JULY 18, 2019

FLI KELPER SCMOS CAMERA

MATLAB DRIVERS

PETER OLEYNIKOV

Conventions

The following conventions are used in this manual:

	This icon denotes a note that contains an important information.
Bold	**Bold** text denotes the User Interface items in the software (clickable, for example, menu items, dialog box buttons etc.).
Italic	*Italic* text denotes emphasis, cross-references, or an introduction to a key concept. Italic text can also denote a text that that the user must enter.

Abbreviations

sCMOS	scientific CMOS camera
GenTL	GenICam driver

Contents

Chapter 1 Introduction 1-1

About the FLI Kepler sCMOS MATLAB driver 1-1

Windows Environment 1-1

Chapter 2 FLI MATLAB Driver Installation 2-1

Installation 2-1

Installation of the FLI Kepler sCMOS GenTL driver 2-1

MATLAB Hardware Support Packages 2-2

Configuring Image Acquisition Toolbox 2-5

Acquisition Parameters 2-7

Chapter 3 MATLAB programming 3-1

Single frame acquisition 3-1

Video preview 3-2

Chapter 1 Introduction

The current document describes a MATLAB GenICam driver installation.

About the FLI Kepler sCMOS MATLAB driver

The FLI MATLAB GenTL (GenICam) driver gives you the ability to use the FLI Kepler cameras to control camera from the MATLAB environment.

Note The cameras may operate at frame rates, depending on camera settings. Please refer to the FLI web-site for the details.

Windows Environment

The MATLAB GanICam driver requires the following software to be installed on the target PC:

- MathWorks MATLAB;
- MathWorks MATLAB Image Acquisition Toolbox;
- Visual C++ 14.1/2017 libraries for FLI library.

The FLI Driver require Visual C++ 14.1/2017 libraries. The installer will provide these if they are not already present on your PC.

Chapter 2 FLI MATLAB Driver Installation

This chapter contains the installation notes of the FLI Kepler sCMOS camera MATLAB GenICam driver. MATLAB R2014 and above runs only on Windows 64-bit.

Installation

The setup package will install the following components:

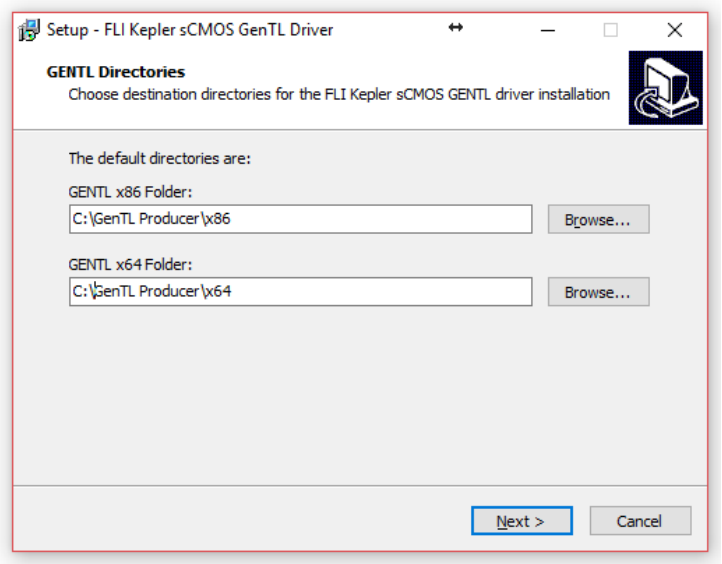
- The FLI DLLs **libflipro.x86.dll** and **libflipro.x64.dll** (on Windows 64-bit);
- The FLI GenICam driver DLLs **FLIPro.gentl.x86.cti** and **FLIPro.gentl.x64.cti** (on Windows 64-bit).

The setup will also install the Visual C++ 14.1/2017 libraries in automatic mode depending on the selected installation(s).

Installation of the FLI Kepler sCMOS GenTL driver

GenTL driver model supports 2 types of drivers on 64-bit Microsoft Windows: 32- and 64-bits. They should be copied to separate folders. However, MATLAB R2017b always runs in x64 mode. The setup will copy both 32- and 64-bit GenTL drivers. However, in MATLAB only the 64-bit GenTL FLI sCMOS driver will be used.

