

# **NeuroCheck Image Acquisition and Triggering Notes**

## **Handout for FSI Machine Vision Training Course**

### **Overview of this document and currency of it's information**

One may divide this document as follows:

The first ½ on the subject matter in general

The second ½ on parallel processing.

This paper was written when NeuroCheck 5.0 was the current version. At that time (compared to now) the terminology for the parallel processing tools was different and more confusing. Also, at that time, there was no explanatory section in the NeuroCheck software manual regarding parallel processing. NeuroCheck 5.1 included these changes (compared to NeuroCheck 5.0) to make parallel processing easier to understand and use:

1. The tool for capturing an image for parallel processing has been renamed from “Capture Image Asynchronously” to “Capture Image in Parallel”
2. The tool for initializing parallel processing is now a separate tool “Control Image Acquisition” rather than a variant of “Capture Image Asynchronously”.
3. An explanatory section on this has been added to the manual, making some of the explanations in this paper unnecessary.

So, for users of 5.1 on, the need for certain portions of this document has been eliminated, and the terminology for parallel processing tools has been eliminated. To support 5.0 users, this document has been “frozen”. 5.1-on users are advised to:

- Keep the above in mind
- Translate tool names per #1 and #2 above
- Read the parallel processing section of the NeuroCheck 5.1 manuals and help files.

### **Process Overview / Terminology**

An oversimplification of the first stages of a machine vision image is useful:

1. “Take the Picture” (Image Acquisition)
2. Make the image available to NeuroCheck
3. Transfer the image “into the NeuroCheck program”

For applications requiring taking a picture of a moving part, proper equipment selection evaluates these factors:

“Interlaced” cameras take the picture in two stages. If the item moved significantly during the imaging process, these two stages will occur at different item positions. The remedy is to specify progressive



**FSI Technologies Inc.**

668 Western Avenue • Lombard, IL 60148-2097

(630) 932-9380 • FAX: (630) 932-0016

SDOC-058-001 8/25/06  
www.fsinet.com

scan type imaging. This is standard on the CVS-500 (Compact) and a low cost (generally always chosen) option on the CVS-700 (separate-camera machine vision system)

“Taking the picture” can involve waiting for the camera to “get ready”, and also waiting for the computer/software to process the trigger signal. This can induce a random delay (example: 0 - .04 seconds) before the image is acquired. If such a variable delay will cause a problem, the remedy is to use our “direct hardware trigger”. (details below)

### **There are two ways to trigger the image acquisition:**

#### **Driven by the NeuroCheck program**

**Driven by directly triggering the imaging electronics.** This enables extremely fast and repeatable control of the image acquisition process to provide consistent image position on moving processes.

To obtain direct hardware triggering capability, select either a CVS-500 series unit (which has this standard as a user-selectable option) or specify the “direct hardware trigger” option in a CVS-700. In both cases, these options also include ability to force an immediate image acquisition rather than waiting for completion of the camera’s current cycle. Terms used to refer to this process are “Asynchronous Image Acquisition”, “Frame Reset” and “Vertical Reset”. (This is a different meaning of “asynchronous” than that of the NeuroCheck tools with that name.) Then follow the unit specific instructions from the CVS-500 / CVS-700 manual.

**Sequence for program-driven image acquisition.** The check routine is started by a (typically) one-shot triggering device such as an FSI # RET-002 and RET-003 one-shot retro-reflective photo-eye. (The “one-shot” feature is the most common way to avoid multiple program executions which can occur if the trigger signal is still high when the program is completed and ready to restart) A NeuroCheck “capture image” tool is inserted, and its execution causes an image acquisition.

**Program sequence for hardware-trigger driven image acquisition** There are many various ways to do this, depending on the system configuration and the requirements of the application. A quick overview of NeuroCheck’s tools in this area provides a useful starting point.

**NeuroCheck “Capture Image” type tools.** NeuroCheck has two types of “capture image” type tools:

**“Capture Image”** In a direct hardware trigger context, this tool causes the program to stop on the “Capture Image” tool and wait for the next image (which will be caused by a direct hardware trigger)

**“Capture Image Asynchronously”** Used only with a direct hardware trigger, this is a capability / mode which allows a new image to be acquired even if the program has not completed processing the previous image. If an image has already been acquired, the program takes the existing image rather than waiting for the next one.

### **Examples of “Capture Image” tool based programs for direct hardware trigger applications**

There are 2 common methods to do this. Here is a basic example of each. For each example, the unit is a single camera, progressive scan CVS-700 ordered with the direct hardware trigger option.



**“Pre-Trigger” method** A (typically “one-shot”) “pre-trigger” starts the check routine. It comes to the “Capture Image” tool (typically the first tool) and waits for an image to be acquired. A short time later the direct hardware trigger triggers an image acquisition; the “capture image” tool receives it and the program progresses.

This specific example is a conveyor belt production line running at 180 feet per minute inspecting parts which are approx 1 foot cubes spaced (center-to-center) 3’ apart. So, the inspection rate is 60 parts per minute. For this example, the inspection program takes .1 second to complete after it receives the image. Two retro-reflective photo-eyes are placed successively on the line. The first one is an FSI # RET-002 or RET-003 retro-reflective photo-eye with the one-shot duration set for about 50 milliseconds. (long enough to be easily seen by the software, short enough to be off by the time that the program finishes executing.) 2” downstream from it is a second standard photo-eye (one shot not required).

