

VN-25MX-M/C 72

PIXEL SHIFTING CAMERA WITH COAXPRESS INTERFACE



LABVIEW VI APPLICATION EXAMPLE

Business Confidential – Page 1

Please Contact Vision Systems Technology, A Vieworks Master Distributor North America
www.visionsystemtech.com - 858.449.1562 - sales@visionsystemtech.com - [skype:visionsystemtech](https://www.skype.com)

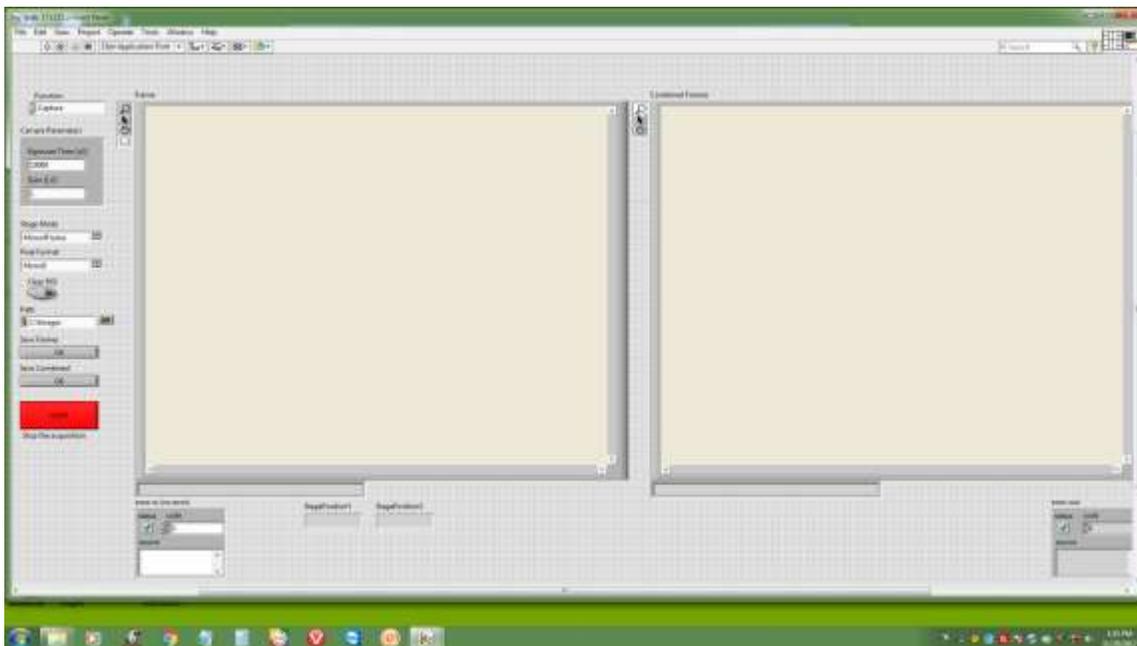
Grab (v2.1.0.2_171218) VI LabVIEW software

Scope:

The Grab (v2.1.0.2_171218), hereafter referred to as “Grab” LabVIEW software is an example program designed to demonstrate the various functions and operation of the Vieworks VN-25MX-M72AO 25MP pixel shift coaxpress mono 72fps camera connected to a Euresys Quad G3 coaxpress board in a Win 7 computer running LabVIEW Full version or higher. This software was developed using National Instruments LabVIEW 2015 and should run on LabVIEW 2015 or more recent. The software is provided “as-is”. If there are questions please contact kris.balch@visionsystech.com. If there are modifications that you feel are needed, please forward them to sales@visionsystech.com and we will discuss with our developer the changes. In order to use the larger array sizes as in Mono9Frame the 64 bit version of LabVIEW is necessary. The (v2.1.0.2_171218) reference above is to indicate that the name of the software on disk is the word Grab followed by a (v2.1.0.2_171218). For example, as of this writing the full name of the program is “Grab 171218.vi” as it was released on Dec. 18, 2017. In addition to LabVIEW 2015 or better, the National Instruments Vision Acquisition toolkit must be installed as well as the directory containing the Grab program and all of the supporting Vis and SubVIs, and assorted files. The system this was built on was 64 bit Win7 on an Intel Core i7-6700 @ 3.4 GHz with 64GB of RAM.

Functionality:

The following image is of the Front Panel or user interface to the program.