If the Windows Environment Variables *are not present*, then the 32- and 64-bit GenTL paths (can be modified) will be added to the Windows Environment Variables by the FLI GenTL driver installer:



The setup will

- Check the list of Windows Environment Variables for presence of **GENICAM_GENTL32_PATH** and **GENICAM_GENTL64_PATH** variables;
- If variables *are not found*, then the setup will add a new **GENICAM_GENTL32_PATH** and **GENICAM_GENTL64_PATH** to the Windows Environment Variables list. The setup will set the variables to the values, specified by the user in the **GENTL Directories** (see the screenshot above);
- If the variable *is found*, then the setup will use its value and install the FLI GetTL drivers to that path.

The following 2 files will be copied to the GenTL 32-bits path:

- FLIPro.gentl.x86.cti
- libflipro.x86.dll

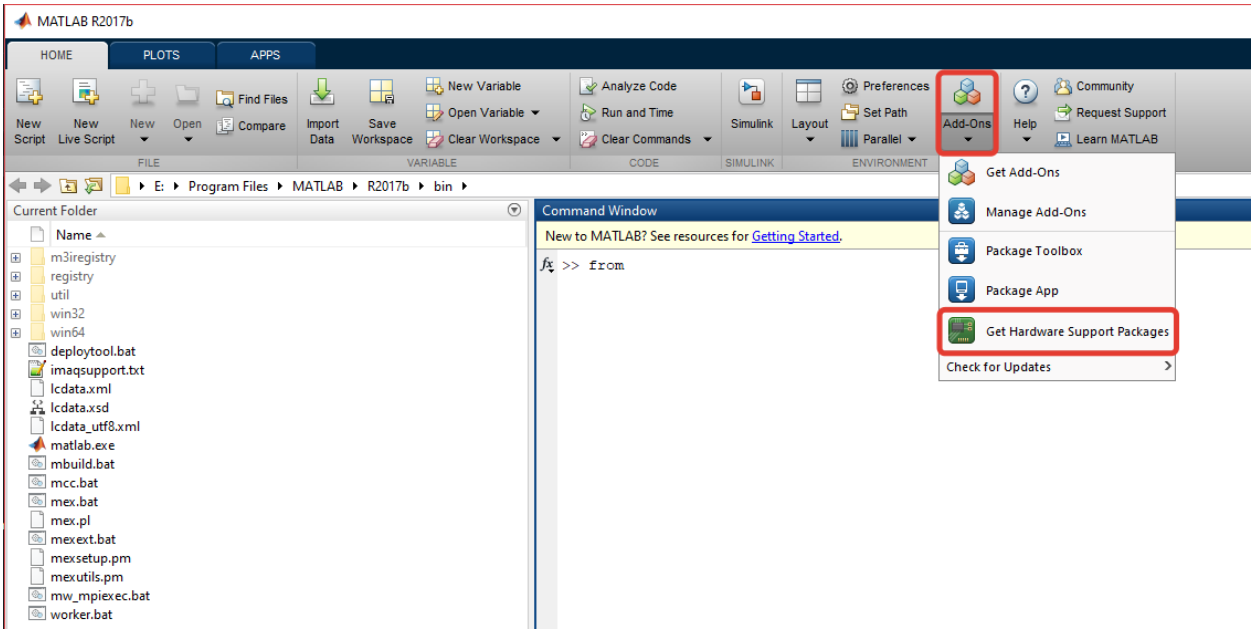
The following 2 files will be copied to the GenTL 64-bits path:

- FLIPro.gentl.x64.cti
- libflipro.x64.dll

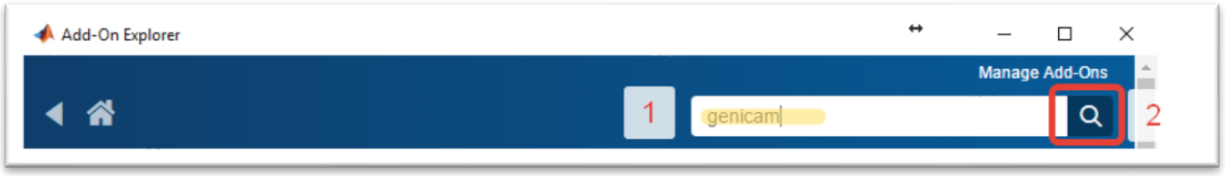
MATLAB Hardware Support Packages

The MATLAB setup requires the installation of the GenICam driver hardware support.