**“Automatic Restart” Method** The program is set to restart automatically with zero time delay. A single (non-one shot) photo-eye triggers an image acquisition through the direct hardware trigger input. The program always starts / restarts immediately, and it waits on the “Capture Image” tool until an image is received. Although this simplifies the required trigger inputs, it does create a need to deal with a few other issues. If “programming type devices (mouse/keyboard etc.) are connected, a unit that is running a program that is waiting for the external trigger (particularly if the waiting / time-out time is set to infinity) will be less responsive to the mouse, and this condition is sometimes mistaken for a malfunction/ being “locked”. The remedy is familiarization, and either giving it a trigger (to allow switch out of automatic mode) or using a windows tool to close NeuroCheck.

Also, the standard pre-defined global check routine level I/O are not usable/ used; and instead discrete I/O control is defined using “Check” or Check function level I/O control, or set/reset I/O tools.

**“Capture Image Asynchronously” tool based programs for direct hardware trigger applications.**

Again, this is a capability / mode which allows a new image to be acquired even if the program has not completed processing the previous image. If an image has already been acquired, the program takes the existing image rather than waiting for the next one. This capability and these tools are designed for use only with the direct hardware trigger option present and enabled. (otherwise the behavior of the system is confusing and not useful)

The 2 most common reasons for using this method are:

1. High-speed direct hardware triggered simultaneous image acquisition is required for 4 or more cameras.
2. Very high-speed line, too fast for sequential processing. The next image may be acquired while the program is still being executed on the previous image. As a timing example, imagine an inspection routine which requires .100 sec to execute, broken down as follows: .033 second to acquire the image (and make it available to the program), and .067 seconds for the rest of the processing. With conventional processing methods, the production rate must be kept down to 1 part every .100 seconds, or 10 parts per second for proper operation of the machine vision unit. With this special mode, proper machine vision system operation is maintained down to approx .067 sec. between parts, or about 14.9 parts per second. Machines which are required to track part positions (for reject mechanism control etc.) and



which run in this realm generally require more sophisticated methods than “zone-based” methods which are common at lower speeds. At 14.9 parts per second, the total time to inspect a part is still .100 sec, but the time between parts is approximately .067 seconds. So, by definition, when this capability is utilized, a part will have traveled farther than the part-to-part spacing during the inspection process.

### **Example on use of “Capture Image Asynchronously” tools/ capabilities due to high speed**

Beer bottles are traveling down a conveyor at a rate of 20 bottles per second. The brewer wishes to measure 3-4 dimensions and angles on the neck and cap of each bottle against a set of tolerances, and provide a daily report of the percentage of bottles that fell within those tolerances. He wishes to stick with a standard economical CVS-700 without exotic cameras, but with (economical) 1.4 GHz processing and the direct hardware trigger option.

A program is written which (not counting image acquisition) takes .073 seconds to execute including image acquisition, and .04 seconds to execute not counting image acquisition. There is only .05 seconds between bottles, and so the parallel processing capabilities of NeuroCheck are required.

The implementer chooses to have NeuroCheck automatically restart itself with no time delay.

The program uses two entries of the “Capture Image Asynchronously” tool. In the first entry, the “initialize” box is checked, but not the “always” box. This converts this step to merely an initialization, I.E. it does not capture an image. When the system is switched to automatic, the first “capture image asynchronously” tool initializes the system and then does nothing else during that session. The next entry does not have any of these of these boxes checked, and this is the actual “capture image” tool.

While the program is still executing, the next bottle arrives and triggers the image acquisition process. The image acquisition begins and occurs in parallel with the execution of the inspection program on the previous bottle. The inspection program completes execution about .01 sec. before the new image is available to it. So the program restarts itself, and waits .01 sec. on the second “capture image asynchronously” tool until the image is available for the next inspection.

### **Example of use of “Capture Image Asynchronously” tool on a “lower” speed system.**

This application inspects parts which are traveling at high speed, but the inspection rate is “only” 2 parts per second. The inspection requires 6 views/ 6 cameras, all to be taken absolutely simultaneously. The entire inspection program (including image acquisition) takes .3 seconds. So, parallel processing is not required for speed considerations. However, due to the application requirements, the unit-specific instructions furnished with the CVS-700 said that “capture image asynchronously” tools must be utilized to assure that all images are absolutely simultaneously. They also included an example of those steps of the program which the implementer entered, and then proceeded to conventional inspection programming.

### **Does use of “Capture Image Asynchronously” tools introduce “one behind processing” type complexities into machine design?**

The simple, direct answer is “no”. This question usually arises from one of the following 2 areas:

1. If a system is “erroneously” run using this tool without direct hardware triggering, “one behind” imaging may occur, raising this question. So, the key thing here is understanding that operation under “erroneous” setup is not indicative of actual operation.



2. The capability of acquiring one image while inspecting the previous image raises this question.

A useful example related to #2 is a PLC-based machine control for removal of individual defective parts (using a reject mechanism) on a production line with regularly spaced parts. The simplest control strategies rely on execution of the entire inspection process (from “taking the picture” and inspection processing through PLC recognition of the vision system output) while the part is well within a single zone, are so the simplest strategies are valid only up to a certain “speed limit”....beyond that, a more different strategy is required. As line speeds increase, that speed limit is exceeded just prior to a speed which utilizes the “#2” capability.

So, a speed-driven requirement for the “#2” capability is also a speed-driven requirement for a more complex control strategy. Both are *results* of high speed, and neither is a cause of or an exact indicator of the other.



*FSI Technologies Inc.*

668 Western Avenue • Lombard, IL 60148-2097

(630) 932-9380 • FAX: (630) 932-0016

www.fsinet.com

SDOC-058-001 8/25/06