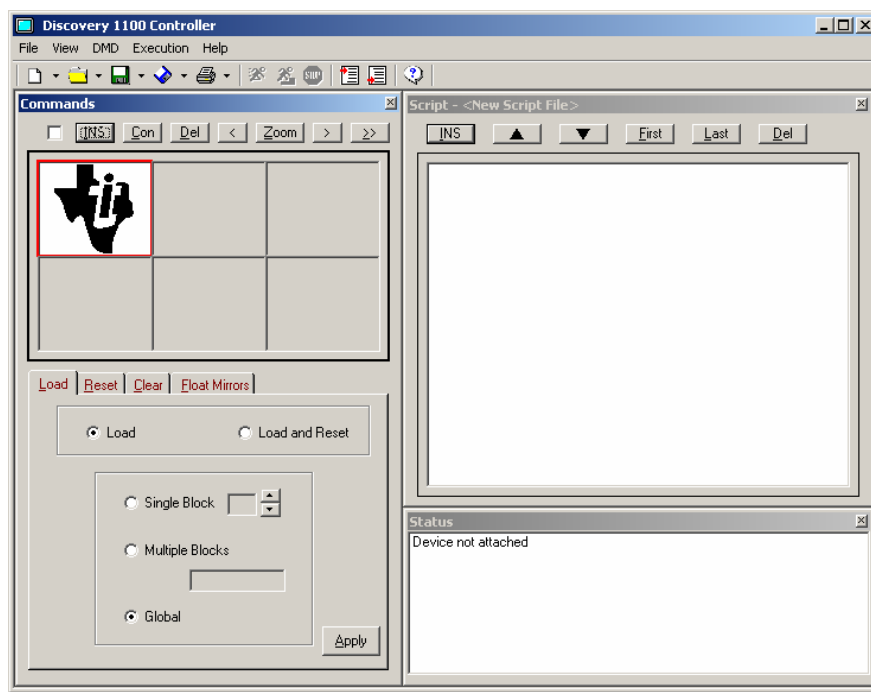


## DMD Discovery™ 1100 Controller Board GUI User's & Programmer's Guide

---

DMD Discovery



This manual describes the Graphical User Interface for the Discovery Controller Board. The Controller Board combines hardware, software, firmware, and documentation to form a stand-alone platform for use in developing and testing applications designed for using the Texas Instruments 0.7 XGA DDR DMD.

Revisions		
Rev	Descriptions	Date

## **IMPORTANT NOTICE**

Texas Instruments and its subsidiaries (TI) reserve the right to make changes to their products or to discontinue any product or service without notice, and advise customers to obtain the latest version of relevant information to verify, before placing orders, that information being relied on is current and complete. All products are sold subject to the terms and conditions of sale supplied at the time of order acknowledgment, including those pertaining to warranty, patent infringement, and limitation of liability.

In no event shall TI be liable for any special, incidental, consequential or indirect damages however caused, arising in anyway from the sale or use of the TI products. Products purchased from a TI authorized distributor are subject to the distributor's terms and conditions of sale.

TI warrants performance of its semiconductor products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements. Customers are responsible for their applications using TI components unless otherwise stated, this documentation and its intellectual content is copyrighted or provided under license and may not be distributed in any form without the express written permission of Texas Instruments Incorporated.

Certain applications using semiconductor products may involve potential risks of death, personal injury, or severe property or environmental damage ("critical applications"). TI semiconductor products are not designed, authorized, or warranted to be suitable for use in life-support devices or systems or other critical applications. Inclusion of TI products in such applications is understood to be fully at the customer's risk.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards must be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance or customer product design. TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used. TI's publication of information regarding any third party's products or services does not constitute TI's approval, warranty or endorsement thereof.

### **Trademarks**

Microsoft, Windows, Microsoft ActiveX, Windows XP and Windows 2000 are trademarks of Microsoft Corporation.

Other trademarks are the property of their respective owners.

## Notational Conventions

This document uses the following conventions.

The Graphical User Interface is also referred to as GUI.

Program listings, program examples, and interactive displays are shown as a *special typeface* similar to a typewriter's. Some examples use a **bold version** of the special typeface for emphasis; interactive displays use a **bold version** of the special typeface to distinguish commands that you enter from items that the system displays (such as prompts, command output, error messages, etc.).

Here is a sample program listing:

```
0011 0005 0001 .field 1, 2
0012 0005 0003 .field 3, 4
0013 0005 0006 .field 6, 3
0014 0006 .even
```

In syntax descriptions, the instruction, command, or directive is in a **bold typeface** font and parameters are in an *italic* typeface. Portions of the syntax that are **bold** should be entered as shown; portions of syntax that are in italics describe the type of information that should be entered. Syntax that is entered on a command line is centered. Syntax that is used in a text file is left justified.

Square brackets (**[and]**) identify an optional parameter. If you use an optional parameter, you specify the information within the brackets. Unless the square brackets are in a bold typeface, do not enter the brackets themselves.

# Table of Contents

<i>Section</i>	<i>Title</i>	<i>Page</i>
<b>1.0</b>	<b>Overview .....</b>	<b>1</b>
<b>2.0</b>	<b>Terms and Definitions .....</b>	<b>1</b>
<b>3.0</b>	<b>Software Overview .....</b>	<b>2</b>
3.1	DMD Image Control .....	2
3.2	Image Commands .....	2
<b>4.0</b>	<b>Discovery 1100 Controller Board ActiveX Interface .....</b>	<b>3</b>
4.1	Configuration/Status Methods .....	3
4.1.1	short AbortPipe(short <i>sEndpoint</i> ) – <i>new for the Discovery 1100</i> .....	3
4.1.2	Sub AboutBox() .....	3
4.1.3	AppPath As String .....	3
4.1.4	bAllowMessages As Boolean .....	3
4.1.5	short DownloadCode(LPCTSTR <i>FileName</i> ) – <i>new for the Discovery 1100</i> .....	3
4.1.6	short EnableParallel(short <i>EnablePar</i> ) .....	3
4.1.7	short GetConfigDescriptor(VARIANT* <i>DescriptorBuffer</i> ) .....	4
4.1.8	Function GetActivexRev() As Long .....	4
4.1.9	short GetDevice( ) .....	4
4.1.10	short GetDeviceDescriptor(VARIANT* <i>DescriptorBuffer</i> ) .....	5
4.1.11	Function GetDriverRev() As Long .....	5
4.1.12	short GetEndPointDescriptor(VARIANT* <i>DescriptorBuffer</i> , short <i>EPNumber</i> ) .....	6
4.1.13	short GetFirmwareRev( ) .....	6
4.1.14	Function GetRevision() As String .....	6
4.1.15	short GetSpeedMode() – <i>new for the Discovery 1100</i> .....	6
4.1.16	short GetUSBStatus(short <i>StatusType</i> , short <i>Index</i> ) .....	7
4.1.17	short GetVendorStatus( ) .....	7
4.1.18	BOOL IsDeviceAttached() – <i>new for the Discovery 1100</i> .....	7
4.1.19	short Reboot( ) .....	7
4.2	DMD Operation Methods .....	8
4.2.1	short Clear(short <i>BlockNum</i> ) .....	8
4.2.2	short ClearAndReset(short <i>BlockNum</i> ) .....	8
4.2.3	short ClearVariant(VARIANT FAR* <i>PVariant</i> ) .....	8
4.2.4	short ConvertImage (LPCTSTR <i>SrcFile</i> , LPCTSTR <i>DestFile</i> , short <i>MirrorImage</i> ) .....	8
4.2.5	short FloatMirrors( ) .....	8
4.2.6	short ImageToBlocks(LPCTSTR <i>ImageFile</i> , short <i>MirrorImage</i> ) .....	8
4.2.7	short LoadAndReset (short <i>BlockNum</i> ) .....	8
4.2.8	short LoadBlock (short <i>BlockNum</i> ) .....	8
4.2.9	short LoadFrame() – <i>new for the Discovery 1100</i> .....	9
4.2.10	short LoadFrameBuffer (VARIANT* <i>VarPBuffer</i> ) .....	9
4.2.11	short LoadImageFileToBuffer(LPCTSTR <i>FileName</i> , VARIANT* <i>ImageBuffer</i> , short <i>MirrorImage</i> ) .....	9
4.2.12	short LoadResetFrame() – <i>new for the Discovery 1100</i> .....	9
4.2.13	short Reset(short <i>BlockNum</i> ) .....	9
4.2.14	short ResetPipe(short <i>sEndpoint</i> ) – <i>new for the Discovery 1100</i> .....	9
4.3	ActiveX Control Usage Examples .....	9
4.3.1	Open USB Device .....	9
4.3.2	Display Single Image on DMD .....	9
4.3.3	Display Multiple Images on DMD .....	10
4.3.4	Clear Block on DMD .....	10
4.3.5	Clear Entire DMD Display .....	10
<b>5.0</b>	<b>User Interface Layout .....</b>	<b>10</b>
5.1	Menu Bar .....	11

5.1.1	File Menu .....	12
5.1.2	View Menu .....	12
5.1.2.1	Adjust Image .....	12
5.1.3	DMD Menu .....	13
5.1.3.1	Attach Device .....	13
5.1.3.2	Load Firmware .....	13
5.1.3.3	Diagnostics .....	14
5.1.4	Execution Menu .....	15
5.1.4.1	Run .....	15
5.1.4.2	Step .....	15
5.1.4.3	Next Step .....	15
5.1.4.4	Stop .....	15
5.1.4.5	Set Start .....	15
5.1.4.6	Set End .....	15
5.1.4.7	Configuration .....	16
5.1.5	Help Menu .....	16
5.2	Toolbar .....	16
5.2.1	File Menu Buttons .....	16
5.2.2	Operational Mode Controls .....	17
5.2.3	Run, Step and Stop Controls .....	17
5.2.4	Set Start and End Buttons .....	17
5.2.5	Help Button .....	17
5.3	The Commands Window .....	18
5.3.1	Inserting an Image .....	19
5.3.2	Mirror Images .....	19
5.3.3	Zoom .....	20
5.3.4	Load Tab .....	20
5.3.4.1	Load Command .....	20
5.3.4.2	Load and Reset .....	22
5.3.5	Reset Tab .....	22
5.3.5.1	Reset Command .....	23
5.3.6	Clear Tab .....	24
5.3.6.1	Clear .....	24
5.3.6.2	Clear and Reset .....	25
5.3.7	Float Mirrors .....	25
5.4	The Status Window .....	26
5.5	The Script Window .....	27
5.5.1	Inserting Commands .....	27
5.5.2	Moving Commands .....	27
5.5.3	Deleting Commands .....	27
<b>6.0</b>	<b>Discovery 1100 Operation .....</b>	<b>28</b>
6.1	USB Operation .....	28
6.2	The Configuration Dialog .....	30
6.3	Stepping Through the Script .....	30
6.4	Set Start and Set End .....	31
<b>7.0</b>	<b>Saving Scripts and Statuses .....</b>	<b>32</b>
7.1	Saving a Script .....	32
7.2	Saving a Status .....	32
<b>8.0</b>	<b>Printing Scripts and Statuses .....</b>	<b>32</b>
8.1	Printing a Script .....	32
8.2	Printing a Status .....	32
<b>9.0</b>	<b>Opening Scripts and Statuses .....</b>	<b>32</b>
<b>10.0</b>	<b>Creating New Scripts and Statuses .....</b>	<b>33</b>
10.1	Creating a New Script .....	33

