

APPLICATION NOTE

AN-G01

December 21, 2005

LYNX GigE CAMERA SERIES

MIL DRIVER

Abstract: This application note describes the usage of LYNX GigE MIL Driver

1 INTRODUCTION

1.1 The Scope of this Application Note

This application note describes how to use Imperx LYNX GigE MIL driver. The goal of the manual is to give users the information they need to build application using the Matrox's MIL library with Imperx cameras.

1.2 Requirements

The LYNX GigE MIL Driver allows users to access and control any of Imperx LYNX GigE cameras in conjunction with Matrox MIL.

In summary, this application note complements, and should be used in conjunction with, three other documents:

- LYNX GigE Hardware User's Manual.
- LYNX GigE Software User's Manual.
- LYNX GigE Driver Manual.

This document assumes the following:

- That users are familiar with C, C++, and the Microsoft Visual Studio environment;
- That the users are familiar with Matrox MIL (or MIL-lite); and
- That LYNX GigE software and the C++ SDK have been installed.

1.3 Installation

The MIL driver is installed as part of the *Third-Party Software Support* option of the main installation package.

2 ARCHITECTURE

The LYNX GigE MIL Driver is designed to simply replace the *Mdig* set of functions of the original MIL library. The *Mdig* functions are used to control and acquire images from a Matrox frame grabber into a MIL buffer (*Mbuf* functions).

When acquiring frames, the MIL driver will simply put new frames into the provided MIL buffer, using the *MbufPut* functions.

Note: Because the MIL buffers do not provide direct access to its internal buffers, there may be a slight performance loss due to the copy of memory from the LYNX GigE buffer to the MIL buffer as compared with other third party packages and custom developments.

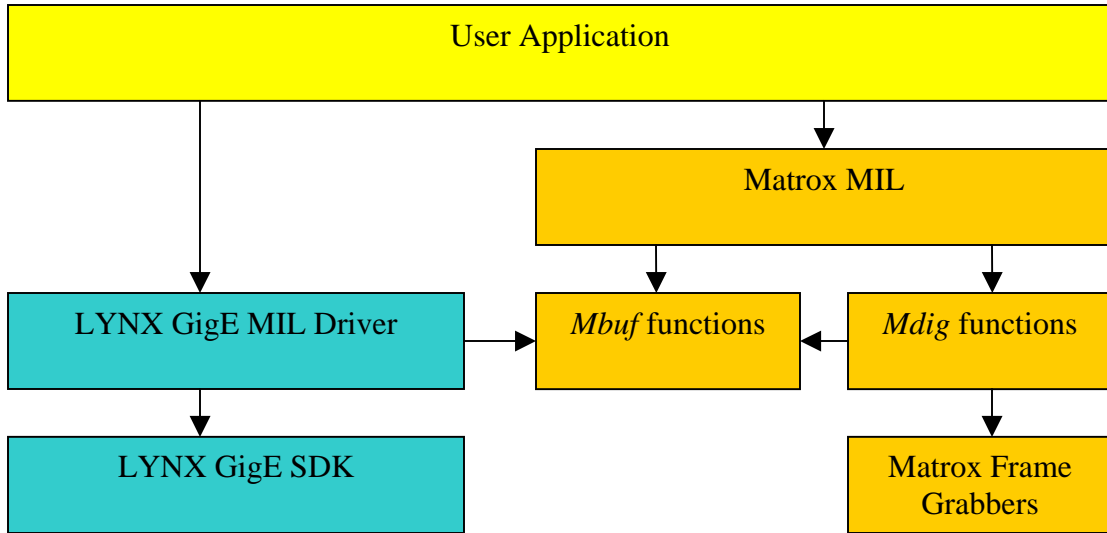


Figure 1 LYNX GigE MIL Driver Architecture

3 USAGE

3.1 How the LYNX GigE MIL Driver works

The LYNX GigE MIL Driver implements the same functions as the Mdig functions of the Matrox MIL library. These functions are:

- MdigAlloc => CyMdigAlloc
- MdigFree => CyMdigFree
- MdigHookFunction => CyMdigHookFunction
- MdigChannel => CyMdigChannel
- MdigReference => CyMdigReference
- MdigHalt => CyMdigHalt
- MdigInquire => CyMdigInquire
- MdigControl => CyMdigControl
- MdigGrabWait => CyMdigGrabWait
- MdigGrab => CyMdigGrab
- MdigGrabContinuous => CyMdigGrabContinuous
- MdigLut => CyMdigLut (NOT SUPPORTED)
- MdigAverage => CyMdigAverage (NOT SUPPORTED)
- MdigFocus => CyMdigFocus (NOT SUPPORTED)

Each of the functions responds in a manner that is very close to the Matrox MIL implementation. The only notable difference is the CyMdigAlloc function.