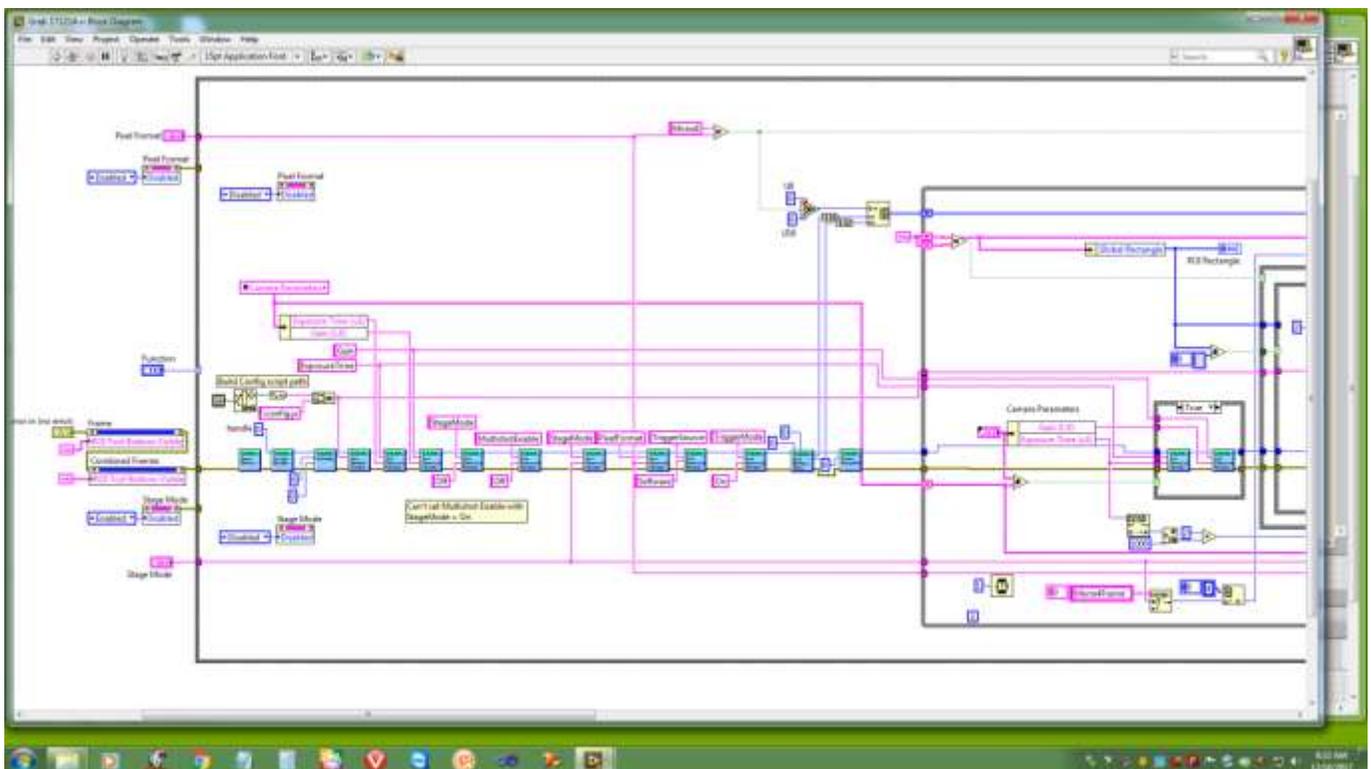


In the top left corner is the Function selector. Before running the program, select what function you want to run. The possible selections are Capture, Camera Reset, Get Image and Stage Reset. The Capture function is the most often required function as it continuously captures images and consolidates them as indicated by the Stage Mode and Pixel Format selection. The Capture function also allows for Region Of Interest (ROI) selection & Clearing, saving frames and consolidated images, setting Exposure time and Gain. There are indicators for the stage position and Error indicators. There are two image frame displays. The one on the left is for the individual frames and the one on the right displays the consolidated image. If you only want one set of frames and the consolidated image you can click on the Stop button and optionally the Save buttons before running the program. Just below the Edit menu is a right facing arrow. Clicking on this will run the program. In Capture mode the program runs until the Stop button is clicked. On the left image display there are 4 tools for zooming selecting panning and a rectangular ROI selection tool. Right clicking on the image offers a variety of other features standard to LabVIEW. The Camera Reset, Get Image and Stage Reset functions do what one would expect.