10.2	Creating a New Status.....	33
------	----------------------------	----

# List of Illustrations

<i>Figure</i>	<i>Title</i>	<i>Page</i>
Figure 1.	Graphical User Interface Layout .....	11
Figure 2.	Menu Bar .....	11
Figure 3.	File Menu .....	12
Figure 4.	View Menu .....	12
Figure 5.	Adjust Image Option.....	12
Figure 6.	DMD Menu .....	13
Figure 7.	Firmware Select Source File Window .....	13
Figure 8.	Diagnostics Dialog .....	14
Figure 9.	Execution Menu .....	15
Figure 10.	Configuration Window .....	16
Figure 11.	Help Menu.....	16
Figure 12.	Toolbar .....	16
Figure 13.	Commands Window .....	18
Figure 14.	Inserting a Mirror Image .....	19
Figure 15.	Load Global Command .....	20
Figure 16.	Load Single Block Selection.....	21
Figure 17.	Multiple Block Selections .....	21
Figure 18.	Load and Reset Command .....	22
Figure 19.	Global Reset Command.....	23
Figure 20.	Reset Single Block Selection .....	23
Figure 21.	Global Clear Command.....	24
Figure 22.	Single Block Clear Command .....	24
Figure 23.	Clear and Reset Multiple Blocks .....	25
Figure 24.	Float Mirrors Command .....	25
Figure 25.	Status Window .....	26
Figure 26.	Script Window .....	27
Figure 27.	USB Operational Mode .....	28
Figure 28.	Image Box Selection .....	29
Figure 29.	Configuration Dialog.....	30
Figure 30.	Set Start and Set End Execution.....	31



## 1.0 Overview

This manual provides:

- A general description of the Discovery 1100 Board GUI
- An overview of the ActiveX interface commands
- Descriptions of menus, toolbars and display windows
- Operation instructions

## 2.0 Terms and Definitions

**DMD** Digital Micromirror Device

**GUI** Graphical User Interface

**USB** Universal Serial Bus

## 3.0 Software Overview

- The Discovery Controller Board GUI allows the user to control the DDC1100 Controller Board hardware via the USB by interfacing with an ActiveX control. The ActiveX control is designed to transmit data, send commands, and read the status of the controller board.
- The GUI provides you with access to the ActiveX control's implementations of the following commands:
  - Initialize the DMD
  - Float the mirrors
  - Read status
  - Load DMD memory block N
  - Load DMD memory block N and reset block N
  - Reset block N
  - Clear memory block N
  - Clear memory block N and reset block N

### 3.1 DMD Image Control

Images are controlled and displayed in blocks on the Digital Micromirror Device. Each image is broken down into 16 blocks, and each block consists of 48 rows of mirrors with 1024 mirrors in each row. That means that each block command controls just over 49,000 mirrors.

Blocks can be loaded and displayed individually, or as an entire image (all 16 blocks). Section 6.0 will explain the commands to control the different ways an image can be displayed.

Section 7.0 describes how to run a script of commands to control the displaying of an image. There are numerous combinations of display options available and this manual is not designed to cover them all. This user's guide will present an overview of how each command can be used to control an image.

### 3.2 Image Commands

- **Load** – Loads image blocks into memory.
- **Load and Reset** – Loads image blocks into memory and displays the contents on the DMD.
- **Reset** – Displays the contents of memory on the DMD.
- **Clear** – Clears the contents of memory.
- **Clear and Reset** – Zeroes the memory and displays the contents of memory on the DMD.
- **Float Mirrors** – Sets the mirrors to an unbiased or flat condition.

## 4.0 Discovery 1100 Controller Board ActiveX Interface

The Controller Board ActiveX control "DDC\_ctrl.ocx" provides an interface for sending messages, retrieving information and configuring the device. This control may be used to build applications in C, C++ and Visual Basic for control of the Discovery board. Refer to the Visual Basic source code in \DMD\Discovery1100\Source for an example of using the ActiveX control. The following sections briefly describe the ActiveX interface controls.

### 4.1 Configuration/Status Methods

#### 4.1.1 short AbortPipe(short *sEndpoint*) – *new for the Discovery 1100*

This method is used to cancel all pending requests on the specified endpoint. Returns a 1 if successful, or displays an error message and returns 0 if unsuccessful.

#### 4.1.2 Sub AboutBox()

This subroutine displays an about box showing the Active X control version and copyright information.

#### 4.1.3 AppPath As String

Attribute defining the path for file access. Read / Write. Direct access to this attribute is not required.

#### 4.1.4 bAllowMessages As Boolean

Attribute controlling display of error messages by Active X control. TRUE = enable messages, FALSE = disable messages. Default is TRUE.

#### 4.1.5 short DownloadCode(LPCTSTR *FileName*) – *new for the Discovery 1100*

This method will load the program specified by *FileName* into the EEPROM of the Discovery 1100. Returned values are 1 if successful or 0 if unsuccessful.

#### 4.1.6 short EnableParallel(short *EnablePar*)

Enables the device for Parallel Mode operation if *EnablePar* is equal to 1, or disables Parallel Mode operation if *EnablePar* is equal to 0. Returned values are 1 if successful or 0 if unsuccessful. Disabling Parallel Mode operation will automatically enable USB Mode operation.

#### 4.1.7 short GetConfigDescriptor(VARIANT\* DescriptorBuffer)

This method retrieves the configuration descriptor of the Discovery 1100. Returned values are 1 if successful or 0 if unsuccessful. Upon successful completion, the descriptor buffer contains the following information:

	Name	Description
Word 0	Length	The length (in bytes) of the descriptor.
Word 1	Descriptor Type	1 = Device Descriptor 2 = Configuration Descriptor 3 = String Descriptor 4 = Interface Descriptor 5 = Endpoint Descriptor 6 = Device Qualifier Descriptor
Word 2	Total Length	Total length of data returned for this configuration. Includes the combined length of all descriptors (configuration, interface, endpoint, and class- or vendor-specific) returned for this configuration.
Word 3	Number of Interfaces	Number of interfaces supported by this configuration
Word 4	Configuration Value	Value to use as an argument to the SetConfiguration() request to select this configuration
Word 5	Configuration	Index of string descriptor describing this configuration
Word 6	Attributes	The configuration attributes of this device. Refer to the USB specification for further information
Word 7	Max Power	Maximum power consumption of the USB device from the bus in this specific configuration when the device is fully operational. Expressed in 2 mA units (i.e., 50 = 100 mA).

#### 4.1.8 Function GetActivexRev() As Long

Returns control revision. Upper 16 bits contain major revision, lower 16 bits contain minor revision.

#### 4.1.9 short GetDevice( )

Opens handles to the device and communication pipes. Returns a 1 if successful, or displays an error message and returns 0 if unsuccessful.

#### 4.1.10 short GetDeviceDescriptor(VARIANT\* DescriptorBuffer)

This method retrieves the device descriptor of the Discovery 1100. Returned values are 1 if successful or 0 if unsuccessful. Upon successful completion, the descriptor buffer contains the following information:

	Name	Description
Word 0	Length	The length (in bytes) of the descriptor.
Word 1	Descriptor Type	1 = Device Descriptor 2 = Configuration Descriptor 3 = String Descriptor 4 = Interface Descriptor 5 = Endpoint Descriptor 6 = Device Qualifier Descriptor
Word 2	BCD USB	The major and minor versions of the USB specification (USB Spec 2.0 = 0x0200)
Word 3	Device Class	The USB Device Class code
Word 4	Device Subclass	The USB Device Subclass code
Word 5	Device Protocol	The USB Device class specific protocol code
Word 6	Max Packet Size	Maximum packet size for endpoint zero (only 8, 16, 32, or 64 are valid)
Word 7	Vendor ID	The Vendor ID of the Discovery 1100
Word 8	Product ID	The Product ID of the Discovery 1100
Word 9	Device	The device release number of the Discovery 1100
Word 10	Manufacturer	The index of string descriptor describing the manufacturer
Word 11	Product	The index of string descriptor describing the product
Word 12	Serial Number	The index of string descriptor describing the device's serial number
Word 13	Configurations	The number of possible configurations

#### 4.1.11 Function GetDriverRev() As Long

Returns driver revision. Upper 16 bits contain major revision, lower 16 bits contain minor revision.

#### 4.1.12 short GetEndPointDescriptor(VARIANT\* DescriptorBuffer, short EPNumber)

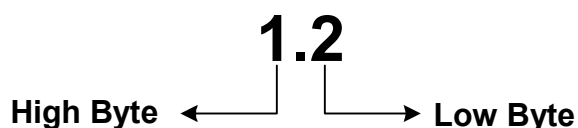
This method retrieves the endpoint descriptor of the endpoint indicated by EPNumber. Returned values are 1 if successful or 0 if unsuccessful. Upon successful completion, the descriptor buffer contains the following information:

	Name	Description
Word 0	Length	The length (in bytes) of the descriptor.
Word 1	Descriptor Type	1 = Device Descriptor 2 = Configuration Descriptor 3 = String Descriptor 4 = Interface Descriptor 5 = Endpoint Descriptor 6 = Device Qualifier Descriptor
Word 2	Endpoint Address	The address of the endpoint of the Discovery 1100 Refer to the USB specification for further information.
Word 3	Attributes	The attributes of the endpoint of the Discovery 1100 Refer to the USB specification for further information.
Word 4	Max Packet Size	The maximum packet size this endpoint is capable of transferring.
Word 5	Interval	The polling interval of this endpoint of the Discovery 1100. Refer to the USB specification for further information.

#### 4.1.13 short GetFirmwareRev( )

This command returns the firmware revision. The low byte of the returned value contains all the digits after the decimal point and the high byte contains all the digits before the decimal point.

**Example:**

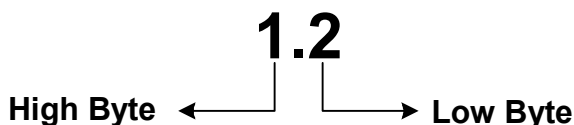


#### 4.1.14 Function GetRevision() As String

This command returns the firmware revision in decimal string format. The low byte of the returned value contains all the digits after the decimal point and the high byte contains all the digits before the decimal point.

**Example:** A returned string of "258" would be returned for revision 1.2 :

258 = 0x0102 ->



#### 4.1.15 short GetSpeedMode() – new for the Discovery 1100

The GetSpeedMode method is used to determine the speed at which the Discovery 1100 is operating. A return value of 1 indicates that the device is operating at high speed (USB 2.0), a value of 0 means it is operating at full speed (USB 1.0).

#### 4.1.16 short GetUSBStatus(short StatusType, short Index)

This method is called to read the USB status from the device. StatusType should be set to 1 for device status and 2 for endpoint status. Index is the index of the endpoint for which the status will be returned. The statuses returned are as follows:

Device Status		
Bit 7	SUCCESS	This bit is set to 1 to indicate a successful status has been read
Bit 6	---	Unused
Bit 5	---	Unused
Bit 4	---	Unused
Bit 3	---	Unused
Bit 2	---	Unused
Bit 1	RWUEN	This bit is set to 1 to indicate that remote wake-up is enabled.
Bit 0	SELF_PWR	This bit is set to 1 to indicate that the device is self powered. A 0 indicates that it is bus powered.

