Application Note

DMD Discovery™ 1100 LabVIEW

ABSTRACT

This document details how a user can make an ActiveX™ call using LabView™ and provides sample code for loading and displaying an image on the Discovery 1100.

Getting Started

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

Open LabView (6.x or later).

Create a new “Blank VI”.

Creating an Active Reference to the Discovery dll [Front Panel].

On Front Panel open the “Controls” palette [right-click].

Select the “All Controls” sub-palette.

Select the “Containers” sub-palette.

Select the “ActiveX Container” control and place it [drag it] on your Front Panel.

Right-click [Context Menu] on the ActiveX Container Box and select “Insert ActiveX Object. .”.

In the “Select ActiveX Object” dialogue box in the first dropdown box select “Create Control” [it should be the default].

In the list box scroll down to and select “Discovery 1100 ActiveX” and then press the “OK” button.

Inserting an “ActiveX Invoke Node” in the Block Diagram.

Switch to [or open] the Block Diagram.

Open the “Functions Palette” [right-click].

Select the “All Functions” sub-palette.

Select the “Communication” sub-palette.

Select the “ActiveX” sub-palette.

Drag the “Invoke Node” to the Block diagram.

Wire the “ActiveX Container” created previously to the “reference” input terminal [on left side] of the “Invoke Node”.

Left-click on the “Method” box in the “Invoke Node” to choose the method you wish to call.

For demo, select the “AboutBox” method and then run the VI. You should receive the “Discovery 1100 ActiveX Control, Version 1.0” splash box.

Adding more nodes.

If you wish to add more nodes either copy and past the current node or get another one from the palette and place it on the Block Diagram.

DO NOT create another ActiveX container to wire to the “reference” input terminal. Instead wire the “dup reference” [Now called “DDC1100Ctrl”] output terminal [on the right] of the first “Invoke Node” to the “reference” input terminal of the second “Invoke Node”.

Now select as before the method you want to use for the second node.

Continue this process for as many “Invoke Nodes” as you need.

Notes

Before you can use certain methods, you must use the “GetDevice” method [this can be thought of as an “Open Device” function]. Please refer to the TI DN 2506021 *Discovery 1100 Controller Board GUI User's & Programmer's Guide* regarding this.

Some methods have outputs [i.e. any of the “Get” Methods] and some methods also require input(s) [i.e. the “LoadBlock” or the “LoadImageFileToBuffer” method]. Please refer to the TI TI DN 2506021 *Discovery 1100 Controller Board GUI User's & Programmer's Guide* for specifications of the inputs to those methods that require an input(s).

Image Load and Display

The Discovery 1100 ActiveX control requires VARIANT* arguments for some methods. LabView does not support VARIANT* data types. Additional methods have been added to the control to provide LabView™ 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 LoadFramBufferFromMemBuf 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 LabView 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 D1100 Sample.vi demonstrates the use of the ActiveX methods in loading and displaying image data.

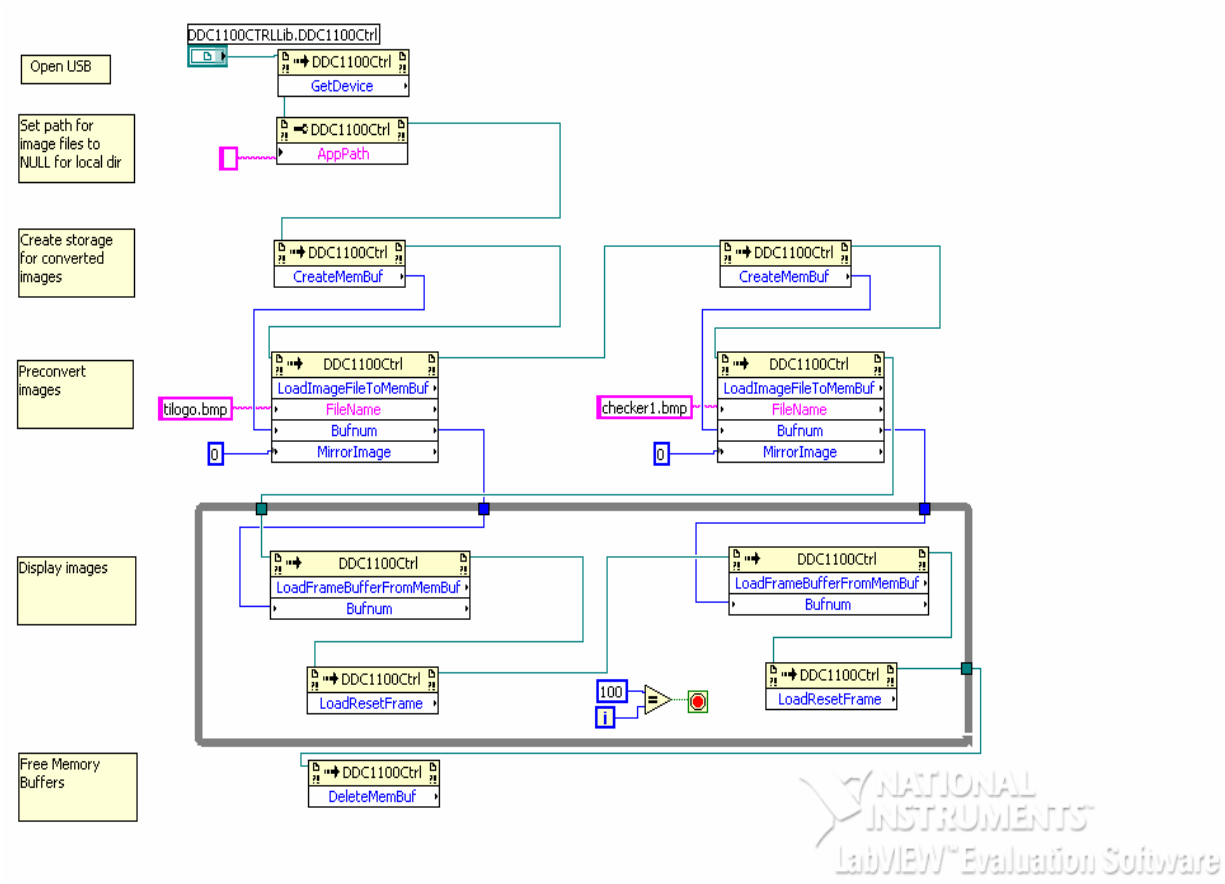

Figure 1 D1100 Sample.vi Block Diagram

The application begins with a call to method *GetDevice* to initialize USB communication. The attribute *AppPath* is then set to NULL to select images files from the local directory.

D1100 Sample.vi uses the *CreateMemBuf* and *LoadImageFileToMemBuf* methods to first load image tilogo.bmp, then to load image checker1.bmp.

The next operation uses methods *LoadFrameBufferFromMemBuf* to load the Active control's frame buffer with the desired image. Method *LoadResetFrame* is then used to load the image to the DMD and reset to display the image. *LoadFrameBufferFromMemBuf* and *LoadResetFrame* are repeated for both images. A while loop repeats the load and display of the two images 100 times. When the while loop is complete method *DeleteMemBuf* is called to free the memory buffers.

When using the methods described in this document care should be taken to check for errors in memory allocation and in using the correct *Bufnum* index value when loading images. It is possible to overwrite system memory when the methods are used improperly.

Trademarks