The Imaging Source Helpdesk

How to use IC Imaging Control without Forms and get triggered images in C#

S.Geißler - 2019-01-28 - Programming

A very simple IC Imaging Control application is a console program, without any Windows Forms and no user interaction. This sample shall setup a camera, add the ImageAvailable event and enable trigger. In general, if a camera is triggered, means in trigger mode, always the ImageAvailable event should be used to react on a new image.

First off all, after the project has been created, the reference to IC Imaging Control must be added. Take care, you use the correct Framework version, because IC Imaging Control is available for the old version 2 as well as the current version 4.

The code looks as follows. For simplified reading, the error handling is not implemented.

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using TIS.Imaging;


namespace Trigger_Example_for_Dynamic_IC

{

   class Program

   {

      static void Main(string[] args)

      {

         // Declare and create IC Imaging Control

         ICImagingControl IC_Control = new ICImagingControl();

         // Open a camera. This must be connected already.

         IC_Control.Device = "DFK 37UX290";

         // Set a video format. Make sure, the camere supports that format.

         IC_Control.VideoFormat = "RGB32 (640x480)";

         // Set a frame rate.

         IC_Control.DeviceFrameRate = 30.0f;
```

```
            // Enable the trigger
            IC_Control.DeviceTrigger = true;
            // LiveCaptureContinuous must be set to true for saving all incoming images in memory
            // and call the ImageAvailable event.
            IC_Control.LiveCaptureContinuous = true;

            // Add the ImageAvailable event handeler
            IC_Control.ImageAvailable += new
System.EventHandler<ICImagingControl.ImageAvailableEventArgs>(OnImageAvailable);

            // Start and stop the live video
            IC_Control.LiveStart();
            System.Threading.Thread.Sleep(1000);
            IC_Control.LiveStop();

        }

        // This is the ImageAvailable event handler
        public static void OnImageAvailable(object sender,
TIS.Imaging.ICImagingControl.ImageAvailableEventArgs e)
        {
            // The image is in e.ImageBuffer
            Console.WriteLine("Image!");
        }
    }
}
```

The complete project can be downloaded from https://github.com/TheImagingSource/IC-Imaging-Control-Samples/tree/master/c%23/Trigger%20Example%20for%20Dynamic%20IC