Endpoint Status		
Bit 7	SUCCESS	This bit is set to 1 to indicate a successful status has been read
Bit 6	NPAK2	NPAK2:0 represents the number of packets in the endpoint FIFO
Bit 5	NPAK1	
Bit 4	NPAK0	
Bit 3	FULL	This bit is set to 1 to indicate that the endpoint FIFO is full.
Bit 2	EMPTY	This bit is set to 1 to indicate that the endpoint FIFO is empty.
Bit 1	---	Unused
Bit 0	STALL	This bit is set to 1 to indicate that the endpoint is stalled.

#### 4.1.17 short GetVendorStatus( )

Returns a vendor specific status from the device. The lower byte of the returned value is interpreted as follows:

Vendor Status		
Bit 7	SUCCESS	This bit is set to 1 to indicate a successful status has been read
Bit 6	---	Unused
Bit 5	---	Unused
Bit 4	---	Unused
Bit 3	---	Unused
Bit 2	DEV_BUSY	This bit is set to 1 to indicate that the device is busy transferring data
Bit 1	EE_STAT	This bit is set to 1 to indicate that a valid bootable EEPROM is available.
Bit 0	EE_LOAD	This bit is set to 1 to indicate that the device program was loaded from the bootable EEPROM.

#### 4.1.18 BOOL IsDeviceAttached() – new for the Discovery 1100

Returns a TRUE (1) if a Discovery 1100 Controller board has been attached using the GetDevice method. Otherwise a FALSE (0) value is returned.

#### 4.1.19 short Reboot( )

This command forces the firmware to reboot. Returned values are 1 if successful or 0 if unsuccessful.

**Cannot be reproduced without permission from Texas Instruments Incorporated**

## 4.2 DMD Operation Methods

### 4.2.1 short Clear(short *BlockNum*)

This method is used to clear the contents of an entire block on the device. Specify the block to be cleared in *BlockNum*. Block numbers range from 1 to 16, a block number greater than 16 will signal a Global Clear. Returned values are 1 if successful or 0 if unsuccessful.

### 4.2.2 short ClearAndReset(short *BlockNum*)

Clears the contents of an entire block on the device, and resets the mirrors once the block has been cleared. Specify the block to be cleared and reset in *BlockNum*. Block numbers range from 1 to 16, a block number greater than 16 will signal a Global ClearAndReset. Returned values are 1 if successful or 0 if unsuccessful.

### 4.2.3 short ClearVariant(VARIANT FAR\* *PVariant*)

Frees the memory allocated by the *VARIANT* argument in methods which pass this argument. This should be called after the data returned by the method has been used. Returned values are 1 if successful or 0 if unsuccessful.

### 4.2.4 short ConvertImage (LPCTSTR *SrcFile*, LPCTSTR *DestFile*, short *MirrorImage*)

This method is used to convert a standard bmp, jpg, or gif image file, indicated by the file *SrcFile*, into a pre-processed image file which may be sent directly to the display. The pre-processed image is stored into the file indicated by *DestFile*. A mirror image may be created by setting *MirrorImage* to a non-zero value. Returned values are 1 if successful or 0 if unsuccessful.

### 4.2.5 short FloatMirrors( )

Places the DMD in a safe state with the mirrors in the "floated" or flat condition, with no bias applied to the DMD. Returned values are 1 for success or 0 if unsuccessful.

### 4.2.6 short ImageToBlocks(LPCTSTR *ImageFile*, short *MirrorImage*)

This method is used to process an image into a format which may be sent directly to the display. Setting *MirrorImage* to a non-zero value will instruct the method to construct a mirrored image. The image file must be in the form of a Bitmap file (bmp), a JPEG file (\*.jpg), a GIF file (\*.gif), or a DDC file (\*.ddc). The converted image is stored in the ActiveX control's image buffer. Returned values are 1 if successful or 0 if unsuccessful.

### 4.2.7 short LoadAndReset (short *BlockNum*)

Loads a block (6144 bytes of data) on the device and resets the mirrors after the block has been loaded. Specify the block to be loaded and reset in *BlockNum*. Block numbers range from 1 to 16. Image data is loaded from the ActiveX control's image buffer. Returned values are 1 if successful or 0 if unsuccessful.

### 4.2.8 short LoadBlock (short *BlockNum*)

Use this method to load a block of data (6144 bytes) on the device. Specify the block to be loaded in *BlockNum*. Block numbers range from 1 to 16. Image data is loaded from the ActiveX control's image buffer. Returned values are 1 if successful or 0 if unsuccessful.



#### **4.2.9 short LoadFrame()** – *new for the Discovery 1100*

This method is called to transfer the entire contents of the ActiveX control's image buffer to the display. Returned values are 1 if successful or 0 if unsuccessful.

#### **4.2.10 short LoadFrameBuffer (VARIANT\* VarPBuffer)**

LoadFrameBuffer is used to load the ActiveX control's buffer with the desired image. The image may be read directly from a file generated by the ConvertImage method. Returned values are 1 if successful or 0 if unsuccessful.

#### **4.2.11 short LoadImageFileToBuffer(LPCTSTR FileName, VARIANT\* ImageBuffer, short MirrorImage)**

LoadImageFileToBuffer is used to load memory buffer *ImageBuffer* with the desired 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*. Returned values are 1 if successful or 0 if unsuccessful.

#### **4.2.12 short LoadResetFrame()** – *new for the Discovery 1100*

This method is called to transfer the entire contents of the ActiveX control's image buffer to the display and resets the mirrors after the block has been loaded. Returned values are 1 if successful or 0 if unsuccessful.

#### **4.2.13 short Reset(short BlockNum)**

This method causes the mirrors in the block specified by BlockNum to change state. Block numbers range from 1 to 16, a block number greater than 16 will signal a Global Reset. Returned values are 1 if successful or 0 if unsuccessful.

#### **4.2.14 short ResetPipe(short sEndpoint)** – *new for the Discovery 1100*

This method is used to clear a stall condition on the specified endpoint. Returns a 1 if successful, or displays an error message and returns 0 if unsuccessful.

### **4.3 ActiveX Control Usage Examples**

Control sequences for common Discovery operations are presented below in programming language independent examples :

#### **4.3.1 Open USB Device**

The USB device must be opened prior to performing any DMD operations. After opening the device will remain open until the calling program ends.

```
DDC_Ctrl.GetDevice()           // Open channel to device
DDCCtrl.EnableParallel(0)      // Set USB operation
```

#### **4.3.2 Display Single Image on DMD**

A single image can be loaded from a standard image file and displayed on the DMD using the following methods :

DDC\_Ctrl.**ImageToBlocks**(*ImageFile*, *MirrorImage*) // Load ActiveX control's image buffer from specified standard bmp, jpg, or gif image file into the ActiveX control's display buffer  
DDCCtrl.**LoadResetFrame** () // Write Image to DMD and reset

#### **4.3.3 Display Multiple Images on DMD**

To maximize display speed images should be converted and loaded to memory buffer(s) prior to starting display -

Multiple images are first converted and loaded to memory buffers by repeating this method as needed:  
DDC\_Ctrl.**LoadImageFileToBuffer**(*FileName*, *ImageBuffer*, *MirrorImage*) // Convert image from specified standard bmp, jpg, or gif image file to a format which can be directly written to the DMD. Store converted file in memory buffer *ImageBuffer*.

An image is displayed from memory by passing a pointer to the memory buffer :  
DDCCtrl.**LoadFrameBuffer** (*VarPBuffer*) // load ActiveX control's image buffer from memory pointer *VarPBuffer*  
DDCCtrl.**LoadResetFrame** () // Write Image to DMD and reset

#### **4.3.4 Clear Block on DMD**

DDC\_Ctrl.**Clear**(*BlockNum*) // Clear block  
DDC\_Ctrl.**Reset**(*BlockNum*) // Reset block

Or

DDC\_Ctrl.**ClearAndReset**(*BlockNum*) // Clear and reset block

#### **4.3.5 Clear Entire DMD Display**

DDC\_Ctrl.**Clear**(17) // Global clear  
DDC\_Ctrl.**Reset**(17) // Global reset

Or

DDC\_Ctrl.**ClearAndReset**(17) // Global clear and reset

## **5.0 User Interface Layout**

The Controller Board GUI consists of a multiple display interface containing a menu bar, toolbar and three display windows: Commands Window, Script Window and Status Window.

The GUI can be separated into three functional areas. These are the Commands window, Script window and Status window. Each one is broken out and described in the paragraphs following the menu bar and toolbar descriptions.

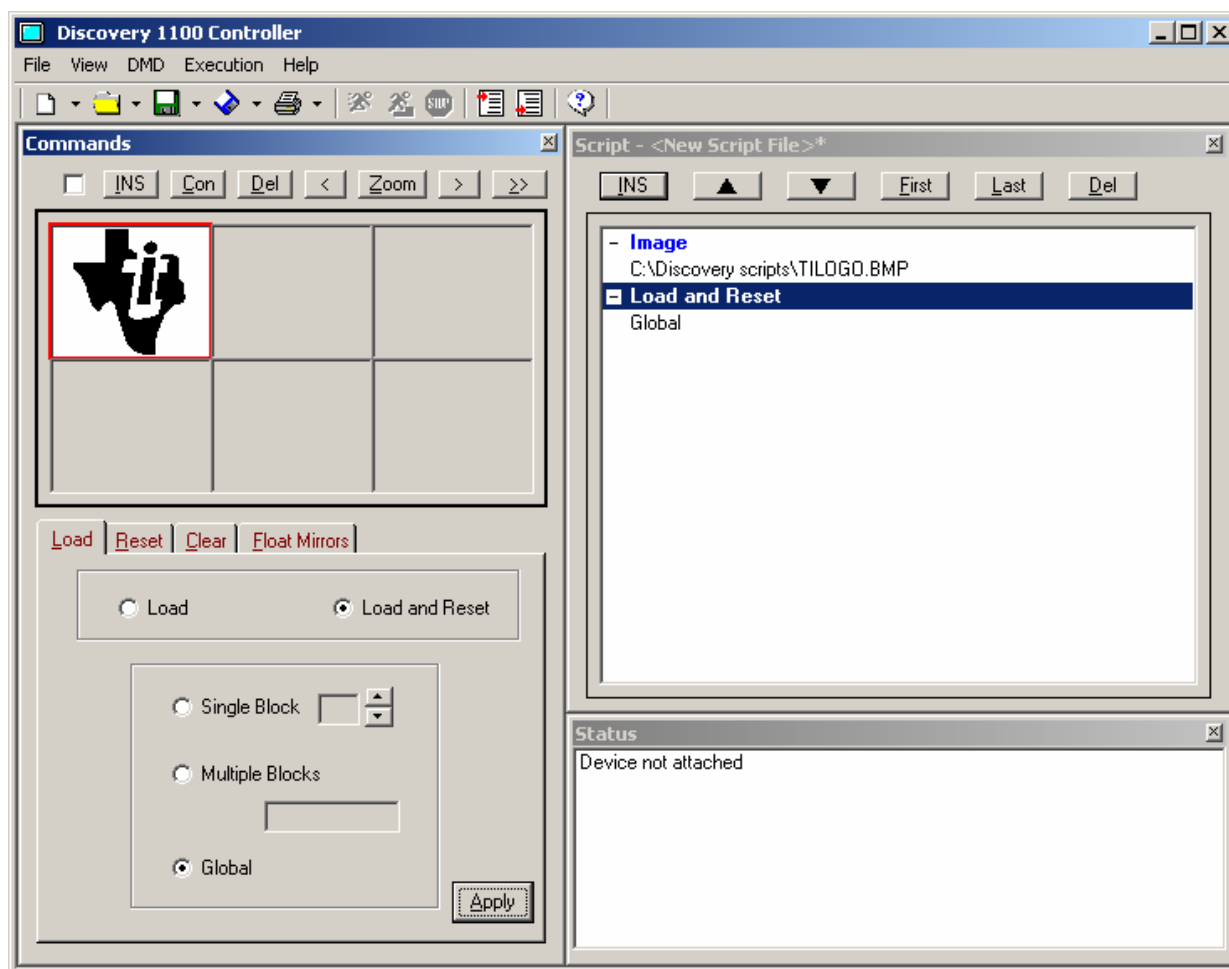


Figure 1. Graphical User Interface Layout

## 5.1 Menu Bar

A menu displays a list of commands that allow you to give instructions about what you want to do. The Controller Board GUI menu bar, Figure 2, is made up of File, View, DMD, Execution and Help menus. These menus are described in the following paragraphs.

