QcSampleToday(), QcSampleInspected(), QcSampleNotPassed() , all follow same logic. You can refer earlier sample, to complete the programming.

About eSoftsystem Corp.

eSoftsystem Technology Corp. is the most innovative embedded solution provider. It has built up a team of world experts in embedded software and hardware systems to provide customer high-performance and high-quality embedded solution product and services.