EasyAccess code for our line scan cameras via Python

In this application example, we show how easy it is to use our e9u-LSMD series line scan cameras with Python under Windows. To do this, we use some functions for asynchronous camera operation from the dynamic camera program library (DLL). The corresponding DLL files for your operating system can be found on our service page at https://service.eureca.de/LSMD/standard/Windows/index.php. At https://service.eureca.de/LSMD/ you will also find information on operating our cameras under other operating systems or languages.

The functions shown here for asynchronous control are already powerful enough to use the camera in a wide range of applications. In fact, most of our application notes for our line scan cameras have been created using these functions.

These functions are therefore an ideal starting point for familiarizing yourself with how the cameras work. Once you have tested them and are familiar with camera control, you can also use the other routines from the DLL for applications that require more complex control of the camera or more sophisticated parameters.

1 Preparations

1.1 Install Python

Check your Python installation to ensure it is up to date. For a necessary update or a complete reinstallation see https://www.python.org/downloads/.

You should allow the installation wizard to add Python to your computer's PATH so that you can conveniently run Python programs from the command line from anywhere. You may need to restart your PC after installation for this change to take effect.

1.2 If needed: install `ctypes`

The Python package `ctypes` is required to use the external camera DLLs to exchange data with the camera.

If this package is not already included in your Python installation, you will receive an appropriate error message when you run the *EasyAccess* Python code. In this case, the `ctypes` package can be installed from the command line using the following command:

```
pip install ctypes
```

1.3 Installing the camera DLL file

All necessary routines are contained in an external DLL called `libe9u_LSMD_x64.dll`. To receive a free copy, please contact us at info@eureca.de or download it at https://service.eureca.de/LSMD/standard/Windows/index.php.

1.4 Get *EasyAccess* Python Code

Copy the *EasyAccess* code shown below from https://www.eureca.de/EasyAccess-en and save it to a new Python file with, for example, the name `EasyAccess.py`. Make sure that the Python code and the DLL `libe9u_LSMD_x64.dll` are in the same directory!

2 Running the *EasyAccess* code

Connect the camera to one of your USB ports using a suitable USB cable. Make sure your cable is definitely a data cable and not just a USB power cable! Run the *EasyAccess* Python code from the command line. You should now see an output containing information about the camera module as well as the sensor data.

3 *EasyAccess* Python code

```
# EasyAccess.py V1.1
#
# Python tool as example code for line scan cameras of the e9u series by
    EURECA Messtechnik GmbH
# - detects and starts an attached camera in asychrone mode
# - reads out the camera once and prints the recorded sensor data
#
# For details please refer to: www.eureca.de

# import library for handling external camera DLL
import ctypes

# basic values of the used linear sensor; e.g. for e9u-LSMD-TCD1304-STD:
    3648 pixel
SENSOR_PIXELS = 3648
EXPOSURE_TIME = 10000  # given in micro seconds
FRAME_TIME = 10000     # given in micro seconds

# open external DLL for 64 bit systems
libe9u = ctypes.WinDLL('./libe9u_LSMD_x64.dll')

# define argument and return types for the used functions
libe9u.e9u_LSMD_search_for_camera.argtype = ctypes.c_uint
libe9u.e9u_LSMD_search_for_camera.restype = ctypes.c_int

libe9u.e9u_LSMD_start_camera_async.argtype = ctypes.c_uint
libe9u.e9u_LSMD_start_camera_async.restype = ctypes.c_int

libe9u.e9u_LSMD_set_times_us.argtypes = (ctypes.c_uint, ctypes.c_uint,
    ctypes.c_uint)
libe9u.e9u_LSMD_set_times_us.restype = ctypes.c_int

libe9u.e9u_LSMD_get_next_frame.argtype = ctypes.c_uint
libe9u.e9u_LSMD_get_next_frame.restype = ctypes.c_int

libe9u.e9u_LSMD_get_pixel_pointer.argtypes = (ctypes.c_uint, ctypes.c_uint)
libe9u.e9u_LSMD_get_pixel_pointer.restype = ctypes.POINTER(ctypes.c_int16)

# search for camera
libe9u.e9u_LSMD_search_for_camera(0)

# start the found device in asychrone mode
libe9u.e9u_LSMD_start_camera_async(0)

# setting integration time
libe9u.e9u_LSMD_set_times_us (0, EXPOSURE_TIME, FRAME_TIME)
```

```
# get an image and store it for readout
libe9u.e9u_LSMD_get_next_frame (0)

# getting the pointer to the array with the sensor data
pointer = libe9u.e9u_LSMD_get_pixel_pointer(0, 0)

# printing the sensor data
for pixel in range(0, SENSOR_PIXELS):
  print(str(pixel) + ": " + str(pointer[pixel]))

exit()
```

4 Error handling

For the sake of clarity, the *EasyAccess* code above does not include any additional error handling lines. However, it is recommended to perform error handling after each call of a DLL routine to check the correct operation of the camera and/or the computer system.

For error handling, all DLL routines provide a return value that can be checked for possible errors. If the respective function ended without error, NULL is returned. If an error occurred, an error code is returned. For the exact meaning of the error codes, please refer to the detailed camera documentation.

The following is an example of suitable error handling using the `libe9u.e9u_LSMD_search_for_camera` DLL routine.

```
print("Searching for camera: ", end='')
i_status = libe9u.e9u_LSMD_search_for_camera(0)
if i_status == 0:
  print("OK!")
else:
  print("Error! Code: " + str(i_status))
  exit(1)
```


EURECA Messtechnik GmbH
Deutz-Kalker Str. 35
50679 Cologne, Germany

Tel: +49 (0) 221 / 952629 - 0
info@eureca.de
www.eureca.de