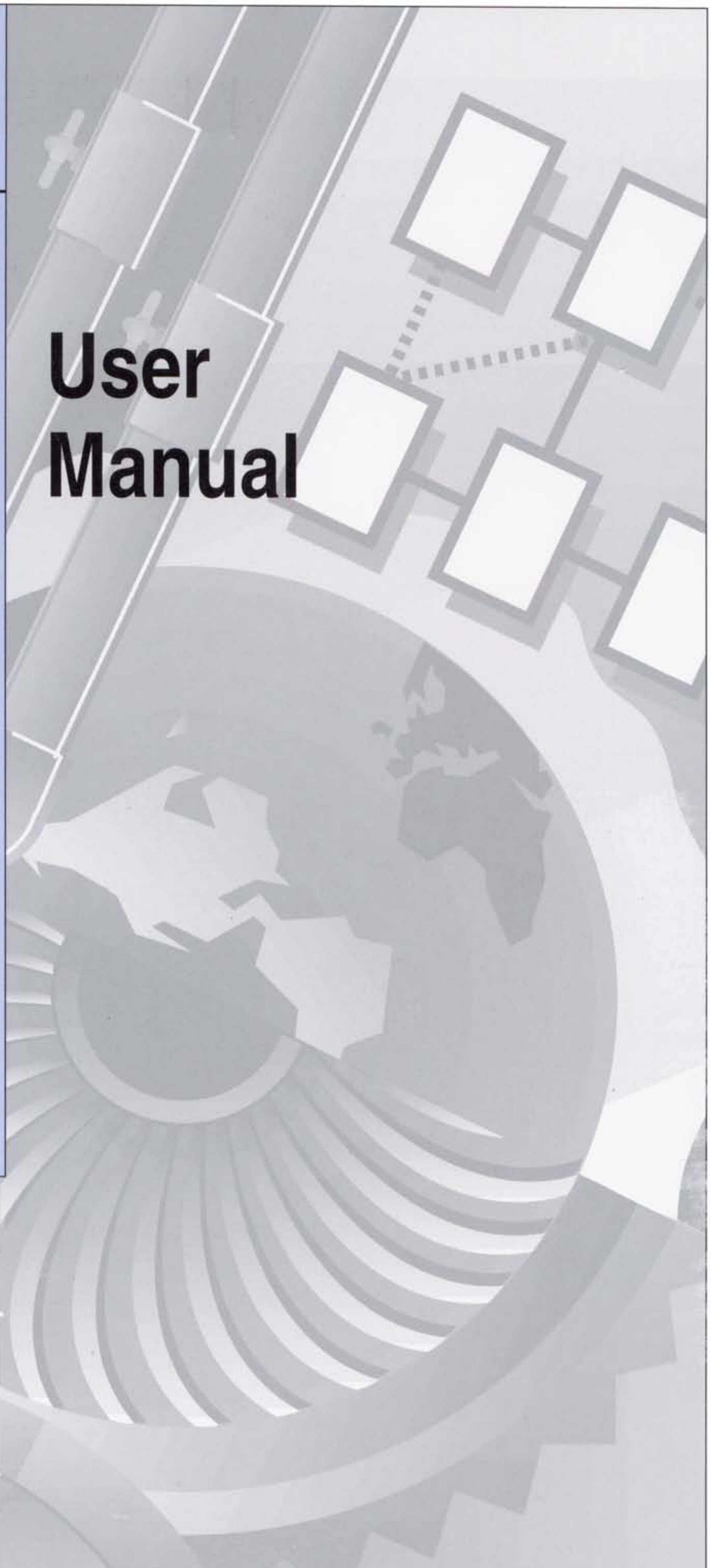
Allen-Bradley

Bulletin 2803 VIM™ Vision Input Module

(Cat. No. 2803–VIM2)

User Manual



Important User Information

Because of the variety of uses for the products described in this publication, those responsible for the application and use of this control equipment must satisfy themselves that all necessary steps have been taken to assure that each application and use meets all performance and safety requirements, including any applicable laws, regulations, codes and standards.

The illustrations, charts, sample programs and layout examples shown in this guide are intended solely for purposes of example. Since there are many variables and requirements associated with any particular installation, Allen-Bradley does not assume responsibility or liability (to include intellectual property liability) for actual use based upon the examples shown in this publication.

Allen-Bradley publication SGI-1.1, *Safety Guidelines for the Application, Installation, and Maintenance of Solid-State Control* (available from your local Allen-Bradley office), describes some important differences between solid-state equipment and electromechanical devices that should be taken into consideration when applying products such as those described in this publication.

Throughout this manual we use notes to make you aware of safety considerations:

ATTENTION: Identifies information about practices or circumstances that can lead to personal injury or death, property damage or economic loss.

Attention statements help you to:

- identify a hazard
- avoid the hazard
- recognize the consequences

Important: Identifies information that is critical for successful application and understanding of the product.

Table of Contents

Chapter	Title	Page

Figure/Table	Title	Page

List of Figures

Figure/Table	Title	Page

List of Tables

Chapter 1 Using This Manual

Chapter Objectives

This chapter provides an overview of the contents of this manual, and also contains a definition of the intended audience, and information on related publications.

What this Manual Contains

This manual provides the information and procedures you need to prepare the Allen-Bradley Vision Input Module™, Cat. No. 2803-VIM2, for an inspection application. It includes information on the installation of the VIM2 module* and use of the VIM2 menus and icons, general guidelines for staging a vision inspection, machine vision theory, and procedures and information for interfacing with a PLC or other host device.

***Note:** The Cat. No. 2803-VIM2 Vision Input Module is referred to in this manual as the "VIM2 module."

Table 1.A, provides a brief overview of the manual contents.

Table 1.A
Manual Contents

Chapter	Title	Summary
1	Using This Manual	Includes chapter overviews, audience definition, and related publications.
2	Introduction to the VIM2 Module	Introduces you to the software and hardware features of the VIM2 module, lists hardware requirements, and shows application examples.
3	Theory of Operation	Introduces the operating principles behind the VIM2 vision tools.
4	Staging Considerations for Vision Applications	Discusses considerations for maximizing image quality during inspections, such as lighting options, lens functionality and selection, and workpiece positioning in the field of view.
5	Connection and Powerup	Provides installation procedures and power-up indications for the VIM2 module.
6	Quick Start – Introduction to the User Interface	Describes how to use the VIM2 light pen/monitor interface, and introduces the VIM2 menus and icons.
7	Operating Environment	Describes how to configure the VIM2 module for hosted or standalone modes, and how to select the trigger source.
8	Image Acquisition and Brightness Compensation	Describes how to use the VIM2 module lighting brightness compensation and threshold images, how to select the image acquisition mode, and how to enable or disable the strobe mode.
9	Using Line Gages	Describes how to select, position, and use line gages 1-22 to inspect a workpiece.

What this Manual Contains
(continued)

Table 1.A (continued)
Manual Contents

Chapter	Title	Summary
10	Using Windows	Describes how to select, position, and use the inspection windows to inspect a workpiece.
11	Using the X/Y Float Gages	Describes how to select, position, and use the X and Y float gages in order to determine and compensate for shifts in workpiece positioning during the inspection process.
12	Using Math Tools	Describes how to use the math tools to combine, adjust, or refine the inspection tool results.
13	Run Modes and Archiving	Describes how to select the arm options, the powerup configuration, the "standard" or "learn" mode, and the runtime display. Also describes how to save configurations to EEPROM.
14	Integration with the PLC	Describes how to use the VIM2 module as part of a PLC system, including how to use discrete bits and block transfers to configure and get inspection results from the VIM2 module.
15	Using the RS-485 Port	Describes how to use the VIM2 module's RS-485 serial port to communicate with and send commands to the VIM2 module. Describes the commands for ASCII and DF1 protocols.
Appendix	**Title**	**Summary**
A	Menu Branching Diagrams	Illustrates the VIM2 module icon menus and branching scheme.
B	Configuration Blocks	Lists in table format the contents of the configuration information transferred between the VIM2 module and the PLC or other host.
C	Results Blocks	Lists in table format the contents of the inspection results information transferred from the VIM2 module to the PLC or other host.
D	Statistics Blocks	Lists in table format the contents of the statistics information transferred from the VIM2 module to the PLC or other host.
E	DF1 Protocol	Describes the structure and message format of the DF1 protocol for use in interfacing with the VIM2 module's RS-485 port.
F	Discrete Bit Addressing	Describes addressing the VIM2 module discrete bits when using two-slot, one-slot, and half-slot addressing formats.
G	ASCII Conversion Table	Use to encode and decode ASCII characters.
H	Definition of Terms	Defines terms used in this manual.
J	Specifications	Provides specifications for the VIM2 module.

Audience

This manual assumes you are familiar with your inspection application, and have determined that the VIM2 module is suitable for that application. It is beyond the scope of this manual to describe the VIM2 module set-up procedure for a specific application.

Basic electrical and mechanical experience is required for wiring, connecting, and mounting the VIM2 module, power supply, rack, and other support equipment (see Chapter 5 for hardware and connection information).

No PLC or computer programming experience is required in order to use this manual to set up the VIM2 module where no PLC or serial host computer is used. If, however, you intend to integrate the VIM2 module in a PLC system, you should be familiar with the Allen-Bradley PLC and other devices you are using, and have experience programming ladder-logic. If you are connecting a computer or other device to the VIM2's RS-485 serial port, you should be able to create a communications program which supports either of the two protocols, ASCII or DF1 (refer to Chapter 15 for more information regarding the RS-485 port).

Warnings and Cautions

Warnings and Cautions occasionally appear in this document. They are included in order to protect both you and the equipment. They appear as follows:

Warning: A warning means that **people** might be injured if the stated procedures are not followed.

Caution: A caution is used when the **equipment** could be damaged or performance seriously impaired if stated procedures are not followed.

Related Publications

The following publications are cited in this manual where appropriate. Consult your local Allen-Bradley representative for ordering information.

Grounding and Wiring Guidelines -
Publication Number 1770-4.1

Mounting Instructions for 1771 I/O Chassis and Power Supply -
Publication Number 1771- 4.5

PLC 3 Installation Manual -
Publication Number 1775-6.7.1I

PLC 5/15 Processor Manual -
Publication Number 1785-6.8.1

PLC 5/15 Assembly and Installation Manual -
Publication Number 1785-6.6.1

PLC 5/250 Installation and Configuration Manual -
Publication Number 5000-6.4.7

Solid State Control, General Information -
Publication Number SGI-1.1

A System of Universal I/O -
Publication 1771-1.2

Mounting Dimensions for 1771 I/O Chassis and Power Supplies -
Publication Number 1771-4.5

PLC Controllers 2/16 and 2/17 Processor User Manual -
Publication Number 1772-6.5.8

Product Data – Lights and Fiber Optic Cables
Publication Number 2801 – 2.1

Product Data – Cameras and Accessories
Publication Number 2801 – 2.3

Product Data – Cables and Accessories
Publication Number 2801 – 2.4

Product Data – Monitors and Accessories
Publication Number 2801 – 2.5

Product Data – Lenses and Optics
Publication Number 2801 – 2.6

Chapter 2 Introduction to the VIM2 Module

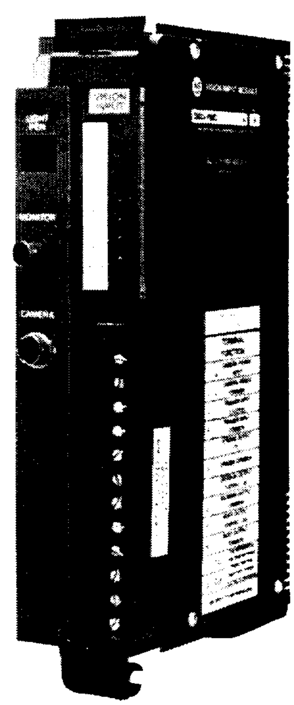
Chapter Objectives

In this chapter, we introduce the features, functions, hardware requirements, and application of the VIM2 module.