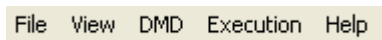


Figure 2. Menu Bar

### 5.1.1 File Menu

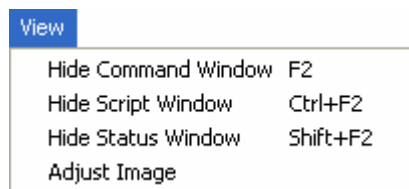
The File menu contains the standard New, Open, Save and Exit menu items with options to open a script or a status, and to save a script or a status.



**Figure 3. File Menu**

### 5.1.2 View Menu

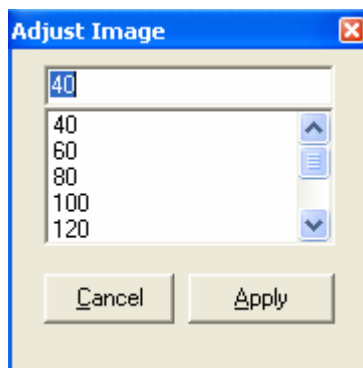
The View menu allows you to show or hide any of the display windows. You can also adjust the appearance of the image displayed in the GUI's Commands window.



**Figure 4. View Menu**

#### 5.1.2.1 Adjust Image

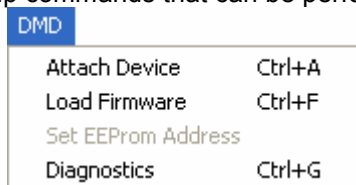
The Adjust Image option is used to set the cutoff value at which the applications decides whether a certain color on an image will be represented as white or black as the image is displayed in the Commands window. Adjusting the Image only affects the way an image is displayed in the GUI. It does not affect the way the image is displayed on the DMD device. The image adjustment must be set before the image is loaded and must be set each time the device is attached. Once set, the contrast will stay the same for all images loaded. The contrast ranges from 40 (light) to 200 (dark), and the default value is 40.



**Figure 5. Adjust Image Option**

### 5.1.3 DMD Menu

The DMD menu is comprised of set up commands that can be performed on the device.



**Figure 6. DMD Menu**

#### 5.1.3.1 Attach Device

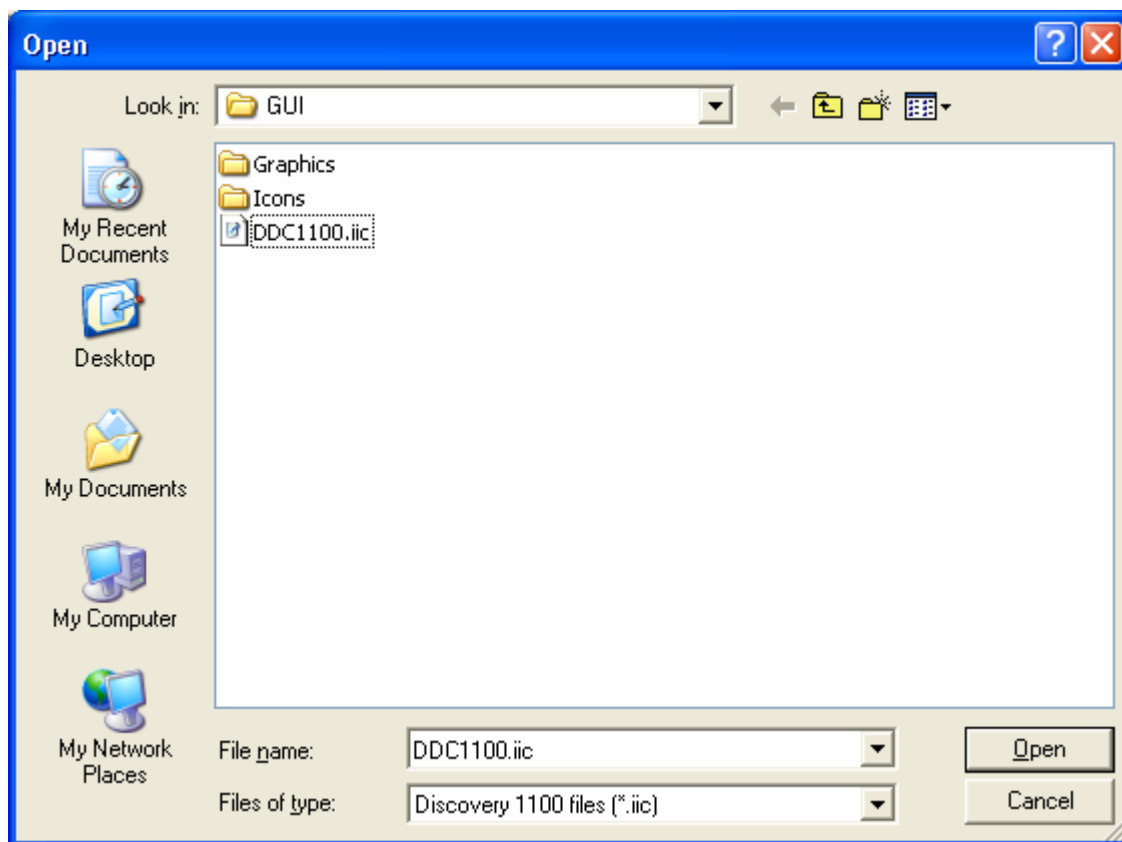
The Attach Device command is used to attach the GUI to the device at the initialization of the device, or after a power interrupt.

#### 5.1.3.2 Load Firmware

The Controller Board firmware is preloaded before the device is shipped. The Load Firmware command is used to load new versions of the firmware, or reload the current version of firmware, if it becomes necessary.

When this command is invoked, the Select Source File Window (Figure 7) will open for you to use to locate a version of firmware. The software will compare the currently loaded firmware against the selected source file and display a prompt to either notify you that the currently loaded firmware is newer than the selected file, or to ask you if you want to replace the existing file with the file selected.

This command causes the device to go off-line, and reattach if the load is successful.



**Figure 7. Firmware Select Source File Window**

### 5.1.3.3 Diagnostics

The Diagnostics command is used to retrieve information about the device such as descriptors, firmware revision and statuses both USB and vendor specific. The information retrieved by the Diagnostics Dialog is displayed in the Status Window.

The Diagnostics Dialog allows the user to Test the USB connection between the host and the device by retrieving USB device specific information needed to facilitate communication between the host and the DDC device. This Dialog is helpful when experiencing problems communicating with the device, such as failing to attach to the device or failing to read or write data to the device.

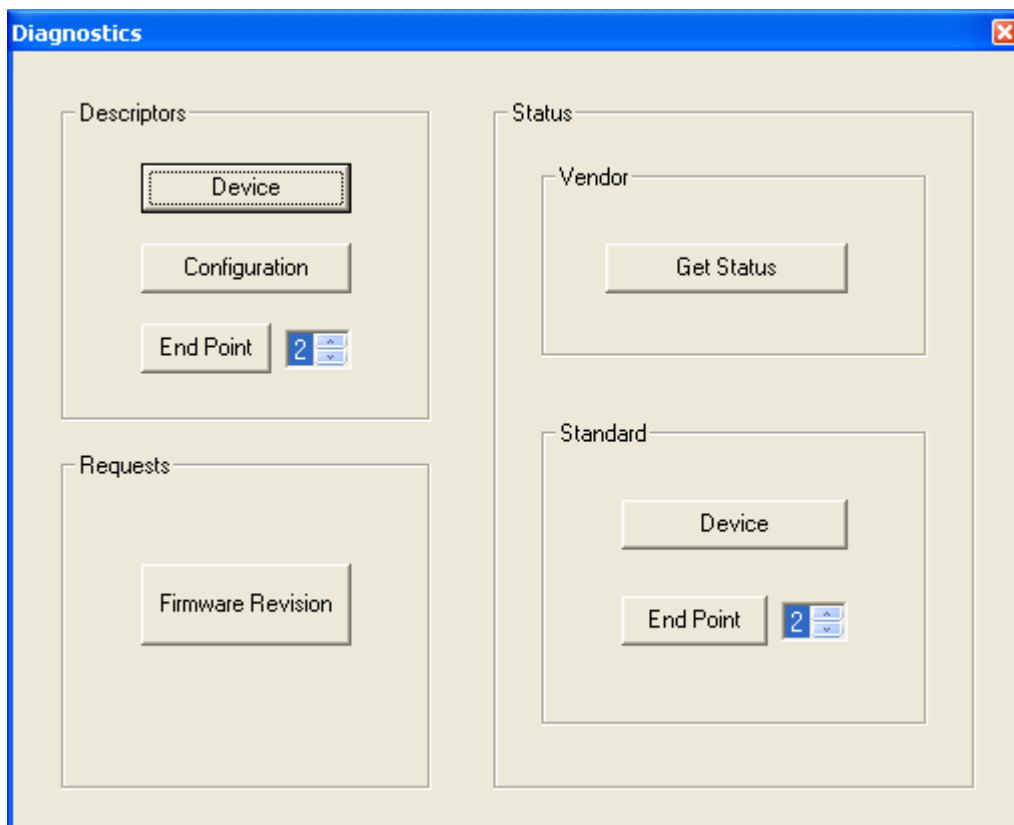


Figure 8. Diagnostics Dialog

The following information can be retrieved from the DDC device using the Diagnostic Dialog:

- **Descriptors:** A descriptor is a record format used to provide information about a standard part of a logical device such as a configuration or an interface. The Device Descriptor characterizes the device as a whole by providing information specific to the device such as the Vendor ID and the Product ID. The DDC contains only one configuration and the Configuration Descriptor provides information pertaining to the device configuration. There is an end point for each data pipe used by the host to communicate to the device. The End Point Descriptor provides information pertaining to each end point.
- **Requests:** The Firmware Revision Request is used to find the revision of the firmware running on the device. This request is useful when determining whether the device firmware requires upgrading.

- **Vendor Status:** The Get Status command retrieves the following firmware statuses:
  - Firmware is loaded to EEPROM
  - EEPROM is not programmed or there is a checksum error
  - Device is idle
  - Device is busy
- **Standard Status:** The Standard Device Status retrieves the power characteristics for the device. For the DDC, the power characteristic is set to Self Power. The Standard End Point Status retrieves the Stalled or Working status and the FIFO status for the specified end point.