When allocating a new digitizer, `CyMdigAlloc` will first search for a file named `Config.xml` in the current directory of the application. Each entry in the configuration file will determine a single digitizer index that can be opened with the `CyMdigAlloc` function.

Refer to the document entitled “LYNX GigE Software User’s Manual” for more details about creating XML configuration file using the LYNX GigE Camera Interface Application.

3.2 Replacing the MIL digitizer functions

To replace the Matrox digitizer functions with the LYNX GigE MIL Driver functions, the application programmer simply needs to replace the inclusion of the MIL header with the LYNX GigE MIL Driver header.

```
//#include <mil.h>  
#include <CyMILDriver.h>
```

The LYNX GigE MIL Driver header file will do the following for your application:

- Include the `<mil.h>` header file;
- Replace all the `MdigXXX` functions for `CyMdigXXX` functions using the C preprocessor (`#defines`); and
- Automatically include the LYNX GigE MIL Driver library file.

The only thing to be careful of is NOT to use the `MsysInquire` function with the `M_DIGITIZER_NUM` parameter to inquire the number of digitizers. It is part of the MIL library and will only return the number of Matrox Frame Grabbers on the system.

3.3 LYNX GigE Specific Functions

Because the LYNX GigE MIL Driver is actually external to the MIL driver, an application may need to control some things outside of the MIL world.

3.3.1 CyMdigGetCamera

Prototype	<code>CyCameraInterface* CyMdigGetCamera(MIL_ID DigId);</code>
Description	Returns the LYNX GigE SDK camera object associated with the digitizer ID.
Parameters	<code>DigID</code> The MIL identifier representing an opened digitizer that was created using <code>CyMdigAlloc</code> .
Return	A pointer to the camera object associated with the digitizer ID. NULL if the digitizer ID is invalid.
Header File	<code>CyMILDriver.h</code>

Library File CyMILDriver.lib, CyMILDriver_Dbg.lib

Notes Do not destroy the returned pointer!

See Also

3.3.2 CyMdigSetBufferCount

Prototype void CyMdigSetBufferCount(unsigned long BufferCount);

Description Changes the number of buffers that are used internally for image acquisition.

An acquisition thread uses a pool of free buffers, which starts with the specified number of buffer. Each time a buffer is used to receive new data and the data has arrived completely, the buffer is given to a processing thread.

The processing thread then puts the data in a MIL buffer, using *MbufPut*. When the put is completed, the buffer is returned to the acquisition thread.

When the free buffer pool is empty, the acquisition thread will not queue any more buffers until the processing thread returns a buffer. It is therefore important, especially at high frame rates, to have a valid buffer count.

The default value is 32.

Parameters BufferCount The number of internal buffers to use. Minimum value is 4.

Return

Header File CyMILDriver.h

Library File CyMILDriver.lib, CyMILDriver_Dbg.lib

Notes The value at the when *CyMdigAlloc* is invoked is fixed for the newly allocated digitizer. If the number is changed, it will not affect existing digitizers.

See Also [CyMdigGetBufferCount](#)

3.3.3 CyMdigGetBufferCount

Prototype	<code>unsigned long CyMdigGetBufferCount();</code>
Description	Returns the number of buffers that will be used for the next calls to <i>CyMdigAlloc</i> .
Parameters	
Return	An unsigned integer containing the number of buffers to use.
Header File	<code>CyMILDriver.h</code>
Library File	<code>CyMILDriver.lib, CyMILDriver_Dbg.lib</code>
Notes	
See Also	CyMdigSetBufferCount

4 TECHNICAL SUPPORT

Additional technical support can be obtained by contacting Applications Support at Imperx Inc. at +1 (561) 989-0006 or via email at support@imperx.com.

Imperx, Inc
Tel: (+1) 561-989-0006
Fax: (+1) 561-989-0045
Email: techsupport@imperx.com
Web: www.imperx.com

Copyright © 2005, Imperx, Inc. All rights reserved.
Any unauthorized use, duplication or distribution of this document or any part thereof, without the prior written consent of Imperx Corporation is strictly prohibited