What is the VIM2 Module?

The VIM2 module provides the power of machine vision to the Allen-Bradley line of Programmable Logic Controllers (PLC). A member of the "Universal I/O" family of products, the VIM2 module gives you the ability to make non-contact inspections, and allows you to communicate the results to your PLC system or host computer. The VIM2 module can inspect for workpiece presence or absence, find edge and center locations of workpiece features, measure feature widths, and measure feature area. The VIM2 module measurements can accommodate variations in part position and workstage lighting.

Figure 2.1
The Vision Input Module



91-139-1

The VIM2 module uses a solid state video camera for image collection. The VIM2 module is simple to install, set up, and operate. PLC users will find the VIM2 module to be a natural extension of their PLC tool kit.

What is the Vision Input Module? (continued)

The VIM2 module (Cat. No. 2803-VIM2) is a dual-slot intelligent I/O module, which mounts into a standard 1771 I/O chassis. The VIM2 module can be integrated into your process to inspect products and provide direct feedback to the system's PLC terminal for closed-loop process management.

The VIM2 module can also accommodate a host computer or controller through its RS-485 port. You can communicate with the VIM2 module through the serial port using either of two communication protocols (DF1 or ASCII).

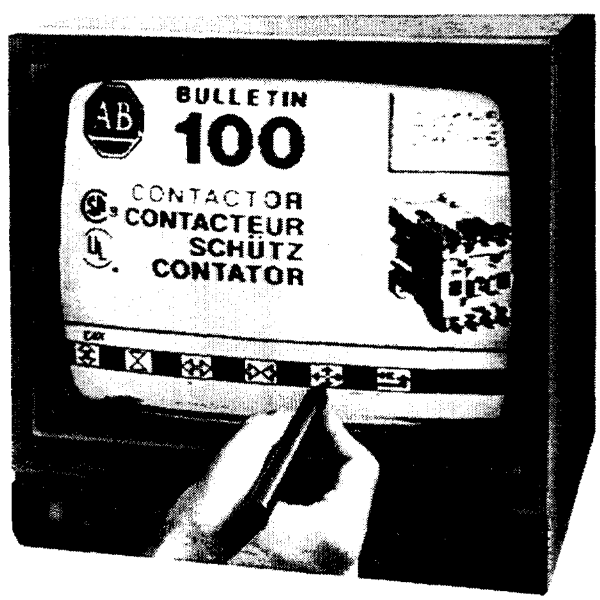
Note: See Chapter 15 for more information regarding the VIM2 module's RS-485 port.

The VIM2 module can be connected to either or both the PLC and serial line host, and still operate in standalone mode for process control.

Light Pen/Monitor User Interface

The VIM2 module is easily configured through icons displayed on the monitor screen. You simply "pick" the icon that corresponds to the function you want to activate by pressing the tip of the light pen against it. The icons appear in logically organized groups called "menus." Selecting some menu icons leads to the display of other related icon menus. This "menu branching" is organized to allow you, for the most part, to complete each specific part of the set up procedure within a particular menu branch.

Figure 2.2
Light Pen/Monitor Interface



87-283-7

VIM2 System Integration

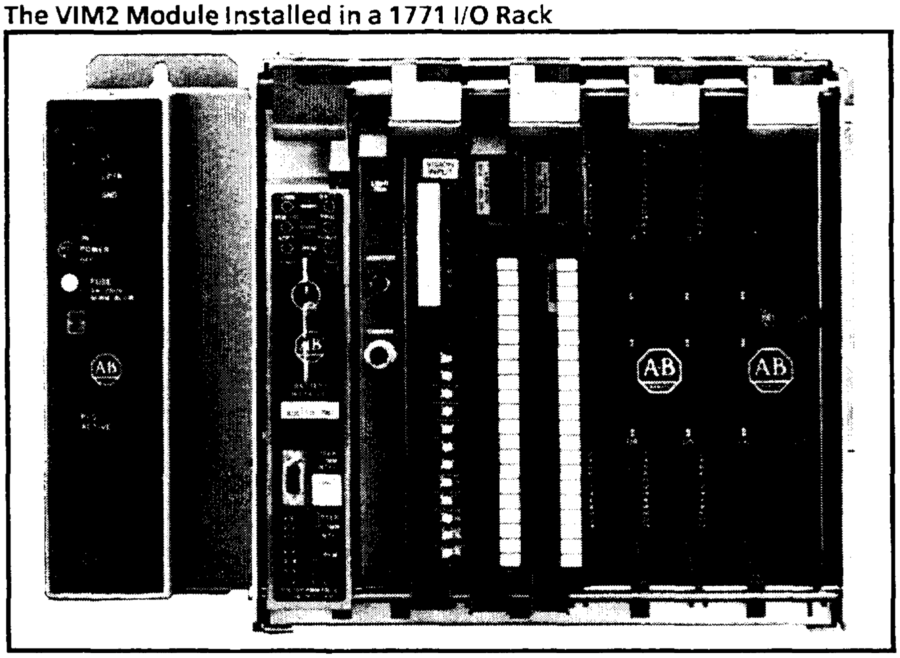
The VIM2 module is a member of the "Universal I/O" family of products, using the same racks, power supplies, and swing-arm terminations found in all Allen-Bradley 1771 systems. The VIM2 module installation requires a 1771 I/O rack and power supply, in addition to the VIM2 module and camera hardware. The VIM2 module occupies two slots in a standard 1771 I/O rack (see Chapter 5 for installation procedures). The VIM2 module may be installed into existing PLC I/O racks (see Figure 2.3).

Figure 2.5, page 2-5, illustrates the available paths for vision inspection results, statistics, and configuration data between the VIM2 module and PLC, and/or between the VIM2 module and other host (via the RS-485 port).

Integrating the VIM2 Module with PLC systems

You can store many different VIM2 module configurations using the PLC controller, and download the appropriate configuration into the VIM2 module as needed. You can also obtain comprehensive tool inspection results and statistics. The configuration and results data may be remotely managed through the Allen-Bradley Data Highway. The VIM2 module itself can store 2 different configurations in non-volatile EEPROM.

Figure 2.3
The VIM2 Module Installed in a 1771 I/O Rack



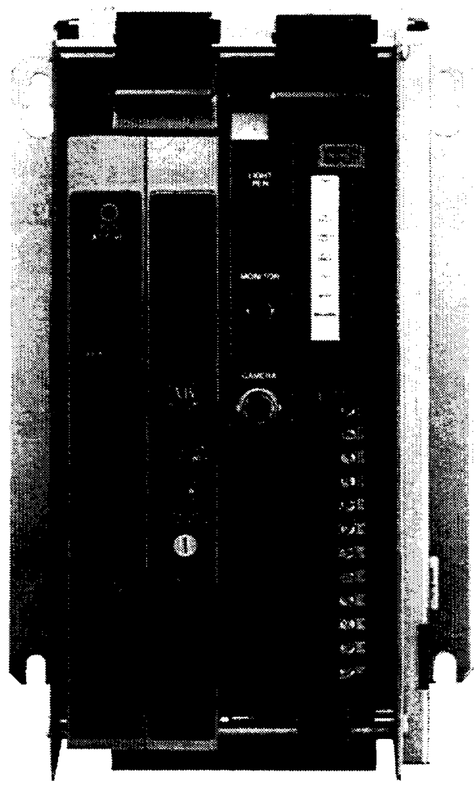
87-282-8

VIM2 System Integration *(continued)*

Using the Serial Port
As with the PLC host, you can store many different VIM2 module configurations using the serial port on the swingarm to connect to a computer through RS-485 communication. You can download the appropriate configuration into the VIM2 module as needed. You can also obtain comprehensive tool inspection results and statistics.

The VIM2 Module as a Stand-alone Vision System
The VIM2 module may be installed as a stand-alone vision system. In standalone mode, you can use the discrete swingarm I/O for process control. Discrete outputs, for example, include "Decision" (inspection accept/reject) and "Busy" (module conducting inspection, or in setup mode). For triggering inspections, there is a "Trigger" input. See Chapter 5 for more information regarding the swingarm I/O.

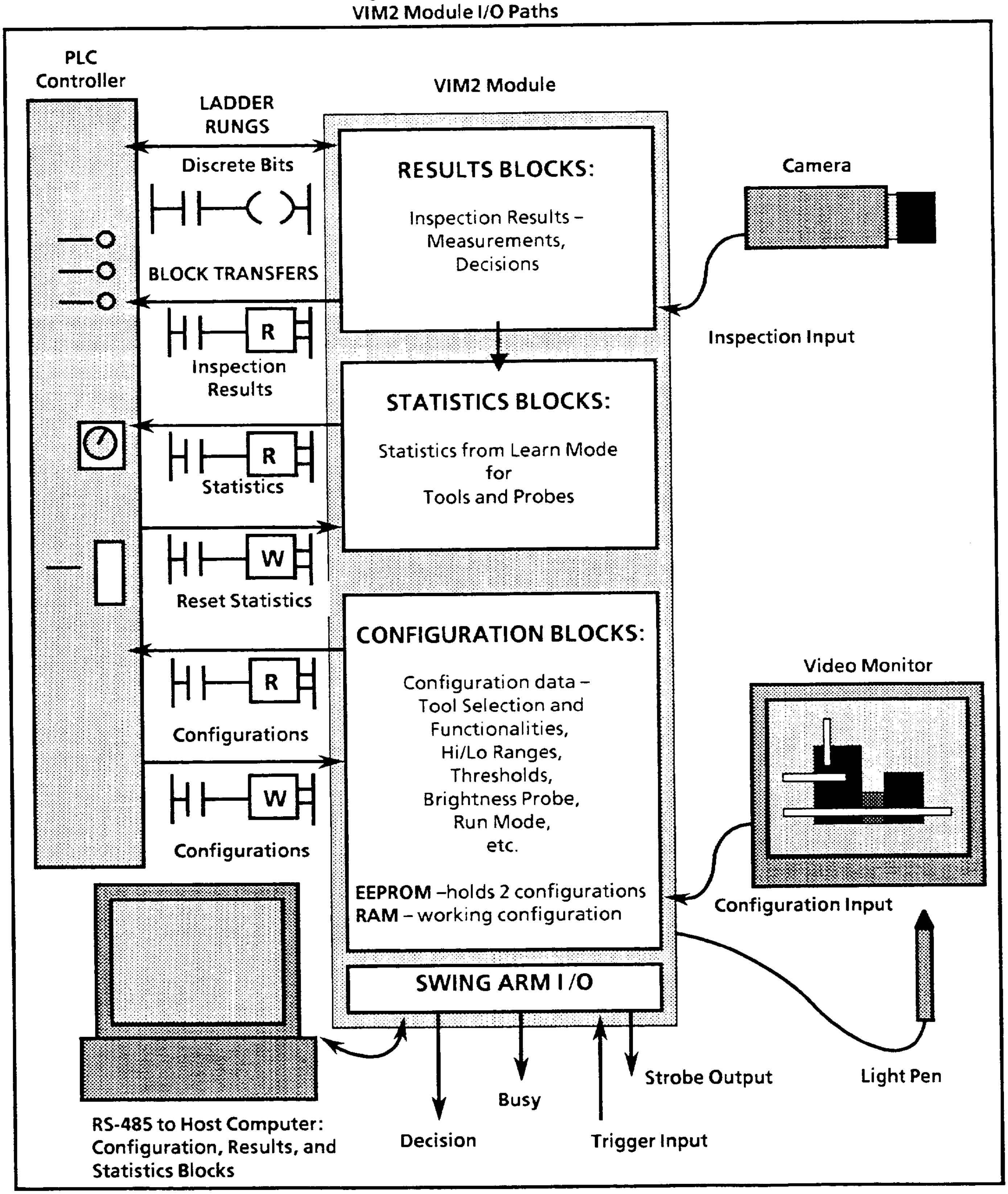
Figure 2.4
The VIM2 Module in a 1771-PSC chassis with 1771-P4 Power Supply