#### 5.1.4 Execution Menu

The Execution menu contains script commands and configuration options.

Execution	
Run	F5
Step	F6
Next Step	F8
Stop	
Set Start	F3
Set End	F4
Configuration	F11

**Figure 9. Execution Menu**

##### 5.1.4.1 Run

Executes all the commands in the list without interruption.

##### 5.1.4.2 Step

Sets the execution of commands to one step at a time mode.

##### 5.1.4.3 Next Step

Executes commands sequentially, without executing any commands following it.

##### 5.1.4.4 Stop

Terminates the execution of the commands in the script.

##### 5.1.4.5 Set Start

Sets the start point of command execution within the script.

##### 5.1.4.6 Set End

Sets the end point of command execution within the script.

### 5.1.4.7 Configuration

Configuration allows you to set the delay, in milliseconds, between the end of one command and the beginning of another. You can also set the delay between the end of execution and the beginning of execution of a whole script when performing multiple iterations. The Loop region of the dialog allows you to choose between executing a set number of iterations on the script and executing the script continuously.

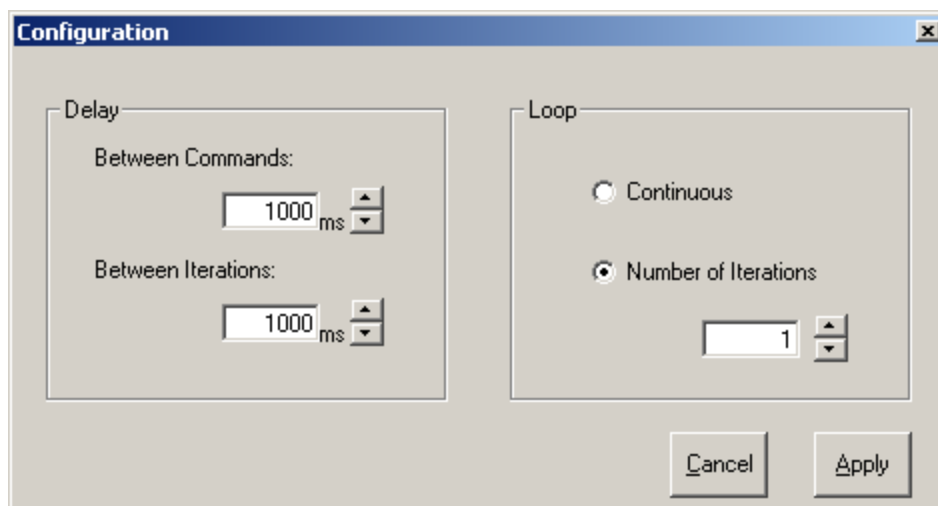


Figure 10. Configuration Window

### 5.1.5 Help Menu

The Help menu contains the standard link to the help contents and the command to load the About dialog box.

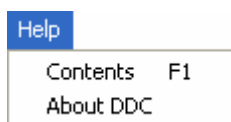


Figure 11. Help Menu

## 5.2 Toolbar

The Toolbar (Figure 12) contains buttons to provide you with shortcuts to perform certain operations.

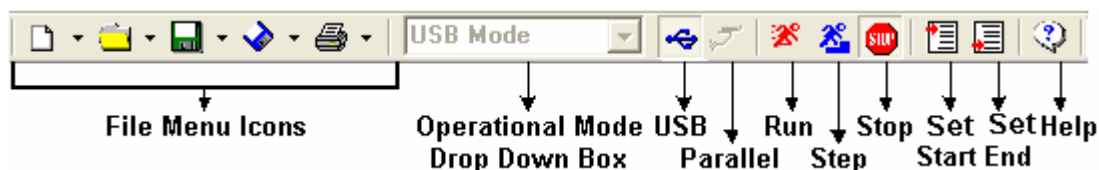


Figure 12. Toolbar

### 5.2.1 File Menu Buttons



The first five buttons in the toolbar, New, Open, Save, Save As and Print have the same function as the five items in the File menu.

### **5.2.2 Operational Mode Controls**

The next area of the toolbar is a drop down window to select the operational mode for the device. The two buttons immediately to the right of the drop down window are for selecting either the USB or Parallel operation modes. The current revision of the GUI is limited to 'USB Mode' only, therefore the drop down window and the Parallel button are disabled and the USB button will remain depressed.

### **5.2.3 Run, Step and Stop Controls**

Run, Step and Stop icons control the execution of the list of commands in the script window. The Run icon allows you to execute the commands in the list without interruption. The Step icon executes each command individually, without executing any commands following it. The Stop icon terminates the execution of commands in the script.

### **5.2.4 Set Start and End Buttons**

Set Start and Set End icons set the start point and end point of command execution within a script.

### **5.2.5 Help Button**

The Help button displays the help contents for the application.

### 5.3 The Commands Window

The Commands Window, Figure 13 consists of an Images region capable of displaying up to six images. There is also a series of Command Tabs, each displaying a command and the command options to be performed on the device.

An image must be added to a script before commands can be added. See Inserting an Image to learn how to insert, select and add an image to a script.

To view and select the options associated with a command, click on the corresponding command tab. Once you've selected the commands and options, click the **Apply** button to add the command options to the script.

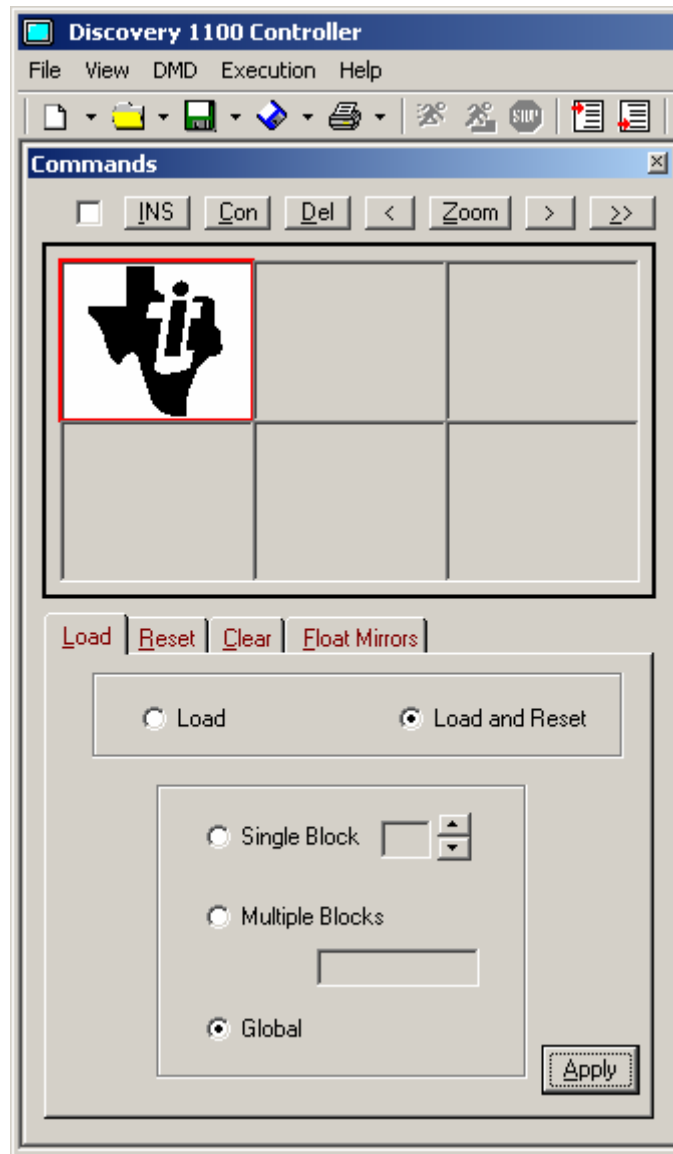


Figure 13. Commands Window

### 5.3.1 Inserting an Image

An image or a script containing an image must be selected before adding any commands to the Script Window. Select an image container from the Images Box located at the top of the Commands Window. You can select an image container by clicking on it, or by using the Left < and Right > Navigation Buttons, at the top of the Commands Window, to move the selector to the desired image box. The border around the selected container will change to red. Click the **INS** button on the Commands window.

When the common dialog box appears, select the image file, click on the **Open** button in the dialog box and the image will appear in the selected image box (the image box with the red border).

If the image appears too light or too dark, you adjust the contrast using the Adjust Image option in the View menu. Adjusting the image only affects the way an image is displayed in the GUI. It does not affect the way the image is displayed on the DMD device. The image adjustment must be set before the image is loaded and must be set each time the device is attached. If you have already loaded an image, you will have to reload it before the changes will take effect. Once set, the contrast will stay the same for all images loaded. The contrast ranges from 40 (light) to 200 (dark), and the default value is 40.

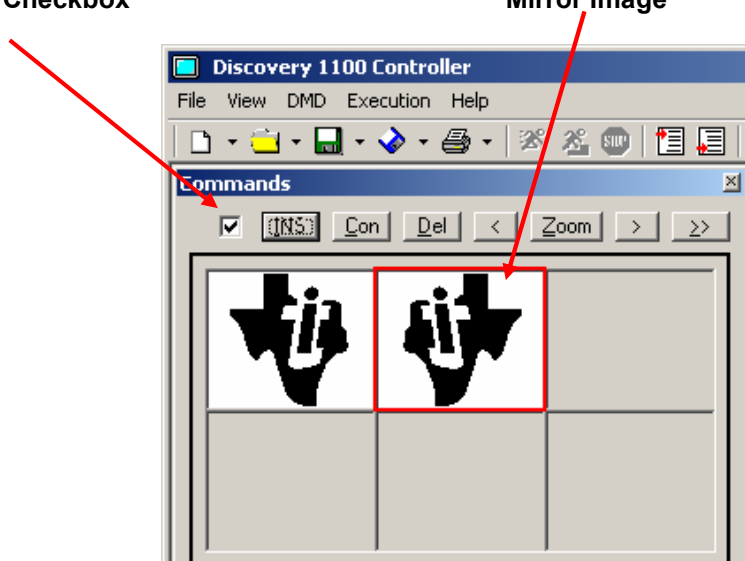
Once the image has been inserted, you will use the 'Add to Script' button **>>** to add the image file name and path to the Script Window. The image will be added to the script under the Image command and will be used to execute any commands that follow it on the script.

### 5.3.2 Mirror Images

To insert a mirror image, you would put a check in the mirror image checkbox (next to the **INS** button), as shown in Figure 14, select an image container for that image and click on the **INS** button.

**Mirror Image Checkbox**

**Mirror Image**



**Figure 14. Inserting a Mirror Image**

When the common dialog box appears, select the image file, click on the **Open** button in the dialog box and the image will appear in the selected image box.

Once the mirror image has been inserted, you can use the 'Add to Script' button **>>** to add the image to the Script Window. The image will be added to the script under the Image command and will be used to execute any commands that follow it on the script.

### 5.3.3 Zoom

An image can be enlarged using the **Zoom** button. When pressed, the selected image will fill the entire image container area with the single image. Press the zoom button again to return the image to its original state.

### 5.3.4 Load Tab

Load commands allow you to load an entire image into memory, or load just individual blocks (see Para. 4.1). The image blocks must be loaded into memory before they can be displayed on the DMD.