Software:

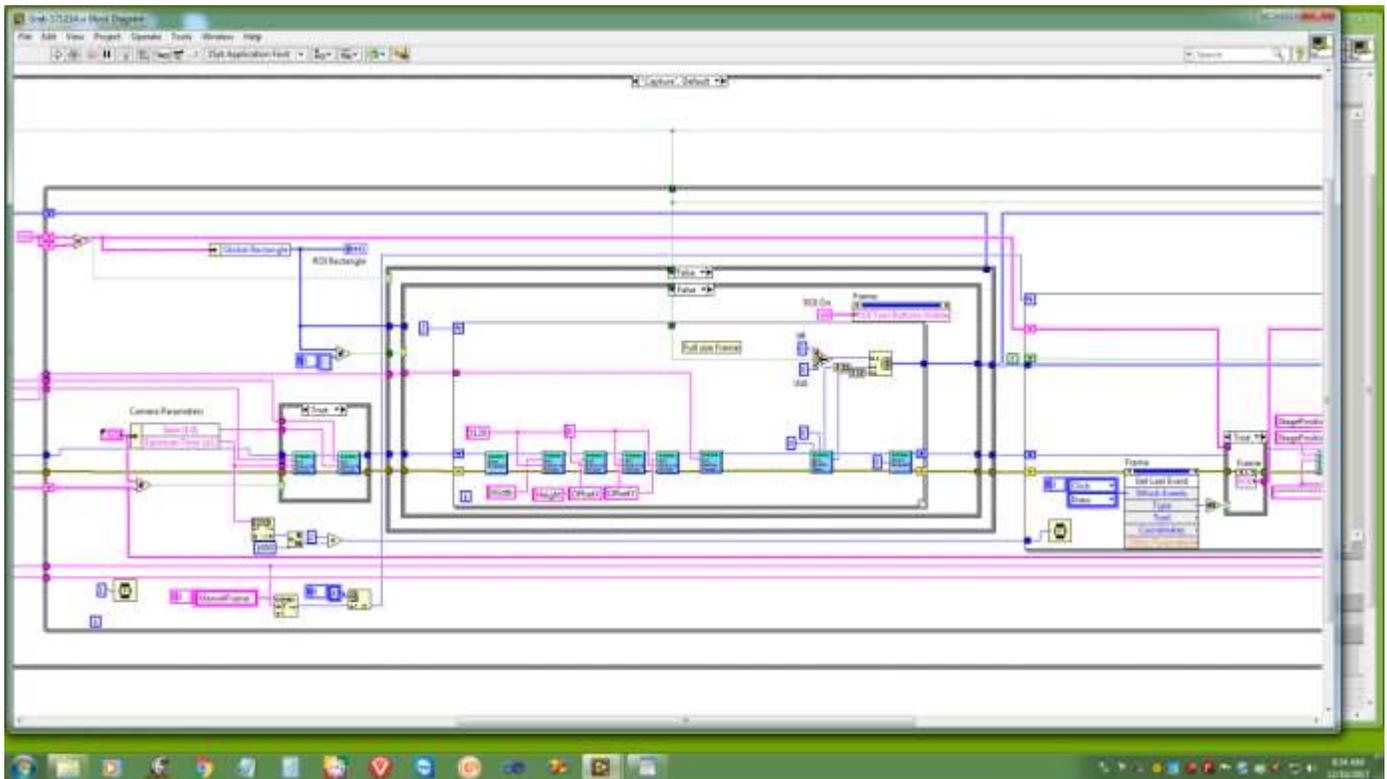
The Block Diagram or source code is shown below in 3 separate sections. They are Block Diagram 1, Block Diagram 2 and Block Diagram 3