87-282-6

VIM2 System Integration
(continued)

Figure 2.5
VIM2 Module I/O Paths



VIM2 System Integration
(continued)

The VIM2 Module Discrete I/O

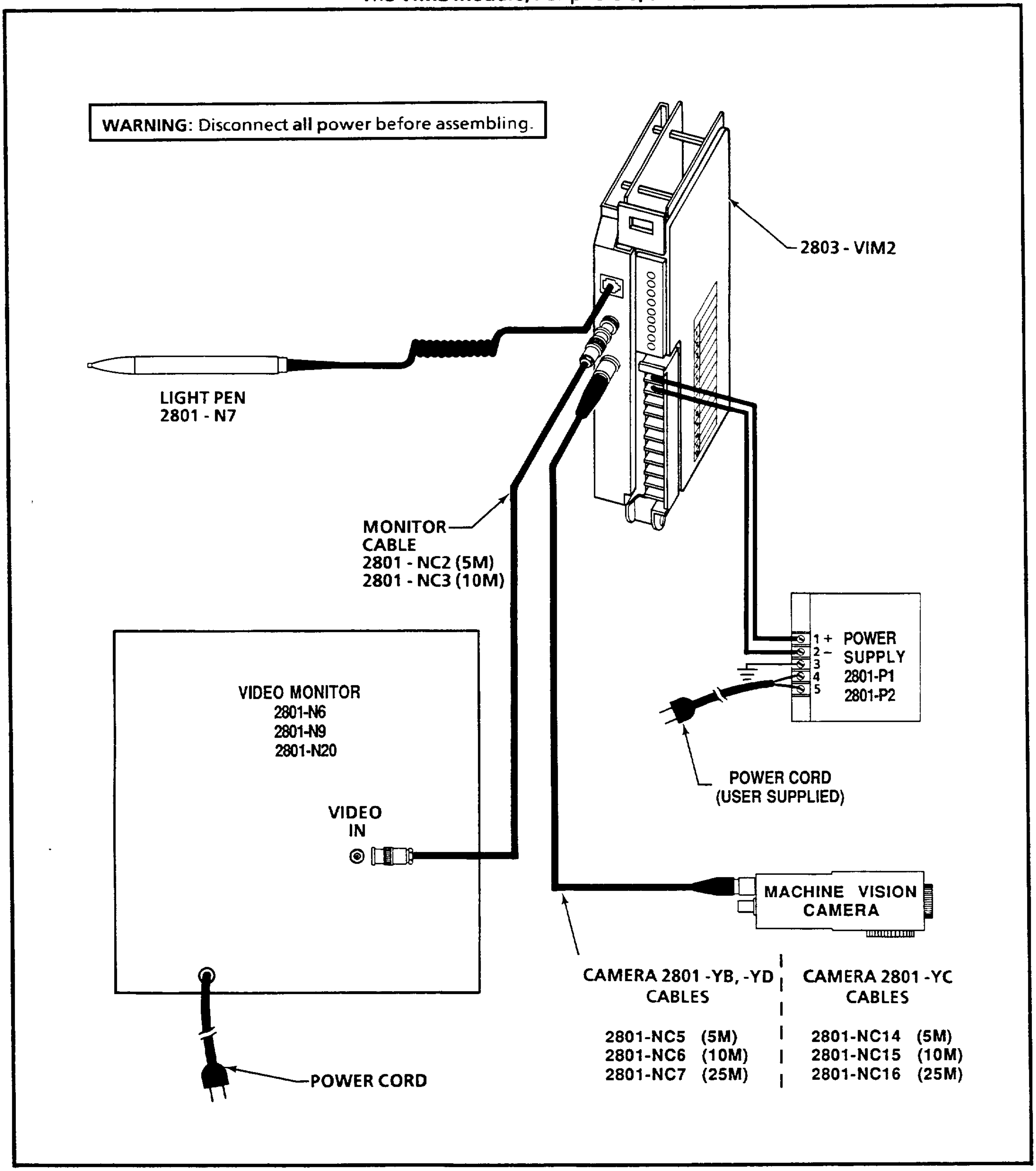
The VIM2 module has a discrete I/O interface which you can directly connect to through the swingarm terminals (see Figures 2.8 and 2.9). You can use the discrete swingarm I/O with or without the use of a PLC or other host. Discrete outputs include "Decision" (inspection accept/reject) and "Busy" (module conducting inspection, or in setup mode). There is also a "Strobe" output for synchronizing inspections with strobe lighting. For triggering inspections, there is a "Trigger" input. See Chapter 5 for more information regarding the swingarm I/O.

Hardware Descriptions

This section provides descriptions of the VIM2 module, peripherals, and cables. Figure 2.6 illustrates the VIM2 module, cables, and connections between the VIM2 module and peripheral equipment.

Hardware Descriptions
(continued)

Figure 2.6
The VIM2 Module, Peripherals, and Cables

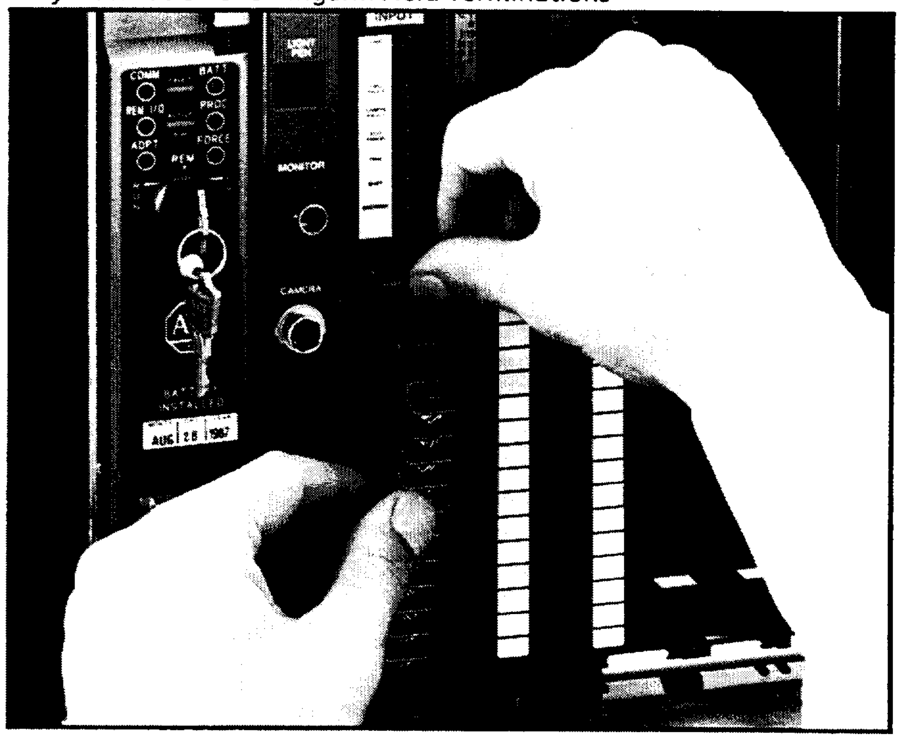


The Vision Input Module (Cat. No. 2803-VIM2)

The VIM2 module is an intelligent I/O module. The main hardware features of the module are:

- Swingarm connections, a characteristic feature of Allen-Bradley 1771 I/O modules, which consists of a removable swingarm with screw type terminals. The swingarm provides easy access to wiring terminations and is easily installed (see Figure 2.7).

Figure 2.7
Easy Installation of Swingarm Field Terminations



87-283-5

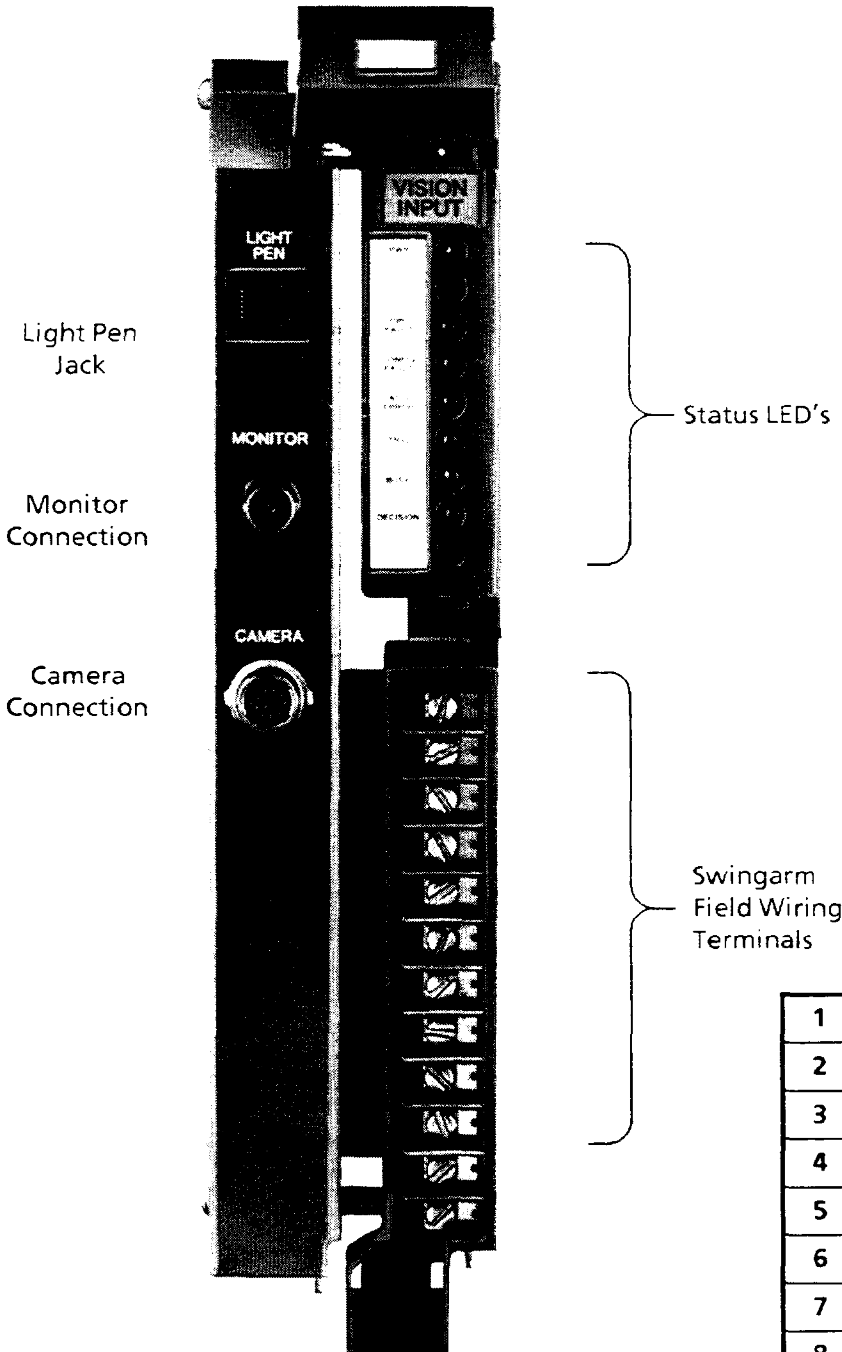
The swingarm swings neatly off the front of the module during VIM2 module removal or replacement and is easily snapped back into place. This eliminates the need to disconnect any of the hard-wired terminations for the module during maintenance and service. Swingarm terminal functions are listed in the table in Figure 2.8.