#### 5.3.4.1 Load Command

You can load all of the image blocks (**Global**) (see Figure 15), or you can choose to load them individually using the **Single Block** and **Multiple Blocks** options. Select the option you want to use and click the **Apply** button. (For more information on blocks, see Paragraph 4.1)

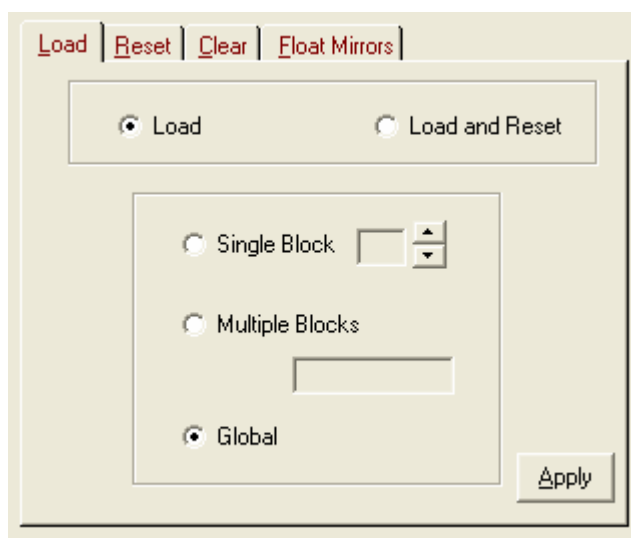
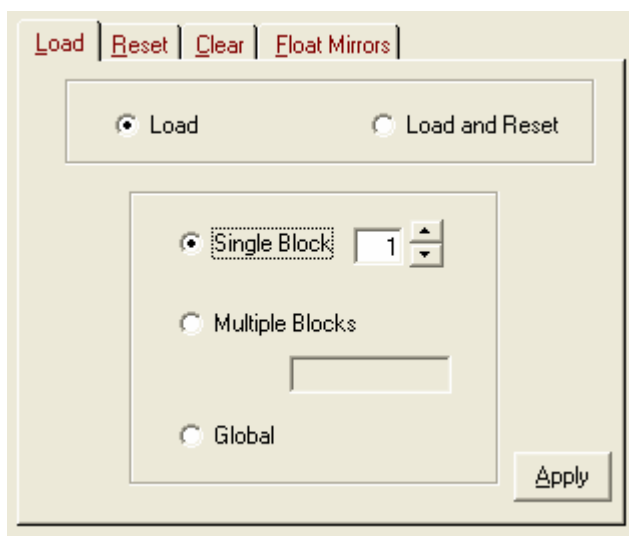


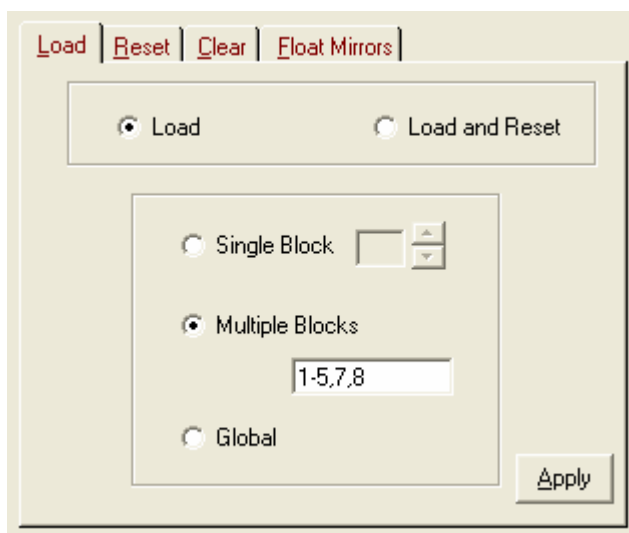
Figure 15. Load Global Command

Single Blocks are selected by clicking the up or down arrows until the desired block appears. Figure 16 shows Block 1 as the selected block.



**Figure 16. Load Single Block Selection**

The multiple blocks option of the load command can be expressed as a range (e.g., 1-5), or as a list of blocks separated by commas (e.g., 7, 8), as shown in Figure 17. Blocks must be within a range of 1 to 16.



**Figure 17. Multiple Block Selections**

#### 5.3.4.2 Load and Reset

Performs the same function as the Load command but also Resets the mirrors (displays the contents of memory) immediately following the completion of loading. Blocks are loaded and displayed one right after another, giving the image a scrolling effect. Figure 18 shows an example of a single block Load and Reset command. The multiple blocks option of the load command can be expressed as a range (e.g., 1-5), or as a list of blocks separated by commas (e.g., 2, 4, 7). Blocks must be within a range of 1 to 16.

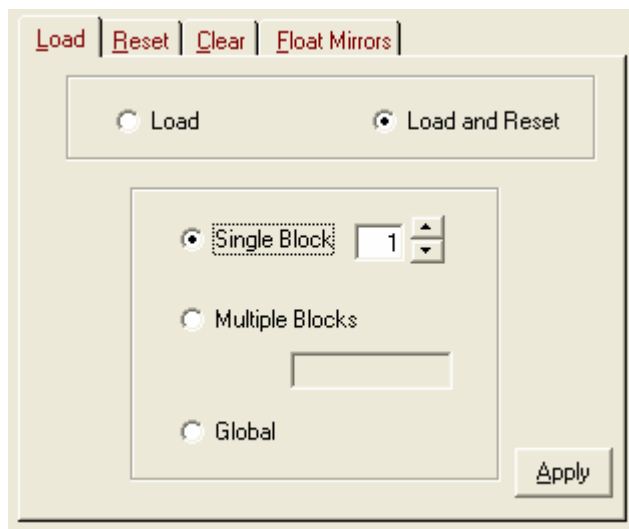


Figure 18. Load and Reset Command

#### 5.3.5 Reset Tab

Provides commands to allow you to reset (display) an entire image, or reset (display) individual blocks of the image.

### 5.3.5.1 Reset Command

The **Reset** command causes the mirrors to change from the current state to the state of that in memory. The contents of memory are determined by the Load or Clear commands. You can choose to reset all the blocks (**Global**, see Figure 19), or you can choose to reset blocks individually using the **Single Block** (Figure 20) and **Multiple Blocks** options. Select the option you want to use and click the **Apply** button.

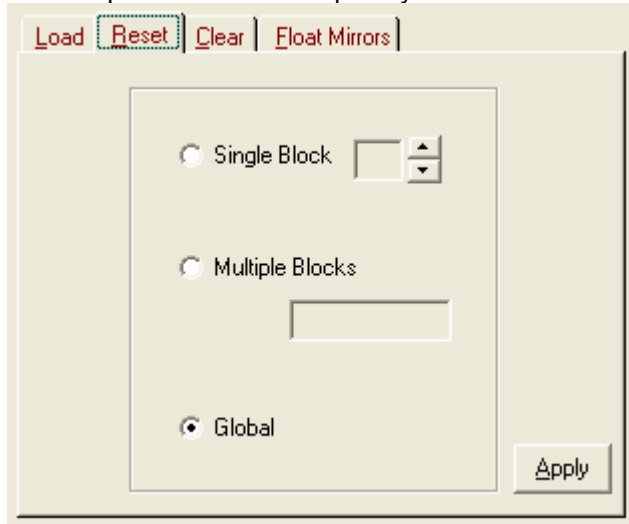


Figure 19. Global Reset Command

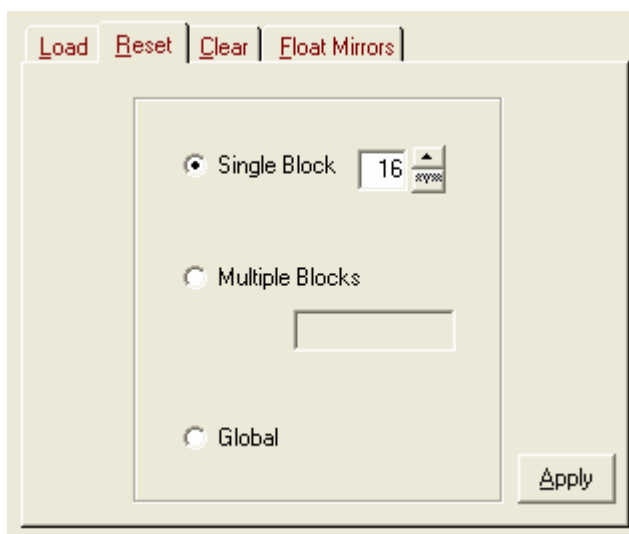


Figure 20. Reset Single Block Selection

The multiple blocks option of the reset command can be expressed as a range (e.g., 1-5), or as a list of blocks separated by comas (e.g., 2, 4, 7). Refer to Figure 17 for an example of how this is done.

### 5.3.6 Clear Tab

The Clear command allows you to clear the contents of memory, or write zeroes to memory and send them to the DMD.

#### 5.3.6.1 Clear

Clears the memory contents of all the blocks (**Global**, Figure 21) that make up an image in the device, or clear individual blocks using the **Single Block** (Figure 22) or **Multiple Blocks** options, by writing zeroes to the contents of memory. Select the option you want to use and click on the **Apply** button.

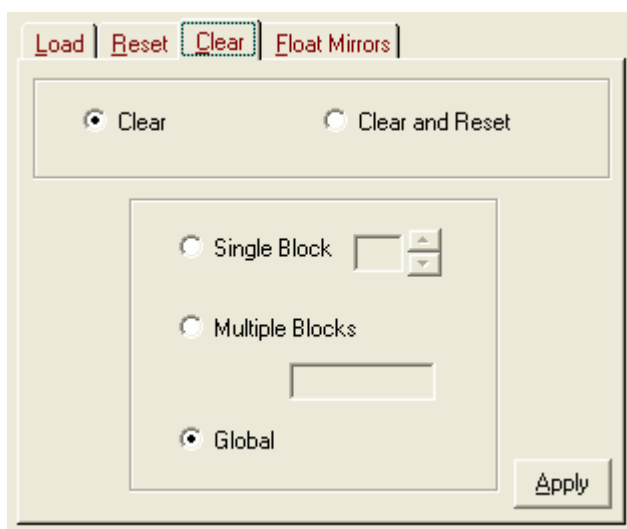


Figure 21. Global Clear Command

The multiple blocks option of the clear command can be expressed as a range (e.g., 1-5), or as a list of blocks separated by comas (e.g., 2, 4, 7). Refer to Figure 17 for an example of how this is done.

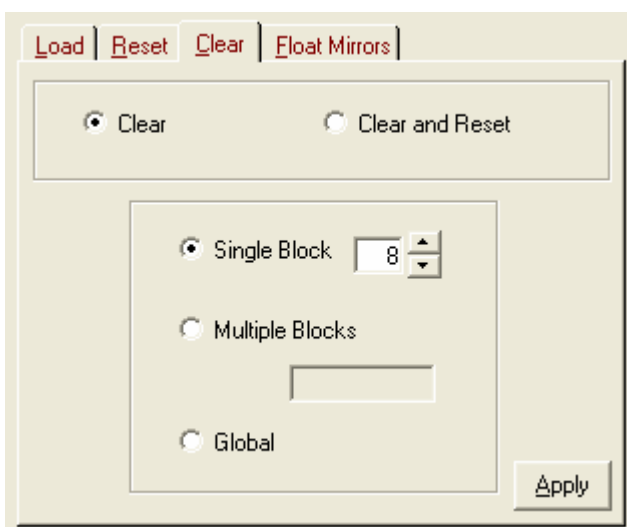


Figure 22. Single Block Clear Command



### 5.3.6.2 Clear and Reset

Performs the same function as the Clear command but also resets the mirrors (displays the contents of memory) immediately following the completion of the clear command. You can clear a single block, multiple blocks (Figure 23), or all the blocks that make up an image.