Block Diagram 1

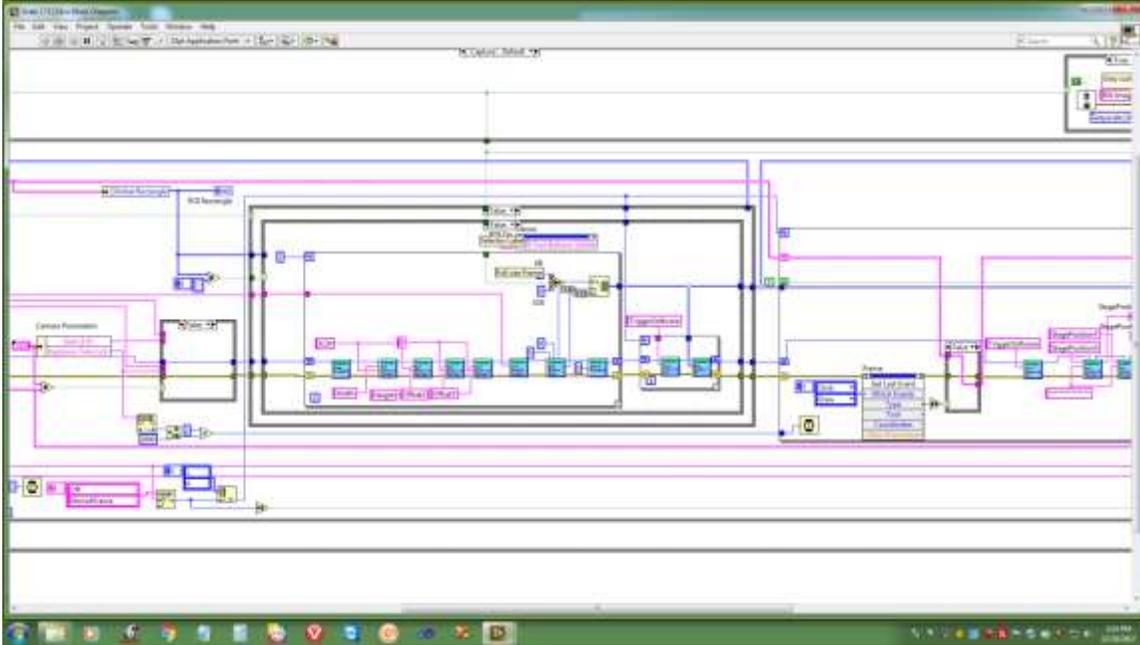


Block Diagram 1 is mostly initialization. Starting from the far left the available image tools on the image frame displays are selected and the Stage Mode and Pixel Format controls are Enabled. The Function selector selects which Case in the Case statement is executed. In the Capture Case the Stage Mode and Pixel Format controls are Disabled, the Driver is opened, a handle is created and the system is initialized. Next the path to the Configuration Script, a .js, (Jscript file), is defined and the file is sent to the VI that loads the configuration, The Gain and Exposure time are initialized and the Stage Mode is set to Off (The Stage Mode must be off before Multishot can be set. Multishot Enable is set to Off and StageMode is set to the front panel control setting. PixelFormat is set to the front panel control setting. TriggerSource is set to Software, and TriggerMode is set to On. The initial buffer size is retrieved from the camera and the Acquisition is started.