- Status LEDs – These indicator lamps light up to show the operating status of the VIM2 module. Input and output status and error conditions are indicated on the front panel LED's (see Figure 2.8).

- Front Panel Peripheral Connections – Simple plug-in type connectors provide easy connection for the light pen, monitor, and camera cables (see Figure 2.8).

The Vision Input Module (Cat. No. 2803-VIM2) *(continued)*

Figure 2.8
VIM2 Module Front Panel Features



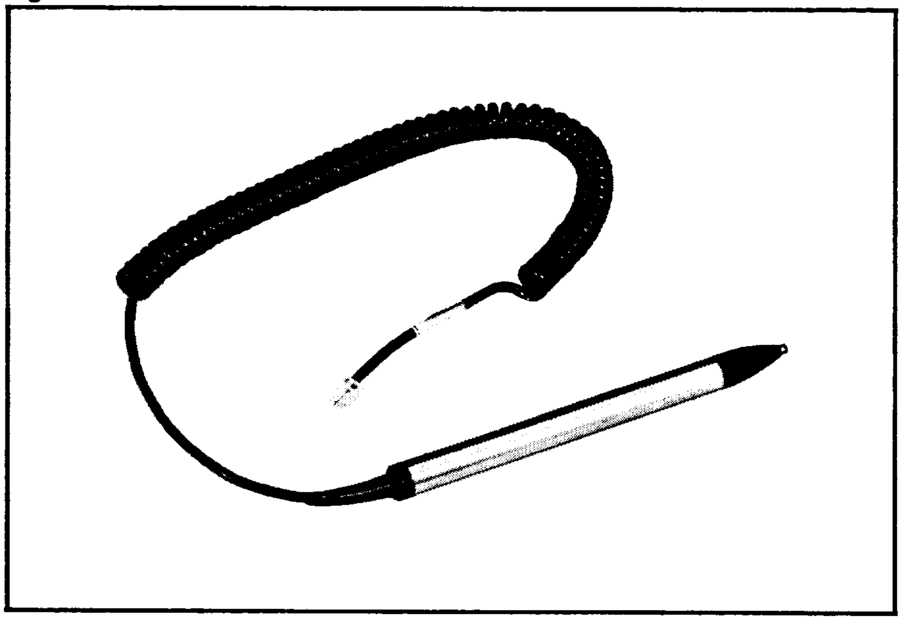
Swingarm Terminal Functions

1	12.5 VDC Camera Power Input
2	Camera Power Common
3	Trigger Input 3.3 – 32 VDC
4	Trigger Input Common
5	Strobe Output TTL
6	Strobe Common
7	Decision Output 3 – 32 VDC 1.0A
8	Decision Output Common
9	Busy Output 3 – 32 VDC 1.0A
10	Busy Output Common
11	CLL (Communication Line Low)
12	CLH (Communication Line High)

Light Pen (Cat. No. 2801-N7)

The light pen (Figure 2.9) is used in combination with the monitor screen to complete the icon-driven user interface. The pen is activated by pressing (picking) the tip against the screen. The tip reads the screen location and the module responds accordingly.

Figure 2.9
Light Pen



87-195-8

Camera

The VIM2 module uses a solid-state camera. The camera can be configured with a variety of lenses to suit individual application needs. See Allen-Bradley Publication No. 2801-2.3, Product Data, for more information regarding the selection of a video camera.

Figure 2.10
Camera and Lens

89-012-4

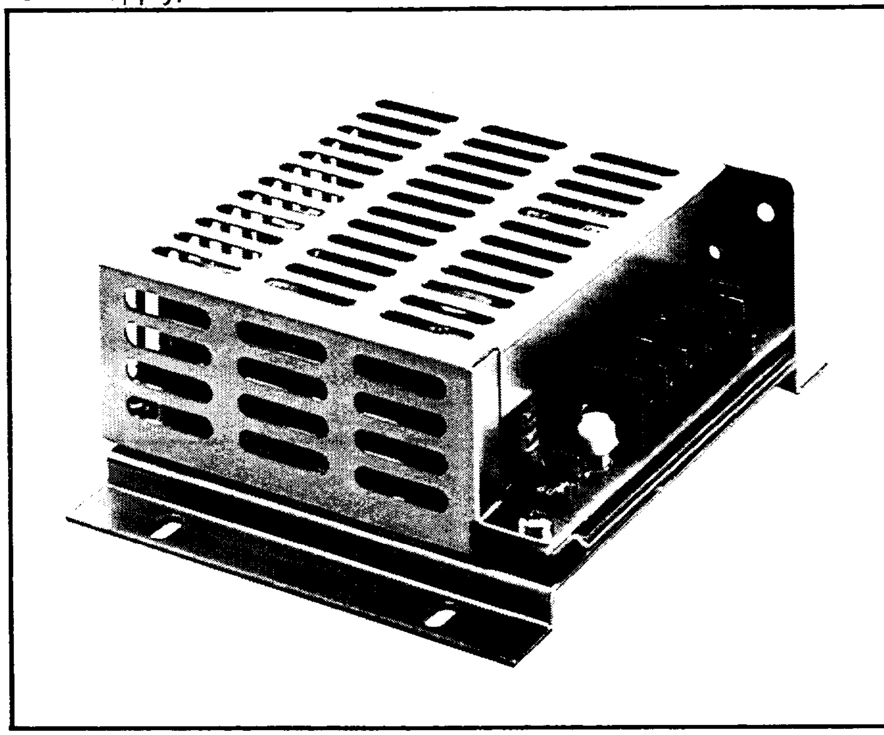
Camera Cables

The camera is available with a variety of cable lengths. See Allen-Bradley Publication No. 2801-2.4, Product Data, Machine Vision Cables, for more information regarding the selection of a cable.

VIM2 Power Supply (Cat. No. 2801-P1)

The VIM2 module power supply is an external 12.5 VDC power supply housed in an aluminum case.

Figure 2.11
Power Supply, Cat. No. 2801-P1

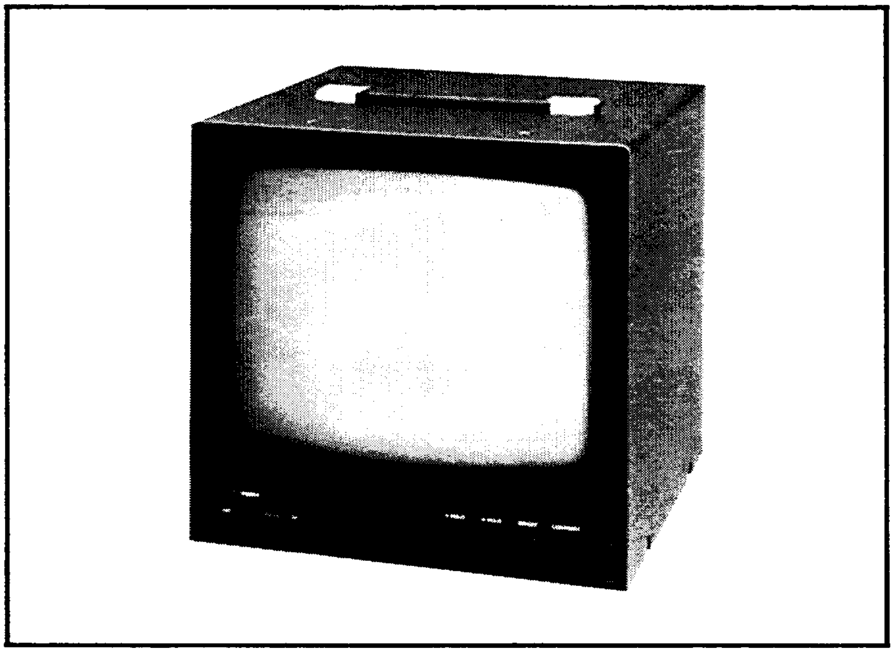


87-282-5

Video Monitor (Cat. No. 2801-N6)

The Video Monitor used for VIM2 module applications is a monochrome video monitor (see Figure 2.12). The video monitor connects to the VIM2 module front panel connector using a BNC type coaxial cable.

Figure 2.12
Video Monitor



91-044-1

Video Monitor Cables

The video connection cable from the VIM2 module to the monitor is available in different lengths. See Publication No. 2801-2.4, Product Data, Machine Vision Cables, for more information regarding the selection of a cable.

Functional Features

The VIM2 module comes complete with a set of image analysis tools which perform vision tasks. The tools include four measurement windows, 22 line gages, two "float" gages, 12 math tools, and a brightness probe. These tools are combined with the VIM2 module's ability to close the process loop through the decision output, or through direct communication to a PLC system or serial host.

Some of the key features of the VIM2 module are:

22 Line Gage Measurements
Up to 22 line gages may be set to perform any of 15 different measurements to inspect features of interest on a workpiece. Each line gage used provides two measurements simultaneously. These include measurements for edge, center, and width of "blobs." They also include counting operations for counting blobs, black or white pixels, and blob edges. You assign an acceptance ("Hi/Lo") range for each line gage measurement used to enable accept/reject decisions.

Line Gage Filtering
You can use filtering to enhance features in order to improve measurement accuracy. Filtering allows the line gage to ignore pixel "noise" in the image. One or two pixel filtering is available.

Four Window Measurements
You can use up to four inspection windows to inspect areas of interest in the image. The windows measure area by counting black or white pixels. You assign an acceptance ("Hi/Lo") range for each window measurement to enable accept/reject decisions.

Multiple Threshold Settings
The VIM2 module makes measurements based upon binarized images (see Chapter 3 for information on the VIM2 module theory of operation). Four binarized (or "threshold") images may be used, each based on a different threshold that you set. By using different thresholds, you can enhance different features in the video image. For example, you must assign each inspection line gage you use to one of the four threshold images. This allows you to reference a threshold image which enhances the specific feature(s) of interest of the workpiece for the tools you are setting up.

Brightness Probe
The brightness probe may be used to measure the brightness of the workpiece or product and to make an accept/reject decision. This tool might be used to test the intensity of a light or the brightness of a surface.

Functional Features *(continued)*

Automatic Lighting Brightness Compensation
The "brightness probe" feature may be used to monitor the light level on the workstage and adjust the threshold images to accommodate lighting variation.

Brightness Registration
You can use the brightness registration feature to automatically change all the threshold image settings, in case of a permanent change in lighting brightness.

Automatic X/Y Shift Compensation
Two line gages (the X and Y "float" gages) can be used to automatically adjust for horizontal and/or vertical shifts in the workpiece's inspection position from inspection to inspection. This allows the VIM2 module to provide measurement consistency throughout the inspection process despite variation in workpiece positioning.

Tool Registration
You can use the tool registration feature when there is a permanent change in the nominal position of the workpiece. Use tool registration to permanently change the positions of all the vision tools which reference the X and Y float gages.

Math Tools
The math tools provide a number of options for combining and/or adjusting the inspection results of the line gages and windows. Math tool operations are: logical AND, logical OR, addition of constants or results, subtraction of constants or results, multiplication, division by a constant, and minimum or maximum value of a tool's results. You can use up to 12 math tools.