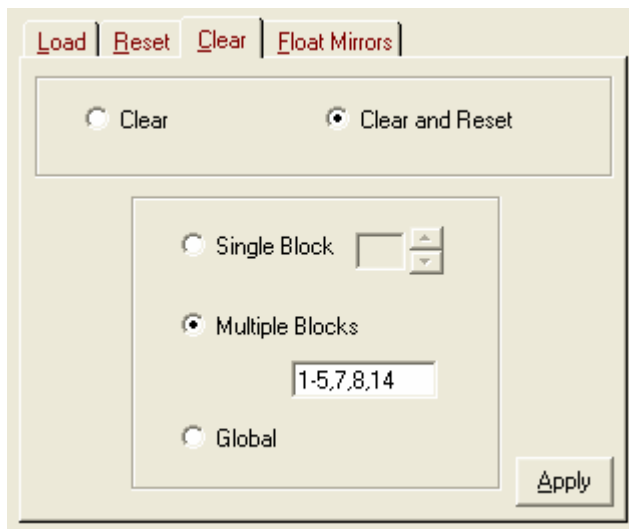


Figure 23. Clear and Reset Multiple Blocks

### 5.3.7 Float Mirrors

The Float Mirrors command places the DMD in safe state with the mirrors in the “floated” or flat condition and no bias applied to the DMD. This command executes automatically when power fails or the controller detects inactivity during an “expose” command lasting for 10 minutes or longer. Inactivity is defined as no resets being issued within this time period.

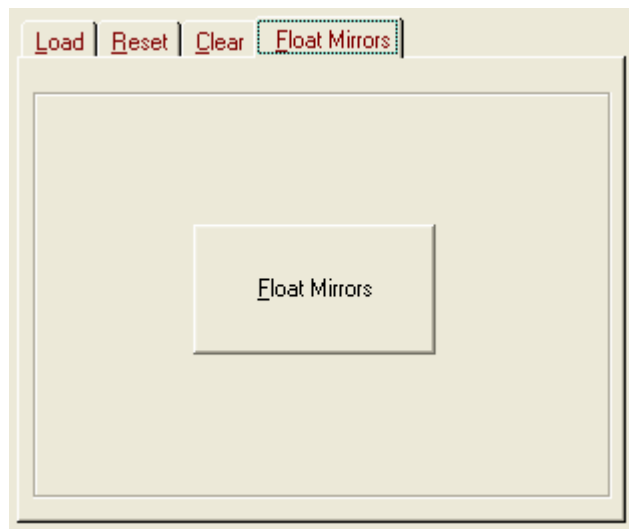


Figure 24. Float Mirrors Command

## 5.4 The Status Window

The Status Windows displays the execution status, retrieved information and any responses sent back to the host from the device after a command has been executed.

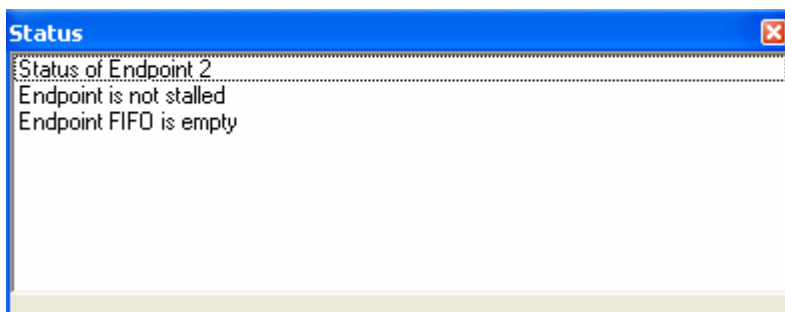


Figure 25. Status Window

## 5.5 The Script Window

The Controller Board GUI uses a Script window (Figure 26) to keep track of the images to be loaded and the commands to be executed on the DMD. Once you've added an image and the commands you want to execute to the script, you can change the command order, delete and insert commands at a specified location within the script.

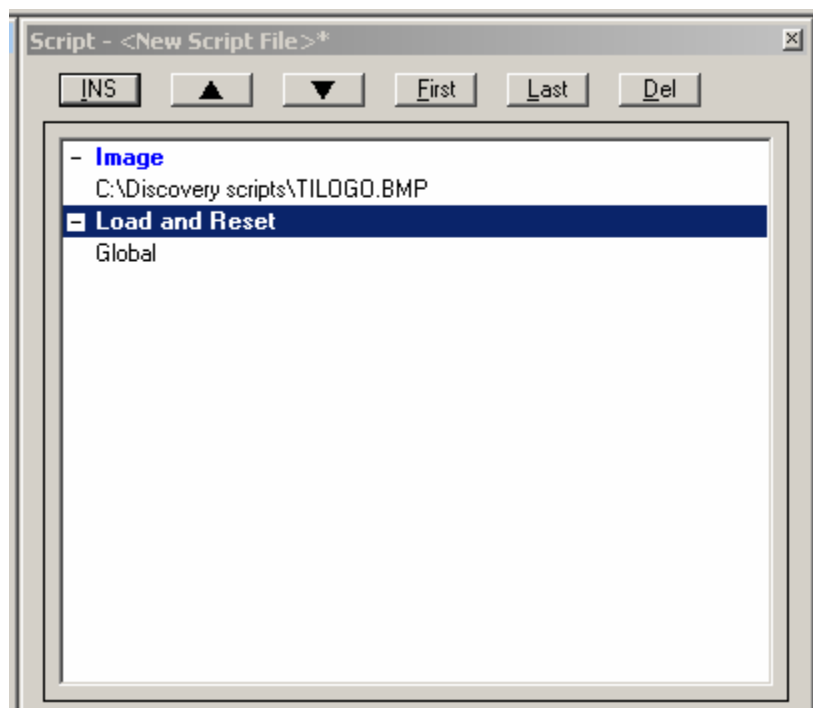


Figure 26. Script Window

The application executes commands in the order in which they are entered in the script. In order for the application to execute commands, the first command in the script must be the **Image** command followed by the correct path and file name for an existing image. See Inserting an Image to learn how to add an image to the script.

### 5.5.1 Inserting Commands

To insert a command in a specified location on the script, select the command from the Commands window, go to the desired location in the script and click the **INS** button at the top of the Script Window.

### 5.5.2 Moving Commands

To alter the command order, select the command you want to move by clicking on it, then click the **▲** (up) arrow or the **▼** (down) arrow to move it to the desired location.

You can also move a command to the first position in the script by using the **First** button, or move it the last position with the **Last** button.

### 5.5.3 Deleting Commands

To delete a command from the script, click on the command you want to delete and click the **Del** button at the top of the Script Window.

## 6.0 Discovery 1100 Operation

The Discovery 1100 is capable of operating in excess of 100 frames per second when connected to a host system's USB 2.0 port. The device will function correctly when connected to a USB 1.1 port, but at a greatly reduced speed. Additional items connected to the USB controller may also reduce the transfer rate.

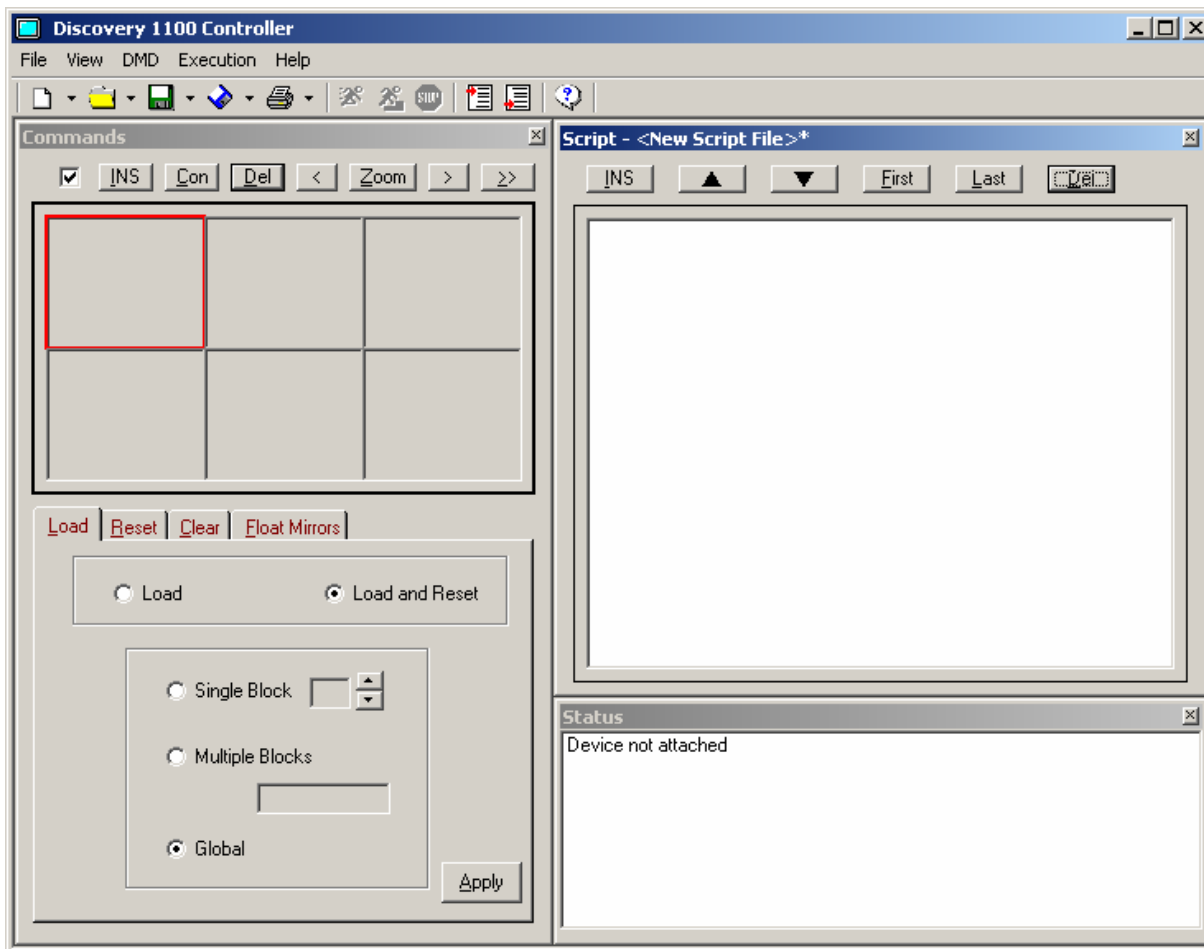


Figure 27. USB Operational Mode

### 6.1 USB Operation

The following steps must be followed in order to operate the device:

**Step 1:** If the device has not been attached, attach to the device using the Attach Device command from the DMD menu. If you encounter a problem attaching the device or if the device has not yet been attached, a message box will appear notifying you of the problem. You should check the device to make sure it is functioning. Also check the connection from the host to the device to make sure the two are properly connected.