Block Diagram 2a

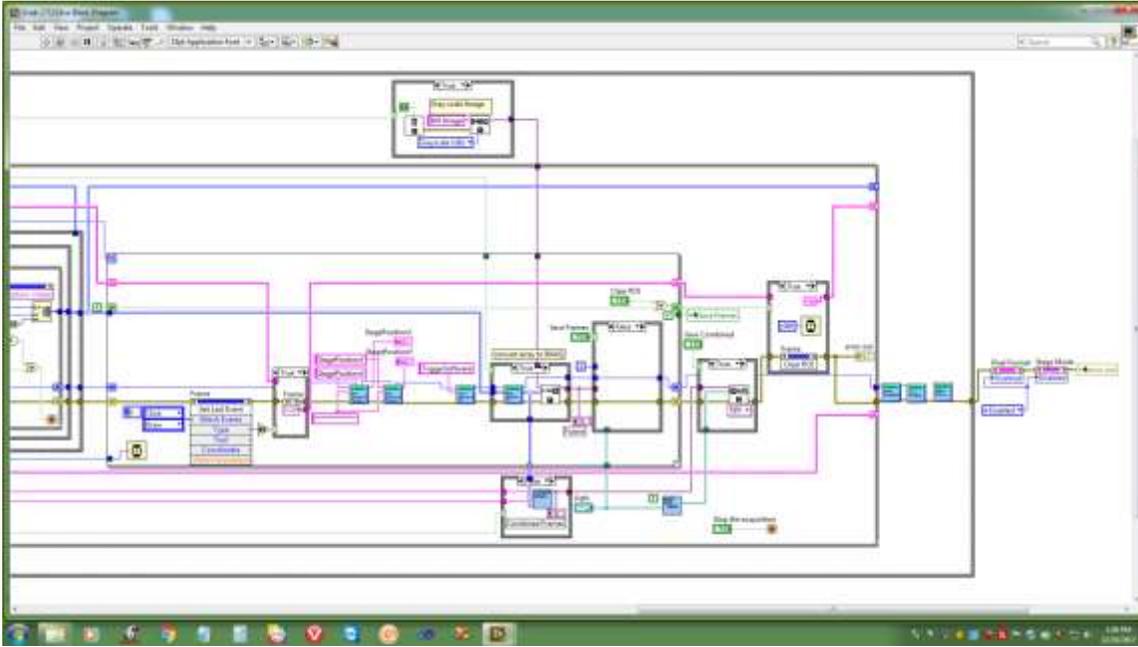


Block Diagram 2b



Block Diagram 2a and 2b show the beginning of the main execution While loop. It starts by allowing for changing the Gain and Exposure time on the fly. Depending on the front panel control setting the Stage Mode controls the frame sequence. The Exposure time sets a delay time between frame captures. The outside case statement is usually True where nothing happens however it goes false if an ROI is either selected or cleared. When an ROI is selected (Block Diagram 2b) the inner Case goes True and False when the ROI is cleared. When the ROI is selected Acquisition is Stopped, the ROI is modified to conform the constraints of multiple rows and columns and offsets the Width, Height, OffsetX and OffsetY are computed and loaded. The Configuration .js file must be reloaded at this time to have the previous 4 steps get fully loaded, the new buffer size is retrieved and the Acquisition is restarted. This operation is done more than once if for some reason the values don't take the first time. When the ROI is cleared the same operation is completed but with the full frame size instead of a computed ROI size (Block Diagram 2a).

Block Diagram 3

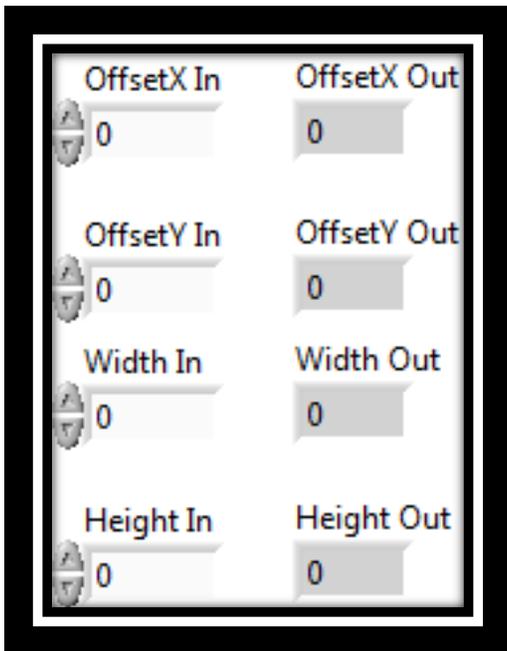


Block Diagram 3 shows the main image capture loop, followed by the cleanup and stop code. Entering the FOR loop selected to run however many times as there are frames in the compiled image. If an ROI event is detected the ROI coordinates are saved so they can be used during the next capture sequence. The X and Y Stage positions are retrieved, depending on the Pixel Format either the Mono8 or Mono 16 buffer is given to the GetBuffer function where the image is returned and sent to the Array to Image VI so it can be displayed in the image displays on the front panel and optionally saved to a directory indicated by a front panel control. When all the frames have been collected they are sent to the Make Combined Image.vi where they are prepared for display on the front panel and to optionally be saved to the directory indicated by the front panel control. Next is where the clear ROI function clears the ROI and reloads the image size array for the full size image. This WHILE loop runs continuously until the Stop control goes True. Finalization, clean up code consisting of StopAcquisition, Free Handle and Close Driver are executed followed by re-enabling the Stage Mode and Pixel Format controls.

This SubVI combines frames in a multiframe capture to form a higher resolution image. Depending on the Pixel Format either an 8 bit or a 10 (16) bit array of frames is sent to the main case statement. The case in the case structure is selected depending on the condition of the Stage Mode control. The case for the Mono4Frame Stage Mode is shown but the other cases will only be slight variations. The numbers of rows and columns is determined and arrays for the first and second rows and the full image size are initialized to either all 8 bit zeros or all 16 bit zeros. In the Mono4Frame mode each pixel in each of the frames are combined to make 4 pixels in the combined frame. The inner FOR loop combines the two top pixels into one array and the two bottom pixels into another array until the array lengths are twice the size of any of the 1 of 4 frames. The outer FOR loop then combines the 2 arrays and accumulates all of the 2 row arrays to for the combined image that is twice as high as any of the 1 of 4 frames. This combined image is converted to a LabVIEW image data type and set to the next level up VI.

The Adjust ROI HW.vi

Front Pane



This VI adjusts the selected ROI to conform with the constraints of the camera. The camera is able to generate ROIs with limitations on where the ROI begins, the offset and how many rows and columns it may contain. For instance, the camera may only have a multiple of 256 rows or columns and may only be allowed to be offset by some 2^n number of rows and columns.

Block Diagram