Two Archived Configurations
You can store, or "archive," two different configurations in the non-volatile EEPROM of the VIM2 module. You must select which of the two configurations is the "start-up" configuration. Upon power-up, the VIM2 module downloads the start-up configuration from EEPROM to RAM. The configuration in RAM is the working configuration.

Learn Mode and Statistics
You can place the VIM2 module in a "learn" mode, perform a "trial run," and accumulate statistics. You can then refine your VIM2 module configuration parameters or your process based on these statistics. Statistics can be accumulated for windows, line gages, brightness probe, and math tools. These statistics include number of triggers processed, number of failures per tool/probe, and tool/probe minimum, maximum, and average values for inspection results falling within the user-defined acceptance range.

Chapter 3 Theory of Operation

Chapter Objectives

This chapter describes some basic principles of machine vision inspection theory, and provides an overview of how the VIM2 module inspection tools work.

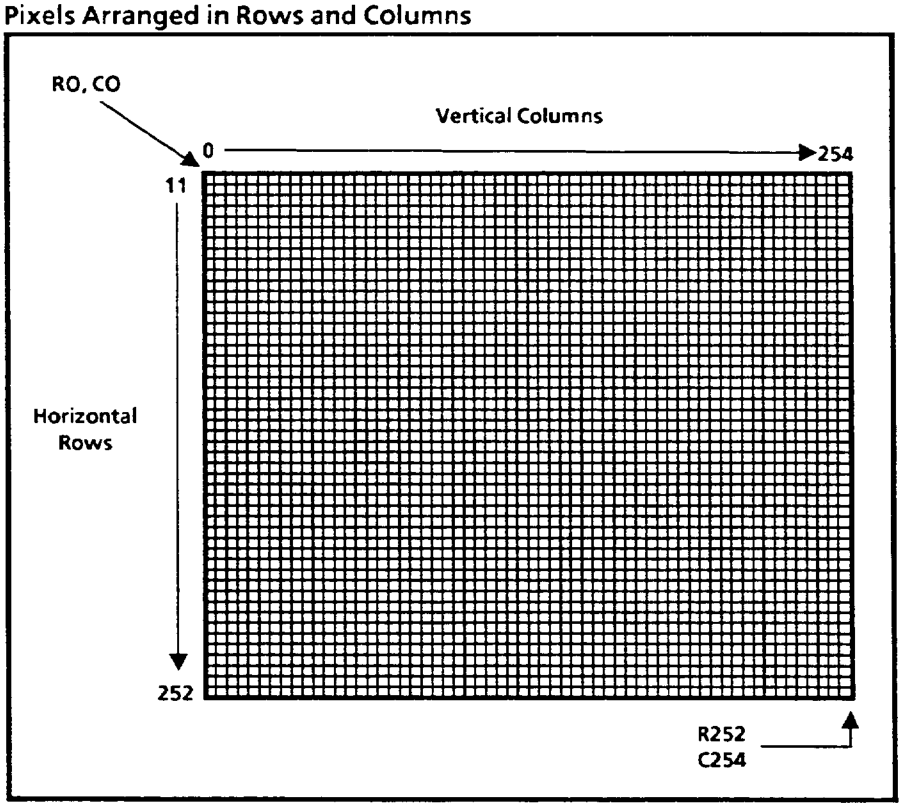
The VIM2 Module Imaging Process

The VIM2 module is designed to receive input from a solid-state video camera. The camera collects light using thousands of light-sensitive elements. Collectively, the light seen in these elements forms the "image" used by the VIM2 module for inspection.

Characteristics of Images

The video image seen on the monitor is made up of many small picture elements referred to as "pixels." The pixels are arranged in a rectangular array consisting of horizontal rows of pixels and vertical columns of pixels (see Figure 3.1).

Figure 3.1
Pixels Arranged in Rows and Columns



Characteristics of Images (continued)

The Camera Array

The camera is set up so that the image of a workpiece is focused on the camera's array of light sensitive elements. Each element responds with an electrical signal corresponding to the intensity of the light which falls upon it. These signals are then sent to the VIM2 module.

The Camera Scanning Process

The camera scans the light sensitive elements and transfers the readings to the VIM2 module using a "raster scan" method. The scan starts at the upper left-hand corner of the array and moves horizontally across the top row of pixels. It then scans across row two in the same manner. This raster scanning process continues until all of the rows are scanned.

Figure 3.2
Image Scanning Pattern and Image Coordinates

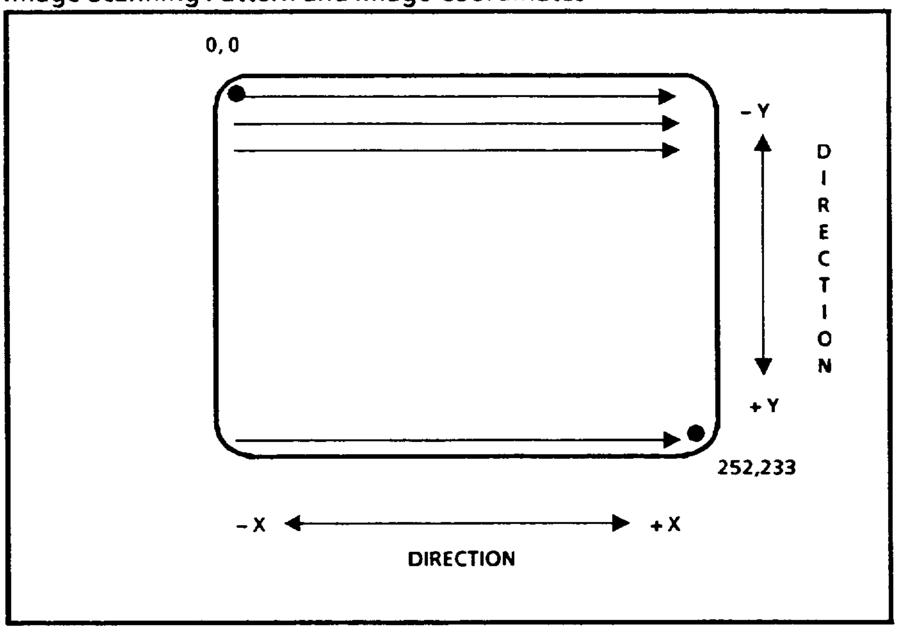


Image Coordinates

Features within the video image are located in terms of X- and Y-coordinates. The coordinate references allow the VIM2 to track feature positions and measurements (see Figure 3.2). The X-coordinate designates the pixel column. X-coordinate values increase moving to the right of the screen. The Y-coordinate designates the pixel row. The Y-coordinate values increase moving toward the bottom of the screen. All X- and Y-coordinate values are positive.

Gray Level Conversion

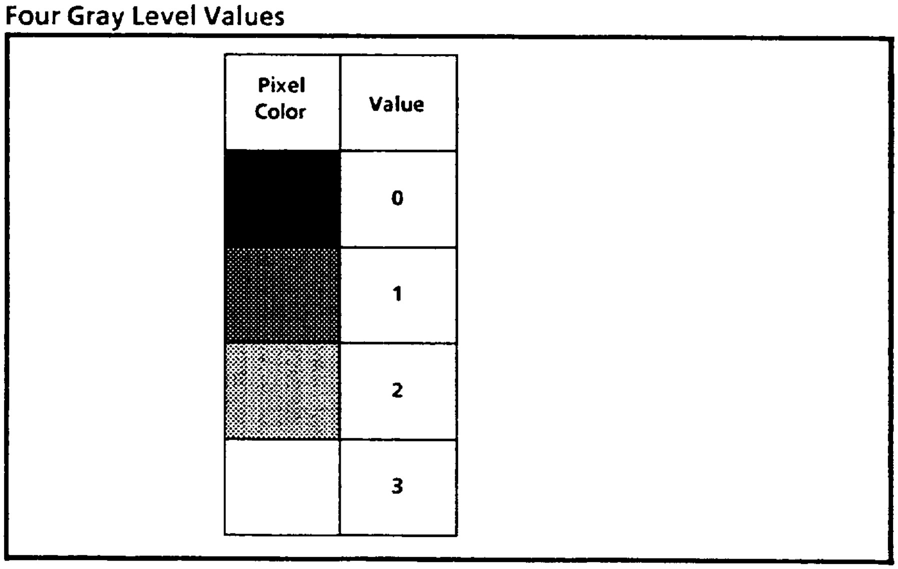
Analog values for brightness are collected by each light sensitive element in the camera's array. These analog values are converted to digital values by the VIM2 module. This is called analog-to-digital (A/D) conversion. The VIM2 module assigns digital brightness values, or "gray levels," to corresponding pixels in the rows and columns of monitor image to create a "gray-level image." Each pixel in the image has a a specific level of brightness which corresponds to the gray level value it is assigned.

The A/D conversion process creates a range of brightness levels corresponding to the range of pixel values present in the image – from black, through a wide range of gray values, to white. A gray-level image is characterized by the number of levels of brightness that can be represented; for the VIM2, this is 256.

To see how gray-level conversion works, let's look at how a simpler, 4-level converter might work. Suppose an image is collected by the camera and sent to the A/D converter, which is designed to convert each analog image signal into one of four gray levels (see Figure 3.3). Each of these gray levels represents a specific measure of brightness.

All dark gray and black analog signals collected by the camera are converted to a value of 0, for black. All signals in the medium range of gray are converted to a value of 1, for medium gray. Lighter gray signals are assigned a value of 2, and very light and white values, 3, for white. Each pixel in the resulting video image would have an assigned brightness, or gray level, of 0, 1, 2, or 3. No other, intermediate grays would appear in the image.

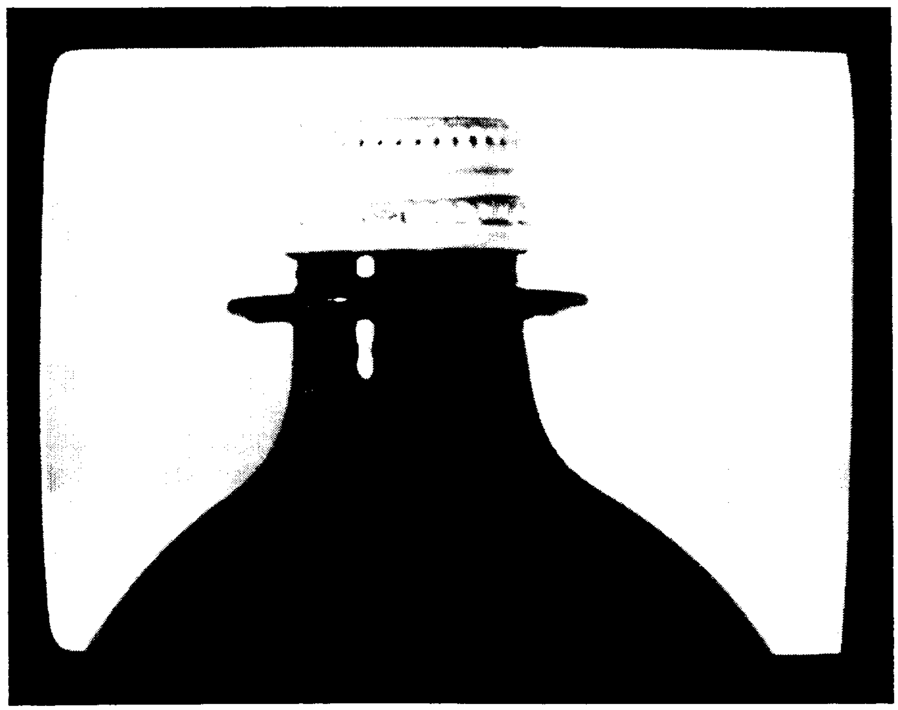
Figure 3.3
Four Gray Level Values



The VIM2 Module Gray Levels

The VIM2 module converts brightness to 256 gray levels. (**Note:** This is called "8-bit" gray scale, because 256 different values for each pixel can be encoded by using one byte (8 bits) per pixel). Images displayed using 256 gray levels look very much like analog black and white television images, with a wide range of grays in the image. Figure 3.4 shows a gray-level image. This type of image, although based on digital gray-level values, is called "analog" in this manual, because of its resemblance to an analog image.