**Note:** The device only has to be attached once in order for any commands to be executed. Once a device has been attached any number of scripts can be executed without having to reattach the device. If there is a power interrupt, you will have to attach the device once power is restored.

**Step 2:** Firmware is preloaded into the device. If you have a newer version of the firmware, load the firmware now by selecting Load Firmware from the DMD menu. When the file dialog is

**Cannot be reproduced without permission from Texas Instruments Incorporated**

Copyright Texas Instruments Incorporated

28

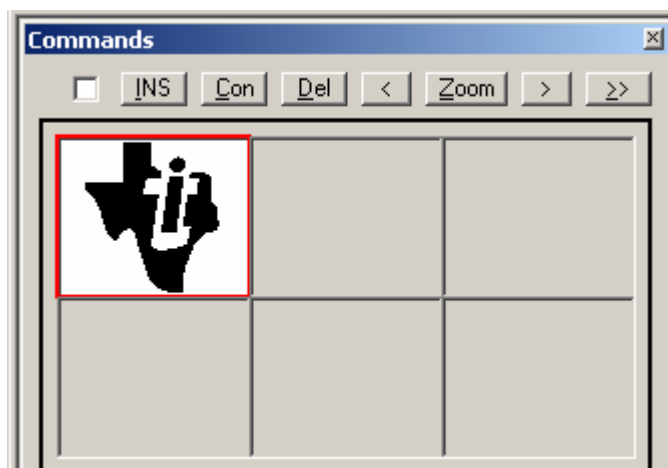
TI DN 2506021  
Rev. A

displayed, select the desired firmware file and click OK to download the firmware. Once download is completed, remove power from the device for at least five seconds, then apply power and reattach the device.


**Step 3:** Insert an image into the Commands Window. To insert an image:

**Step a:** Select an image container from the Images Box located at the top of the Commands Window. You can select an image container by clicking on it, or by using the Left < and Right > Navigation Buttons, at the top of the Commands Window, to move the selector to the desired image box. The border around the selected container will change to red.


**Note:** To insert a mirror image, you would first put a check in the checkbox next to the  button.







**Figure 28. Image Box Selection**

**Step b:** On the control buttons located on top of the Images region on the Commands Window, click the  button, a dialog box will appear asking you to select the image file to be inserted.

**Step c:** When the common dialog box appears, select the image file, click the **Open** button in the dialog box and the image will appear on the selected image box (the image box with the red border).

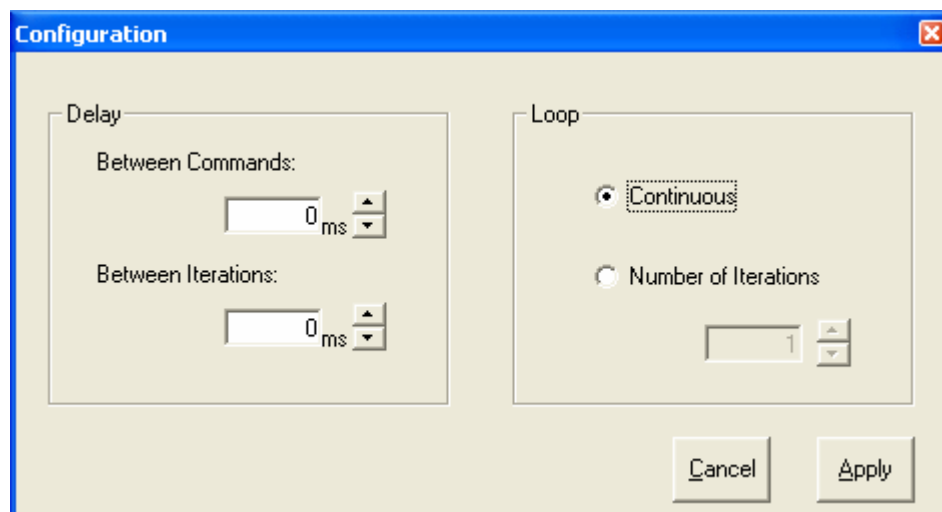
**Step d:** Once the image has been inserted, you can use the 'Add to Script' button  to add the image file name and path to the Script Window. The image will be added to the script under the Image command and will be used to execute any commands that follow it on the script.

**Step 4:** To insert another image, move the image selector to the next container using the navigation buttons, located on top of the image boxes, and insert the image as outlined above.

- Step 5:** Enter an image command in the script. Before any commands can be executed, an image command must be entered in the script. This image command contains the file path and name of the image file to be processed. Processing an image consists of loading the image into a buffer divided into blocks to be loaded into the device when any of the load commands are executed. To add the image to the script, use the 'Add to Script' button  to add the image file name.
- Step 6:** Select and enter commands in the script. You select commands from any one of the command tabs on the Commands Window. Review the command descriptions in section 6.0, beginning at Para. 6.3.4, Load Tab. Each tab contains a set of related commands and the options assigned to each command. Once you have selected the right combination of commands and options, enter the command into the script by clicking the **Apply** button at the bottom of the command tab. Each tab has an **Apply** button to enter that command into the script.
- Step 7:** Execute the script. Once the desired list of commands has been entered into the Script Window, click the  (Run) icon on the toolbar to begin executing commands. You can also choose to step through the commands using the  (Step) icon on the toolbar instead of running through them all. You can also use the Execution menu to perform the same functions. Use the  (Stop) icon on the toolbar to stop executing commands.

## 6.2 The Configuration Dialog


The configuration dialog allows you to set the delay between the end of one command and the beginning of another. You can also set the delay between the end of one execution and the beginning of the next execution of the script when performing more than one iteration. The Loop region of the dialog box allows you to choose between executing a set number of iterations on the script and executing the script continuously.



**Figure 29. Configuration Dialog**

## 6.3 Stepping Through the Script





Instead of running or executing commands one after another without interruption, you can choose to step through the script. Stepping through the script consists of executing each command individually and waiting for your interaction before executing the next command. To execute a script in Step mode:



- Step 1:** Select Step from the Execution menu, press F6 on the computer keyboard, or click on the  (Step) icon on the toolbar. The application will execute the first command in the script.

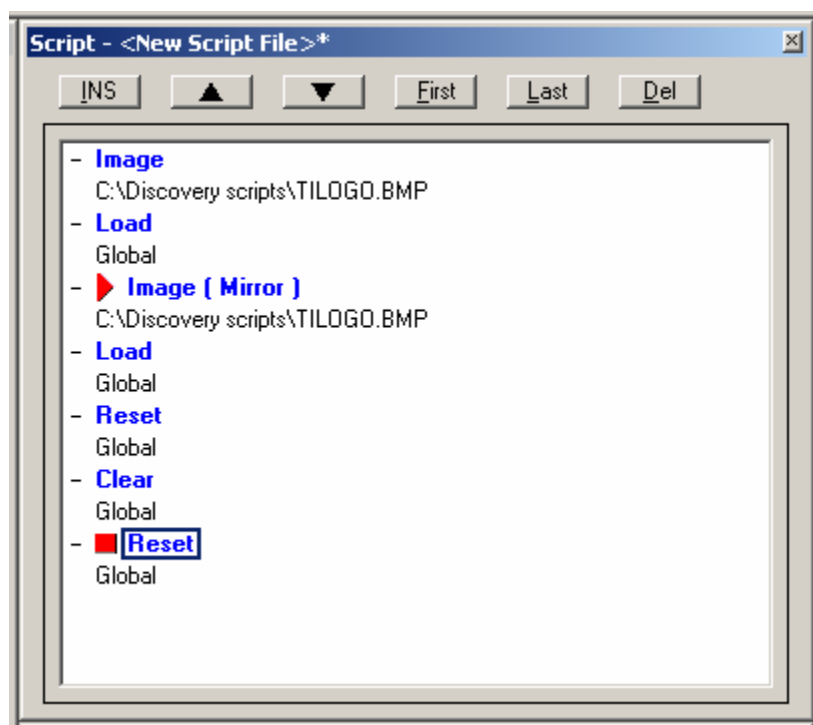
- Step 2:** Select Next Step from the Execution menu or press F8 on the PC keyboard to execute the next command.
- Step 3:** Continue stepping through the commands one at a time using the procedure in Step 2 until all commands have been executed.
- Step 4:** After you have executed the last command, you will get an "End of Execution" message in the Status window. Unless you have set multiple iterations in the configuration dialog, you will need to click on the first script command to run through the list again.

#### 6.4 Set Start and Set End

You can set the starting point and the end point of execution in any location within the script. To set the start and end:

- Step 1:** Click on the line where the execution will start and select Set Start from the Execution menu, press F3 on the computer keyboard, or click the  (Set Start) icon on the toolbar. This will place an arrow (  ) at the starting point in the script.
- Step 2:** Click on the line where the execution will end and select Set End from the Execution menu, press F4 on the computer keyboard, or click the  (Set End) icon on the toolbar. This will place a block (  ) at the end point of the script.

Clicking the  (Set Start) and  (Set End) icons after start and end have been set will cause the start and end of execution to reset to their original positions.





**Figure 30. Set Start and Set End Execution**

## 7.0 Saving Scripts and Statuses



### 7.1 Saving a Script

To save a script, proceed as follows:

- Step 1:** Select the Script option from the  (Save) or  (Save As) icon drop down menu on the tool bar, or select Save Script or Save Script As from the Save option menu on the File menu.
- Step 2:** Once the selection is made, a common dialog box will appear. Select or type the name of the file you would like to save the script.
- Step 3:** Click the **OK** button and a message box will appear to notify you that the script has been saved.

### 7.2 Saving a Status

To save the contents of the Status Window, proceed as follows:


- Step 1:** Select the Status option from the  (Save) or  (Save As) icon drop down menu on the tool bar, or select Save Status or Save Status As from the Save option menu on the File menu.
- Step 2:** Once the selection is made a common dialog box will appear, select or type the name of the file you would like to save the status.
- Step 3:** Click the **OK** button and a message box will appear to notify you that the script has been saved.

## 8.0 Printing Scripts and Statuses


### 8.1 Printing a Script

To print a script, select the Script option from the  (Print) icon drop down menu on the toolbar, or select the Script item from the Print option menu on the File menu.

### 8.2 Printing a Status

To print the contents of the Status Window, select the Status option from the  (Print) icon drop down menu on the toolbar, or select the Status item from the Print option menu on the File menu.

## 9.0 Opening Scripts and Statuses


You can insert scripts that have been previously saved in a text file by clicking the  (Open File) icon on the toolbar or by selecting the Script option from the Open File menu on the File menu. Once you've selected the desired file, the file will be opened and the script will be displayed on the Script window.



## **10.0 Creating New Scripts and Statuses**


You can insert scripts that have been previously saved in a text file by clicking the New File icon on the toolbar or by selecting the Script option from the New File menu item on the File menu.

### **10.1 Creating a New Script**

To open a new script, select the Script option from the  (New) toolbar icon drop down menu, or select the Script item from the New option menu on the File Menu.

Once the selection is made, you will be prompted to save the current script and the blank script will be inserted.

### **10.2 Creating a New Status**

To open a new status, select the Status option from the  (New) toolbar icon drop down menu, or select the Status item from the New option menu on the File Menu.

Once the selection is made, you will be prompted to save the current status and the blank script will be inserted.