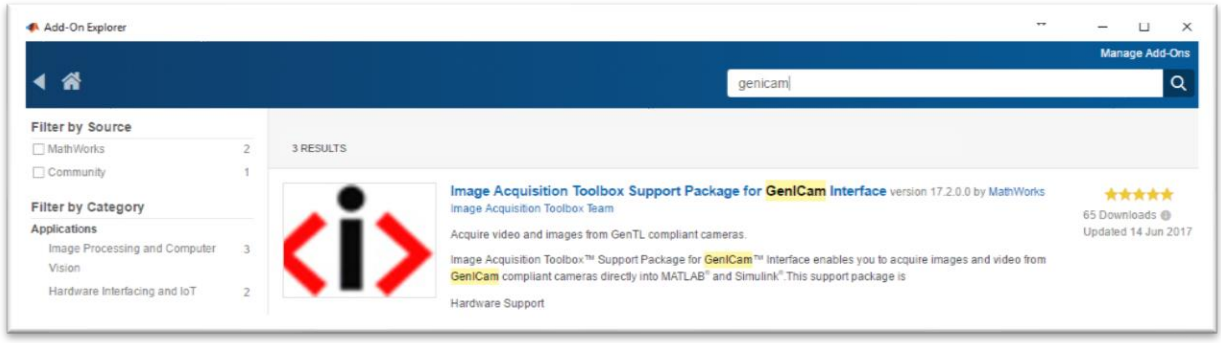
Select **Get Hardware Support Packages** from the **Add-Ons** drop-down list:



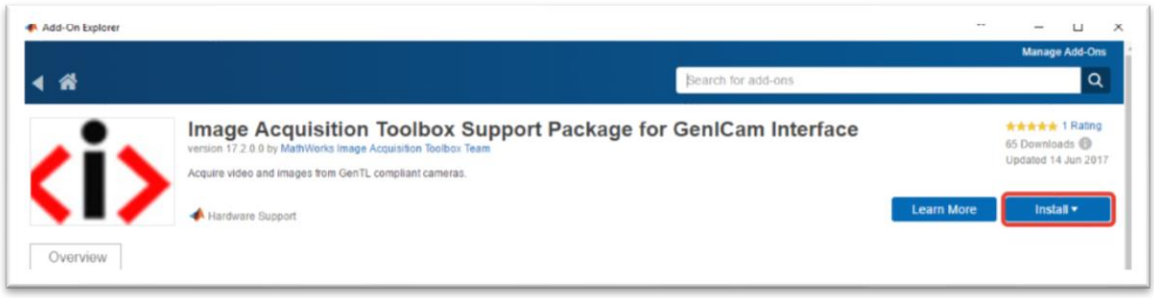
This will bring a list of Hardware support packages. Type "**genicam**" in the search box and press the search button:



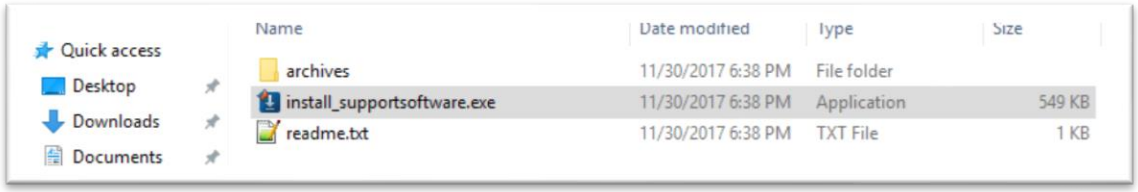
This will show the available list of packages related to the **genicam** search keyword:



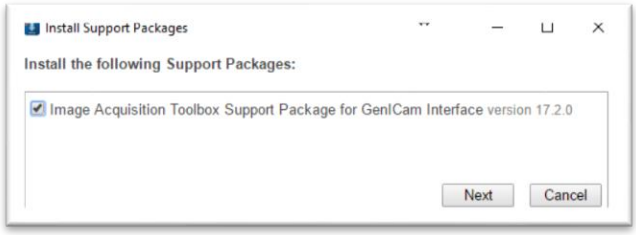
If the package has not been installed yet, select **Install** button:

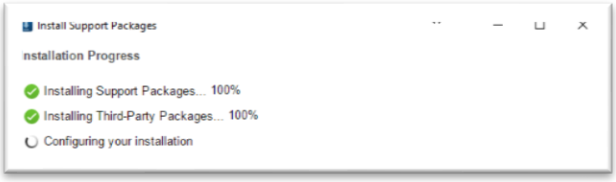


After downloading the Windows explorer will open the Downloads folder with the installation file:

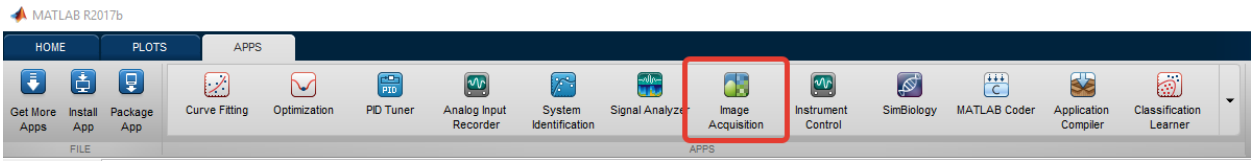


After that MATLAB will offer the installation:





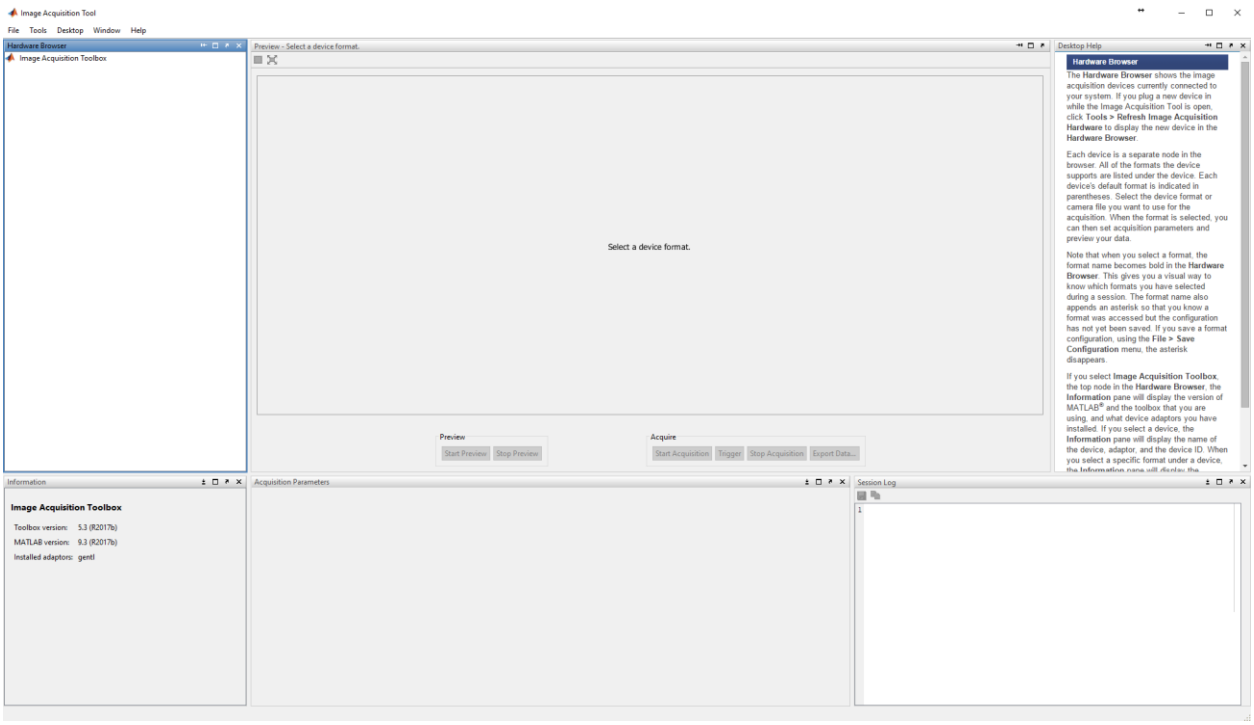
After the successful installation of the GeniCam support, start the **Image Acquisition** toolbox from the **APPS** Tab:



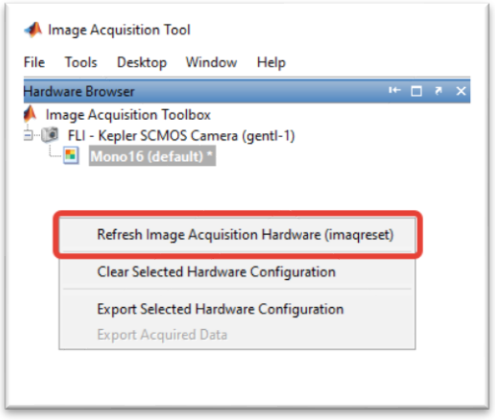
Configuring Image Acquisition Toolbox

The Image Acquisition Toolbox will be empty at the start (unless other components were previously added to this toolbox).

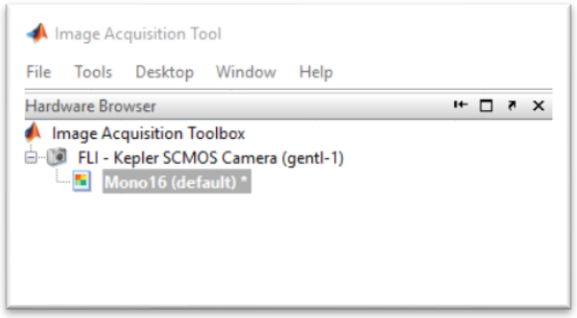
If the installation of the GenICam hardware support from previous section was successful, then the driver name **gentl** must appear in the **Installed adaptor** (bottom left).



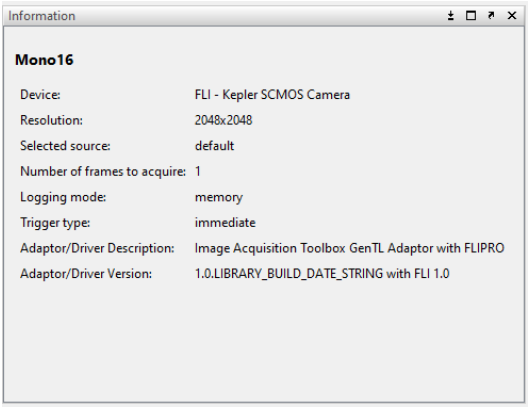
Refresh the Image Acquisition Hardware from the menu (Tools) or from the context menu (mouse right click on the **Hardware Browser**):



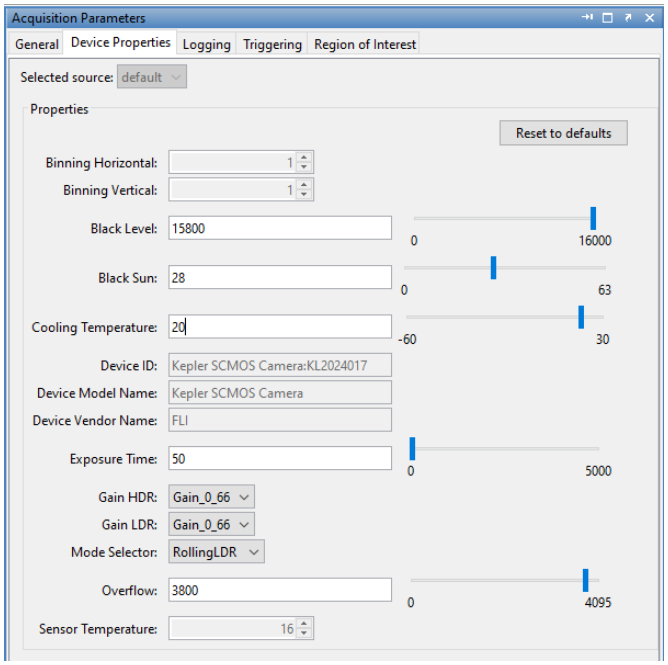
If the GenICam DLL (**FLIPro.gentl.x64.cti**) loads correctly from the GENTL64 path the Image Acquisition Tool should look like this:



The information panel will show some basic information regarding the camera (device type and the size):



All important acquisition parameters are listed in the **Acquisition Parameters** panel, under **Device Properties** tab:



Acquisition Parameters

The following table explains the Acquisition parameters, that are accessible in MATLAB image acquisition toolbox.

Property	Short Description	Value limits	Default	Read/Write
Binning Horizontal	Horizontal camera binning	Kepler supports only binning 1	1	R
Binning Horizontal	Horizontal camera binning	Kepler supports only binning 1	1	R
BlackLevel	Black Level Adjust	0-16000	15800	R/W
BlackSun	Black Level Sun	0-63	28	R/W
Cooling Temperature	The cooling temperature of the camera	–60 - +30 °C	20	R/W
Exposure Time	Camera exposure time (msec)	1 msec – 5000 msec	50 msec	R/W
Mode Selector	LDR/HDR camera modes	Rolling LDR, Rolling HDR	Rolling LDR	W
Overflow	The camera pixel oversaturation level	0-4095 (12-bits)	3800	R/W
Sensor Temperature	The current sensor cooling temperature	–	–	R

Chapter 3 MATLAB programming

This chapter contains examples that shows how to control the FLI Kepler sCMOS camera from MATLAB.

Note The Image Acquisition Toolbox™ Preview window was designed to ***only show 8-bit data***, but many cameras return 10-, 12-, 14-, or 16-bit data. The Preview window display supports these higher bit-depth cameras. However, larger bit data is scaled to 8-bit for displaying previewed data. If you need the full resolution of the data, use the **getsnapshot** or **getdata** functions.

Another way to force the preview window using the full pixel data range is to call the following command before starting the preview

```
imaqmex('feature', '-previewFullBitDepth', true);
```

Single frame acquisition

The following MATLAB code will:

- set the cooling temperature to -10 °C;
- set the exposure time to 500 ms;
- set camera mode to Rolling HDR;
- acquire an HDR frame;
- show the frame;
- save the frame to the disk as a TIFF file.

```
vid = videoinput('gentl', 1, 'Mono16');
src = getselectedsource(vid);
vid.FramesPerTrigger = 1;
src.CoolingTemperature = -10;
src.ExposureTime = 500;
src.ModeSelector = 'RollingHDR';

start(vid);
stoppreview(vid);
data = getdata(vid);
imwrite(data, 'C:\TEMP\test.tif');
imshow(data);
```

Alternatively, the following functions can be used to get a single frame and display it:

```
vid = videoinput('gentl', 1, 'Mono16');
src = getselectedsource(vid);
vid.FramesPerTrigger = 1;
src.CoolingTemperature = -10;
src.ExposureTime = 500;
src.ModeSelector = 'RollingHDR';

frame = getsnapshot(vid);
imwrite(frame, 'C:\TEMP\test.tif');
imshow(frame);
```

Video preview

The following MATLAB code will start the video preview and set the data limits mode to 'auto' for the best preview:

```
% Set preview data to native camera bit depth (default is 8 bit)
imaqmex('feature', '-previewFullBitDepth', true);

vid = videoinput('gentl', 1, 'Mono16');

% Create a preview window and get image and axes handles
h = preview(vid);
a = ancestor(h, 'axes');
% Specify scaled grayscale data mapping to colormap
h.CDataMapping = 'scaled';
% Or use the following
% set(h, 'CDataMapping', 'scaled');

% Specify auto detection of CData limits
a.CLimMode = 'auto';

% Or modify the numbers to reflect the actual limits of the data returned by the camera.
% signalRange = [10000 20000];
% a.CLim = signalRange;
```

Stopping the preview command:

```
stoppreview(vid);
```