Figure 3.4
Gray-level (Analog) Image



87-281-6

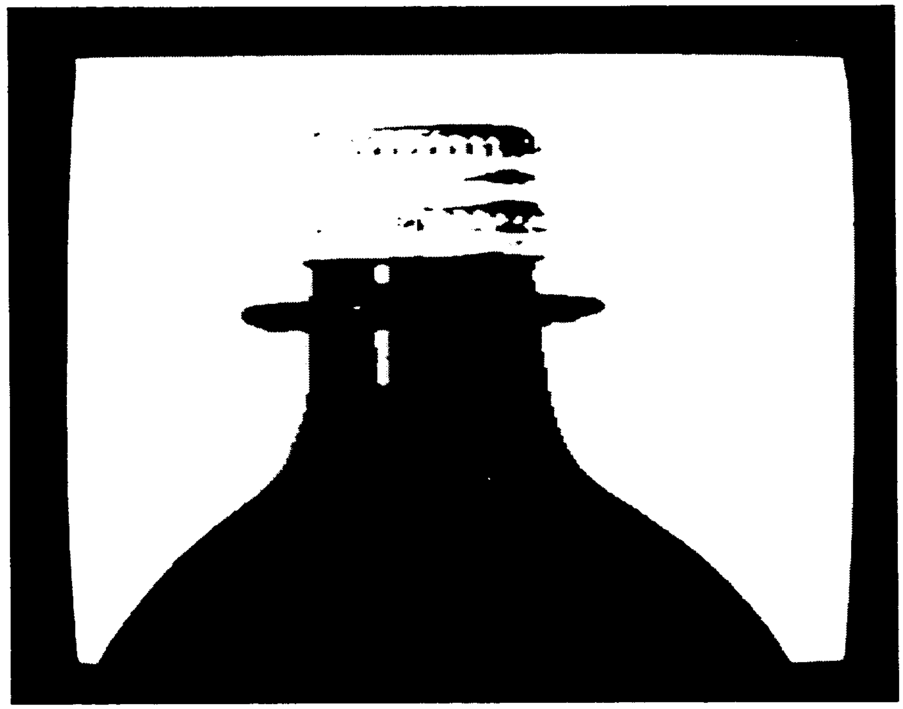
Threshold Images

The VIM2 module uses "threshold" images for image processing, which simplifies image-processing tasks. In threshold images, all pixels are assigned only one bit of information: 0 or 1 – black or white.

In creating threshold image from the gray-level image from the camera, all pixels are converted to one of two colors – black or white. To accomplish this, a "threshold level" value is used. This threshold level is the dividing line used to determine which gray levels are converted to black, and which are converted to white. All gray-level values below or equal to the threshold are converted to 0 (black) and values above the threshold are converted to 1 (white). The resultant "threshold" image shows only black and white pixels.

Threshold Images
(continued)

Figure 3.5
Threshold Image With a Low Threshold Level



87-281-8

The threshold setting can alter the appearance of the image substantially. As the threshold is increased, the image becomes darker; more gray values fall below the threshold and take on the 0 (black) value. As the threshold is decreased, the image becomes lighter; more gray values fall above the threshold and take on the 1 (white) value.

Varying the threshold can have the effect of enhancing certain image features, and/or detracting from the appearance of certain image features.

The ability to vary the threshold provides flexibility to allow you to enhance features of interest. The VIM2 allows you to set up to four different threshold values for image inspection (see "Setting Threshold Images" in Chapter 8).

The Inspection Tools: Line Gages and Windows

The VIM2 module performs inspections through the use of vision inspection "tools" – line gages and windows – and utilizes other tools as well, such as math tools, and the brightness probe.

This section discusses the theory of operation for the line gages and windows. You can use up to 22 line gages for measurement, and two more (the X and Y float gages) can be used for shift compensation. You can use up to four windows.

Setting Acceptance Ranges

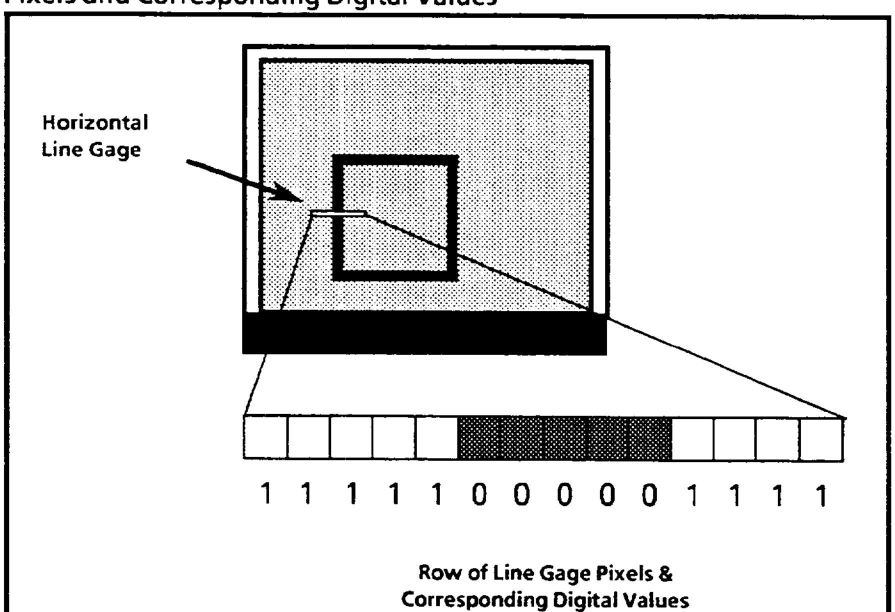
When setting up an inspection tool, you assign an acceptance ("Hi/Lo") range for that tool. You set a low limit value and a high limit value for the inspection result. If the measured value for a particular tool falls within the preset acceptance range for the tool, that tool passes inspection. If the value falls beyond the range, the tool fails. A tool failure causes the VIM2's *Decision* output to indicate a *Reject*.

Line gages

The line gages operate by inspecting a predefined segment of pixels in a row or column in the video image. Line gages are either horizontal or vertical. A horizontal line gage inspects a specified segment of a pixel row. A vertical line gage inspects a specified segment of a pixel column.

Some basics of line gage operation are previewed here.

Figure 3.6
Pixels and Corresponding Digital Values



Line gages *(continued)*

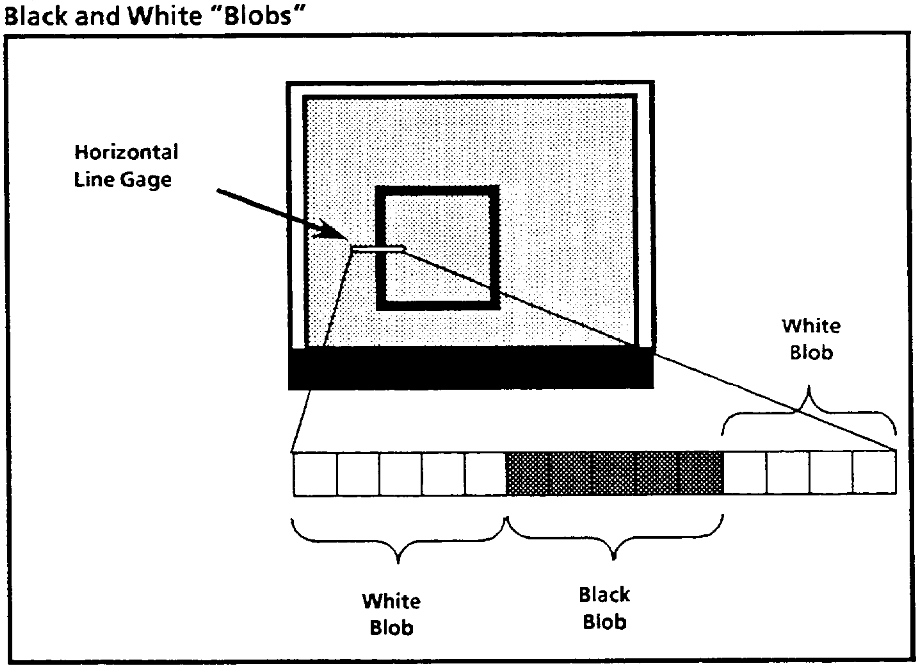
Figure 3.6 shows an example of a horizontal line gage positioned over the perimeter of a square object in the monitor image. The pixels under the line gage are read for their values by the VIM2. The line segment from the binarized image is thus represented by a string of 0's and 1's, corresponding to the value of the pixels in the line segment.

The line gages operate by analyzing line segments as strings of binary bits. These strings can be used to find: "blob" edge locations, "blob" widths, the number of edges, to count white and black pixels, and to count numbers of "blobs."

What is a Blob?

Blobs are defined as clusters of consecutive pixels of the same color (black or white). Blobs typically correspond to features in the image that the line gage crosses (see Figure 3.7). Line gages measure blob width by counting the number of pixels in the blob, and can be set to look for either white pixel or black pixel blob groupings.

Figure 3.7
Black and White "Blobs"



What is an Edge?

An edge is the point at which a white blob and a black blob meet. Line gages locate edges according to row or column coordinates, depending on whether the line gage is vertical or horizontal. This is why it is important to understand the screen coordinate system (see pages 3-1 to 3-2). Edge location is defined as the row or column location of the first pixel at the beginning of a blob. Edges detected by *horizontal* line gages are located by column number. Edges detected by *vertical* line gages are located by row number. Line gages can be set to locate the edge for either end of a blob.

Line gages (continued)

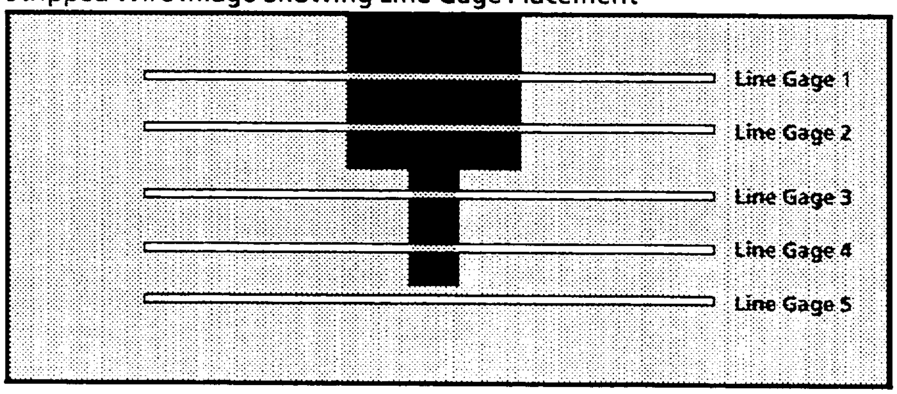
Line Gage Function – The VIM2 module offers 15 different measurement and feature counting functions based on pixel, blob, and edge detection. Each line gage used performs two measurement functions simultaneously.

Line Gage Positioning – You position line gages within the image area.

Line Gage Orientation – Line gages can be set to either vertical or horizontal orientation.

