



## DMD Discovery™ Visual Basic .NET™

---

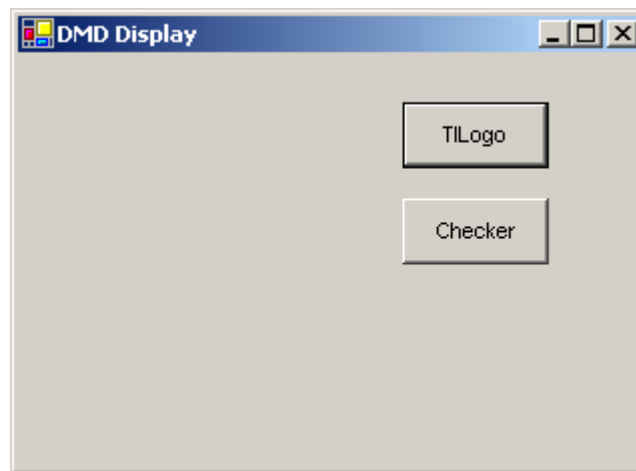
### ABSTRACT

Programming sample for control of the DMD Discovery 1100 (D1100) is provided written in Microsoft Visual Basic .NET. The sample demonstrates the use of the Discovery ActiveX control in performing common control functions.

### Getting Started

Be sure you have installed the Discovery board software on your system.

This sample is written in Microsoft Visual Basic .NET and provides a simple demonstration of image file to DMD load and display using the Discovery ActiveX control. Source code is available in the “VBNET ActiveX Sample Code” folder in the D1100Sample.zip file. The sample opens communication to the DMD board and displays images as selected by two control buttons :



The Discovery 1100 ActiveX control requires VARIANT\* arguments for some methods. Visual Basic .NET does not readily support VARIANT\* data types. Additional methods have been added to the D1100 ActiveX control to provide compatibility :

[short](#) CreateMemBuf()

Method *CreateMemBuf* is called to create storage for an image. This method allocates memory for an image up to 1024 x 768 pixels. The method returns a buffer index

(1.....# buffers) which is used to select the memory buffer to use for subsequent operations. Up to 1000 image buffers may be created using this method assuming sufficient system memory is available. Returned values are 1 if successful or 0 if unsuccessful.

**short** LoadImageFileToMemBuf(LPCTSTR FileName, **short** Bufnum, **short** MirrorImage)

Method *LoadImageFileToMemBuf* is used to load the memory buffer selected by input *Bufnum* with the specified image. The image is converted to a format which can be sent directly to the DMD. The image is read from a standard bmp, jpg, or gif image file, indicated by *FileName* . A mirror image may be created by setting *MirrorImage* to a non-zero value. Returned values are 1 if successful or 0 if unsuccessful.

**short** LoadFrameBufferFromMemBuf(**short** Bufnum)

Method *LoadFrameBufferFromMemBuf* loads the ActiveX control's frame buffer with the desired image from the memory buffer selected by *Bufnum*. Returned values are 1 if successful or 0 if unsuccessful.

**short** DeleteMemBuf()

Method *DeleteMemBuf* is called to free the memory buffers. Returned values are 1 if successful or 0 if unsuccessful.

To load and display an image with Visual Basic .NET you must use Version 2.0.0.3 or later of the Discovery ActiveX control. To check the version of the control find the file DDC\_Ctrl.ocx in the Windows/System32 folder using Windows Explorer. Right click on the filename and select Properties, then click the Version tab. Current versions of Discovery software are available on the Discovery 1100 support website.

The sample application DMDDisplay demonstrates the use of the ActiveX methods in loading and displaying image data.

The Form1.vb Load event calls Init\_USB to open the USB communication with the D1100 and Init\_Images to prepare the image buffers.

```
Private Sub Form1_Load(ByVal eventSender As System.Object, ByVal eventArgs As System.EventArgs)
    Handles MyBase.Load

    Init_USB() 'open USB connection to D1100
    Init_Images() 'prepare memory image buffers

End Sub
```

Form1.vb function Init\_USB() initializes the D1100 USB. The GetDevice call attaches the device and the EnableParallel call configures the device for USB control.

```
Private Sub Init_USB()  
  
    DDC1100Ctrl1.GetDevice() 'Attach device.  
    DDC1100Ctrl1.EnableParallel(0) 'Select USB mode  
  
End Sub
```

Function Init\_Images sets the image path, creates image storage and converts and loads the images to memory. Images TILOGO.bmp and checker1.bmp are loaded.

```
Private Sub Init_Images()  
    Dim ret As Short  
  
    DDC1100Ctrl1.AppPath = VB6.GetPath 'set path for image file location  
  
    Tlbufnum = DDC1100Ctrl1.CreateMemBuf() 'create memory image buffer  
    If Tlbufnum < 1 Then  
        MsgBox("Error creating image buffer")  
    End If  
  
    'convert image to DMD format and store in memory image buffer  
    ret = DDC1100Ctrl1.LoadImageFileToMemBuf("TILOGO.bmp", Tlbufnum, 0)  
    If ret <> 1 Then  
        MsgBox("Error loading image buffer")  
    End If  
  
    CHKRbufnum = DDC1100Ctrl1.CreateMemBuf() 'create memory image buffer  
    If CHKRbufnum < 1 Then  
        MsgBox("Error creating image buffer")  
    End If  
  
    'convert image to DMD format and store in memory image buffer  
    ret = DDC1100Ctrl1.LoadImageFileToMemBuf("checker1.bmp", CHKRbufnum, 0)  
    If ret <> 1 Then  
        MsgBox("Error loading image buffer")  
    End If  
  
End Sub
```

When a control button is clicked the image is displayed by the button event handling subroutine. The selected image is displayed using the LoadFrameBufferFromMemBuf and LoadResetFrame methods.

```
Private Sub CommandTI_Click(ByVal eventSender As System.Object, ByVal eventArgs As  
    System.EventArgs) Handles CommandTI.Click
```

```
Dim ret As Short
```

```
'load from memory image buffer to ActiveX image buffer
```

```
ret = DDC1100Ctrl1.LoadFrameBufferFromMemBuf(TIbufnum)
```

```
If ret <> 1 Then
```

```
    MsgBox("Error loading image")
```

```
End
```

```
End If
```

```
ret = DDC1100Ctrl1.LoadResetFrame() 'load to DMD and reset
```

```
End Sub
```

```
Private Sub CommandChecker_Click(ByVal eventSender As System.Object, ByVal eventArgs As
```

```
System.EventArgs) Handles CommandChecker.Click
```

```
Dim ret As Short
```

```
'load from memory image buffer to ActiveX image buffer
```

```
ret = DDC1100Ctrl1.LoadFrameBufferFromMemBuf(CHKRbufnum)
```

```
If ret <> 1 Then
```

```
    MsgBox("Error loading image")
```

```
End
```

```
End If
```

```
ret = DDC1100Ctrl1.LoadResetFrame() 'load to DMD and reset
```

```
End Sub
```

The Finalize event calls DeleteMemBuf to free all image buffers.

```
Protected Overrides Sub Finalize()
```

```
DDC1100Ctrl1.DeleteMemBuf() 'free all memory image buffers
```

```
MyBase.Finalize()
```

```
End Sub
```