Example Inspection – Stripped Wire Dimensions
In the manufacture of cable harnesses, wires are cut to length, stripped, attached to connectors, and bundled together. Since the wire stripping process feeds the connector attachment process, improperly stripped wires cause jams and other problems for the connector attacher. Positive verification of proper wire stripping is thus a valuable control.

Figure 3.8
Stripped Wire Image Showing Line Gage Placement



In this application, a single, stripped wire end is back-lit in front of a camera so that the entire bare conductor strand and part of the insulation are visible. As shown in Figure 3.8, five horizontal line gages are positioned in order to verify that:

1. The correct wire diameter is being run;
2. The correct amount of insulation has been removed;
3. The conductor has not been severed, damaged, or bent;
4. An appropriate length of bare conductor is exposed.

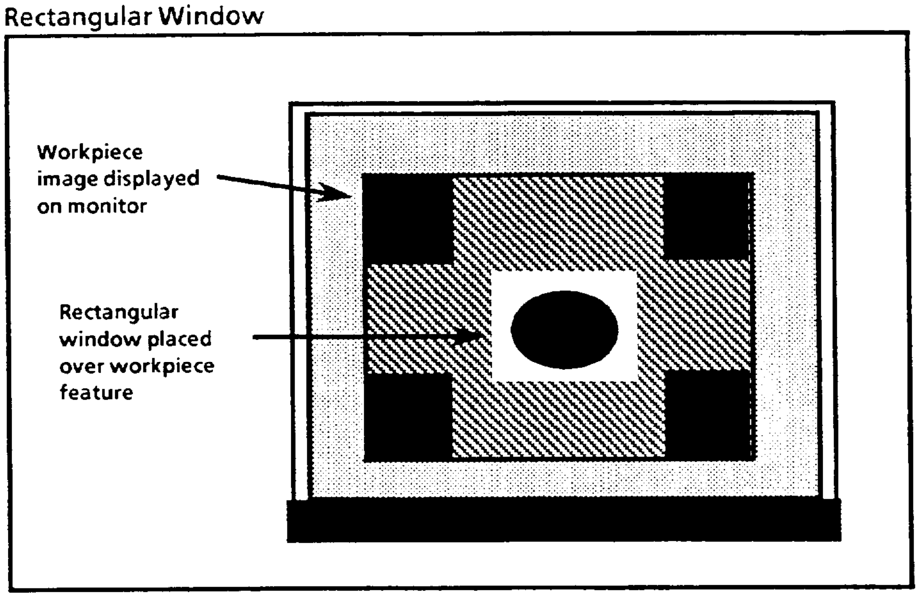
Since the monitor image has high contrast between the wire and its background, a single threshold image can be used for all gages. Image quality is relatively insensitive to light variations. Brightness compensation is not necessary.

Windows

Windows are area measurement tools. Windows count the number of pixels in the area over which the window is positioned (see Figure 3.9). Each window can be set up to count either white or black pixels. You define the window size, shape, and location (see Chapter 10 for more information on windows). You can use up to four windows for your vision inspection task.

In setting up a window, you learn the acceptable range for pixel count using one or more good (nominal) workpieces. The specific feature measured, such as a screw, label, or hole, provides a specific pixel count reading. The pixel count is proportional to the surface area of the feature in the window. You select the pixel color you want to count, then set an acceptance range that checks the measurement and makes an accept/reject decision.

Figure 3.9
Rectangular Window



Setting Window Shape – Each window can be set to one of several different shapes, including: a rectangular window, a right angle triangle with four possible orientations, a circular window, a "donut" window, and a "train-through-the-lens" mask window.

Window Positioning – Each window can be positioned anywhere within the image area, with this exception: the top 48 rows of the screen cannot be used for windows in some cases if brightness compensation is used (see Chapter 10 for more information). The "train-through-the-lens" windows cannot be moved.

Windows *(continued)*

Setting Window Size – Window sizes are individually adjusted. **Note:** The "train-through-the-lens" windows cannot be adjusted for size.

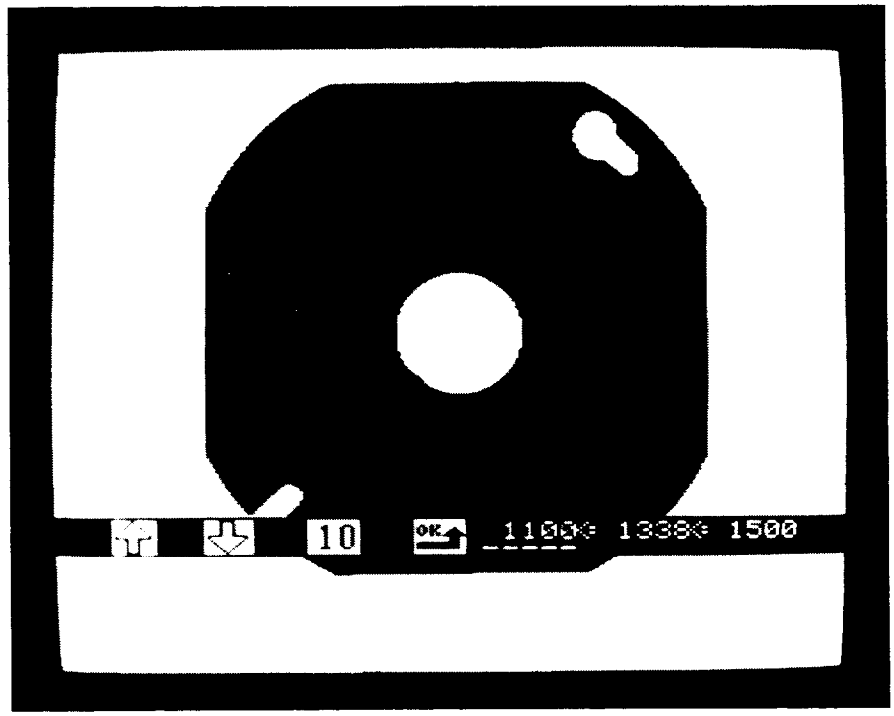
Counting Pixels – The windows operate by counting the number of pixels (black or white) in the window. "Train-through-the-lens" masks count pixels which lie under white areas of the window. Donut windows count the number of pixels under the window, minus the number of pixels covered by the donut "hole."

Example Inspection – Test Punched Holes
Punched hole presence/absence is a simple example of a windowing application. The task is to check for the presence/absence of a hole in a workpiece. The hole is backlit and appears as a white circle. A window is set to view the area where the hole should be found (the window is seen as the gray area over the hole in the part). If the hole is not large enough, or fails to clear through the part, there will be too few white pixels in the image.

During setup, using a known good (nominal) part, the acceptance range is set to detect when there are too many or too few white pixels, and to output an accept/reject signal. Figure 3.10 shows an acceptable hole which has been set up for verification using a circular window. The "Hi-Lo" limits are set to 1100 and 1500. The actual measurement reading of this hole is 1338, indicating an acceptable hole. Figure 3.11 shows an unacceptable part. Notice that the reading is 133, which is well below the low range limit (1100), indicating an unacceptable hole.

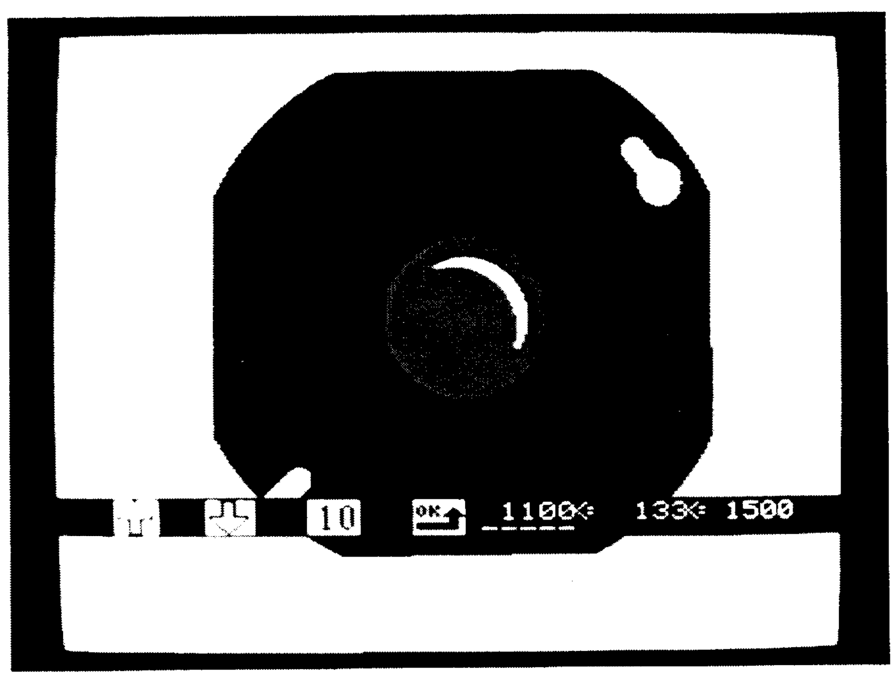
Windows
(continued)

Figure 3.10
Hole Presence Verification Using a Circular Window
Image of a Properly Punched Hole



87-281-13

Figure 3.11
Hole Presence Verification Using a Circular Window
Image of an Improperly Punched Hole



87-281-14

(This page intentionally left blank)

Chapter 4 Staging Considerations for Vision Applications

Chapter Objectives

The objective of this chapter is to provide information for maximizing image quality during inspections through the application of "staging," which includes such considerations as lighting options, lens functionality and selection, and workpiece positioning in the vision system's field of view.

Image Quality

Successful vision applications require image quality. Image quality refers to the quality of the presentation of the workpiece or workpiece features to the machine vision system. Image quality helps determine the feasability, precision, and consistency in vision inspections. To use the VIM2 vision system effectively, present a quality image to the VIM2 vision system.

One way to determine the feasability of your application is through the use of the video monitor. For example, keep the following rule of thumb in mind when setting up your vision application using the VIM2 vision system:

> *If you don't see workpiece features of interest clearly in the video monitor, then you probably won't be able to inspect it with the vision system.*

Image quality is accomplished through the process of "staging," which is the topic of the remainder of this chapter.

Staging

The term "staging" refers to the application of lighting, camera, and lens to present the workpiece features to the vision system for inspection. Through the process of staging, you create a vision "workstage," the area where the vision inspections take place. There are two main objectives in the design of a workstage:

1) To make the features of interest clearly visible

2) To reduce clutter in the image and eliminate irrelevant features from the image.

This chapter discusses specific tools and techniques to achieve the required image quality. Two key elements which affect image quality, focus and image contrast, are discussed in the next two sections.

Focus

The collection of light through the camera lens is a critical step in the formation of the image. The lens must be set at the proper distance from the workpiece and then properly focused. Lens focus determines the sharpness of the features in the image.

Image features such as edges and thin sections must be clearly focused to prevent a loss of clarity due to blurring. Blurred images may change appearance as the binary threshold levels are changed. This is because features like edges may appear as blurred transitions from white to gray to black and threshold changes move the apparent positionof the edge along with the changing gray level of the blurred edge. This problem is not encountered when edges have crisp contrast with no blurring.

Blurring can be caused by rapid motion in high-speed applications. In this case, you can use a strobe light to stop the action and eliminate the motion blurring.

Poor focus also impairs the ability to see small detail. Small features may be blurred and lose the sharp definition required for precision measurements.

If you wish to inspect features at different distances from the camera, you may want to increase the depth of field (depth of focus) of your lens. To do this, you narrow the aperture (increase the F-stop setting), and add more light to the workstage to compensate for the loss of brightness. You can also restrict the depth of field to defocus a busy background. In any case, experimentation is critically important.

Image Contrast

Image contrast is extremely important to the successful application of the VIM2 module. An ideal contrast situation can be created through the use of backlighting. Back-lighting illuminates the object from the rear. The workpiece blocks the light and appears as a solid black silhouette against a bright, white background. This distinct contrast in gray scale, from the very dark object to the very bright background, makes it very easy to set an acceptable threshold and makes the vision system relatively insensitive to small variations in light level. Features that have only minor variation in gray-scale intensity (low contrast) are more difficult to separate.

When setting up your application, try to create as much crisp, well-focused contrast as possible between features of interest and the background. This ensures that the thresholds will be more easily set and that consistent measurements will more easily be obtained.

The Importance of Lighting

We have seen that focus and image contrast are key elements in image formation. Both of these elements are highly dependent upon the lighting provided in the inspection area, or workstage. Because light is the medium used by the vision system to make measurements, it is crucial that the workstage lighting both be appropriate for the intended inspection, and also be consistent from one inspection to the next.

You should light the workstage as best suits your application. Before lighting the workstage, you should consider workpiece's features, color and reflectivity, as well as the background, in order to determine which type of lighting works best.

The image received by the vision system is partially the result of direct and reflected light. The camera "sees" only the light intensity; it does not see color. The gray level of a feature in the workstage is determined by the interaction of light and the surfaces of objects in the workstage area. When light strikes a surface, it can be absorbed, transmitted, or reflected.

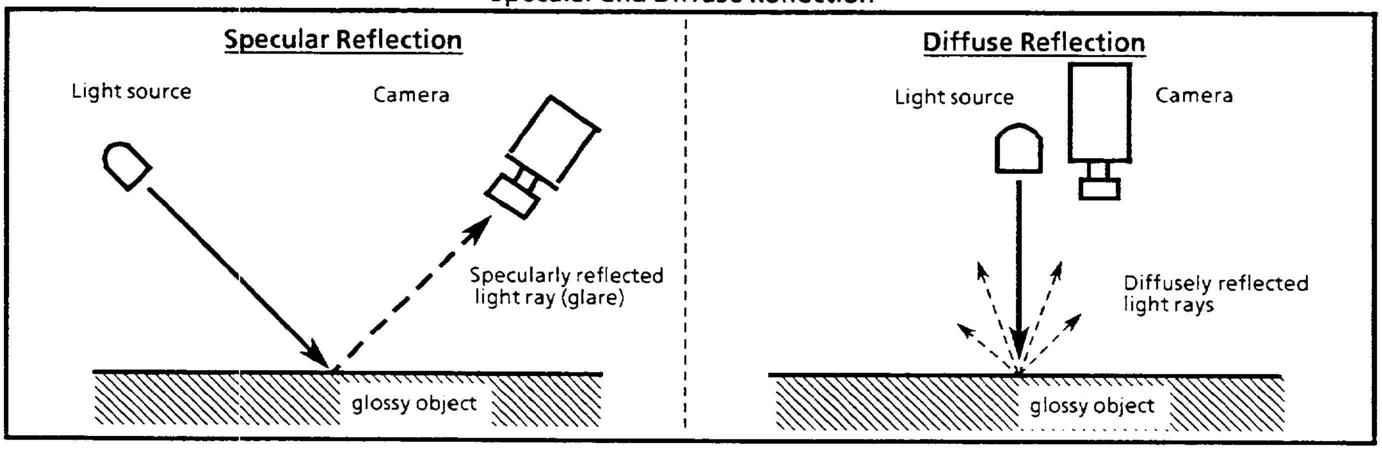
- **Absorbed Light**
  A red object appears red because all the light rays except red are absorbed. Dark objects absorb a lot of light. Light objects absorb very little light and reflect most of it away.

- **Transmitted Light**
  Light passes through many types of glasses and plastics. The light path is often radically modified by this transmission. This light is called transmitted light.

- **Reflected Light**
  Light that is not absorbed or transmitted is reflected. The two types of reflected light are specular and diffuse, and usually both types are present. A glossy magazine cover exhibits both types of reflection. When held at a certain angle, light is reflected directly into the reader's eyes (specular reflection), and the print cannot be observed. When tilted slightly, the specular reflections are directed away from the reader and the diffusely reflected light allows the print to be observed. Figure 4.1 shows both types of reflection.

The Importance of Lighting
(continued)

Figure 4.1
Specular and Diffuse Reflection



Different Types of Lighting

Several types of lighting devices may be used with the VIM2 module. Many of these may be ordered directly from Allen-Bradley as accessories. See your local representative for details.

1. **Incandescent Lighting**
   Incandescent lamps are popular because they are economical and their intensity is easily adjusted. However, ordinary incandescent lamps have limitations since they exhibit a constant degradation in light during their operating life. Using a halogen light source results in a more consistent light output.

2. **Fluorescent Lighting**
   Fluorescent lamps produce less heat than incandescent lamps yet produce the same amount of light. Some fluorescent lamps, in multiple-lamp fixtures, provide large, diffuse illumination. Circular fluorescent bulbs are excellent for illuminating small objects. Some fluorescent lamps flicker at a rate of 60 Hz. This is the same rate as the video frame rate. This flicker may cause some jitter in the image. This jitter is most apparent in image areas such as edges.

Different Types of Lighting (continued)

3. Strobe Lamps
   When an object is moving past the camera at high speeds, a strobe lamp flash of light can "freeze" the motion to create clear images. The strobe produces high intensity light for a very short period of time. The brightness of the strobe flash may vary from flash to flash.

   The timing of the flash must be synchronized so that the workpiece is present when the camera scans the area. The VIM2 module can trigger an accessory strobe light through the trigger output on the swingarm.

Methods of Lighting

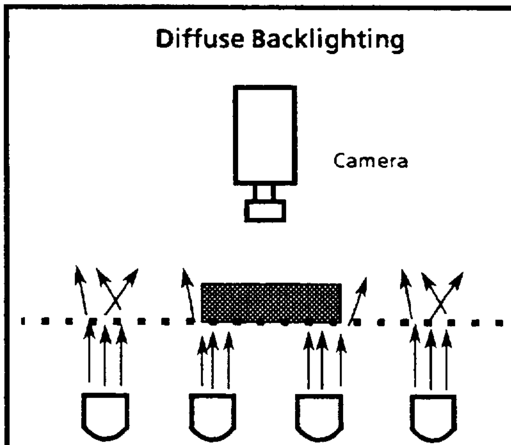
In addition to evaluating what type of lighting is most appropriate, you must also determine the optimal placement of the light source(s).

Lighting methods fall into two categories – direct lighting (backlighting), and indirect lighting (front lighting).

Direct Lighting (Backlighting)

When direct lighting is used, light travels directly from the source to the camera lens, and is not reflected. The workpiece is placed between the light source and the lens, producing a silhouette. Direct lighting is also called backlighting because the light comes from behind the workpiece. Backlighting allows detection of edge features only. Features that fall within the workpiece edges are not detected. Figure 4.2 shows an example of diffuse backlighting.

Figure 4.2
Example of Diffuse Backlighting



This form of direct illumination is useful when a high-contrast silhouette is required. This is the most easily constructed of all backlighting methods. This approach works especially well with flat workpieces; a well defined silhouette is produced.

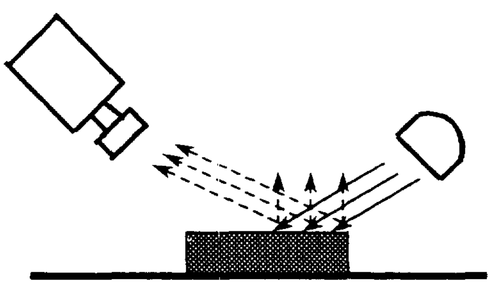
Indirect Lighting (Front Lighting)

When using indirect lighting, light is reflected from the workpiece to the camera lens, making surface features visible. This technique is also called front lighting. Front lighting allows the vision system to capture normally visible features of the part and to make distinctions based upon gray-level appearances. It may also cast shadows or create reflections which may or may not be useful.

Indirect lighting is used when surface features must be inspected, or where backlighting is not possible. Figure 4.3 illustrates several examples of indirect lighting.

Figure 4.3
Examples of Indirect Lighting

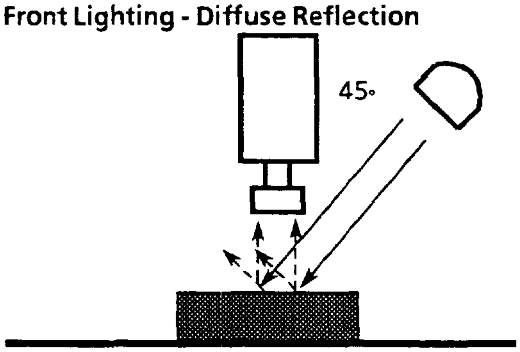
Front Lighting - Directed Bright Field



Directed bright field lighting places the camera near the angle of reflected light. This is the area of brightest intensity in most cases. This is particularly true of highly reflective parts with glossy surfaces or light colors.

The shadows cast by bright field lighting may be used to create strong contrast between a feature in the workstage and its shadow. Use directed lighting to create strong contrast between features of interest and their background areas.

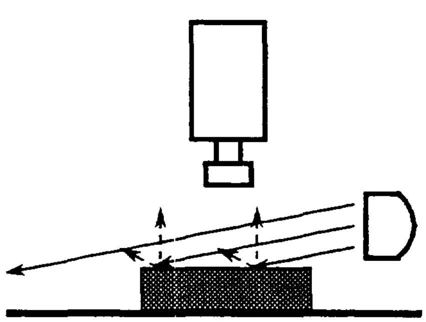
Front Lighting - Diffuse Reflection



Diffuse reflection is usually preferred to directed bright field lighting because it offers a wider range of gray values for image analysis. This is because the image is the result of the light absorbing and diffusing qualities of the features, not the reflecting qualities.

A commonly used method is to place the illumination source at 45 degrees to the workpiece surface and the camera viewing angle, assuming the camera is perpendicular to the surface.

Front Lighting - Directed Dark Field



Directed dark field is a side lighting technique where the angle of illumination (angle of incidence) is very shallow. A very small amount of the diffusely reflected light reaches the camera, and the surface appears dark. Any abrupt change in surface height causes a bright reflection into the camera's lens, often with an accompanying dark shadow next to it. Surface flaws such as scratches and bumps are often detected using this technique.

Indirect Lighting (Front Lighting) *(continued)*

Bright Field Front Lighting

The image is greatly impacted by the angular relationship of the lighting and camera. This is because most reflected light bounces off of a surface at an angle equal to the angle of incidence (angle of approach). The camera lens collects the most light if its angle of incidence is along the axis of the reflected light. This bright reflection creates a bright image in the camera field of view and is referred to as "Bright Field" lighting (see Figure 4.3). Bright Field lighting can produce glare which may or may not be useful.

Diffuse Front Lighting

Diffuse lighting moves the camera out of the main reflected beam to collect the secondary light that is diffused (scattered) by the object surface. The angle is enough to reduce glare and still catch strong illumination from the diffused light.

Dark Field Front Lighting

Dark field lighting places the camera at such an extreme angle, away from the angle of reflection, that very little light is reflected or diffused to the camera and a dark image is formed. Directed dark field lighting is often used to highlight surface variations such as scratches and pits.

Backdrops

You can create high contrast between the workpiece and background simply by providing an appropriate backdrop. For example, a medium gray object appears light against a black background and dark against a white background. You can also place the backdrop far enough from the workpiece that it is out of focus enough to eliminate clutter.

Lens Selection and Adjustment

Another factor in quality image formation is lens selection and use. The lens projects the image of the workstage into the camera and onto the image collection electronics (the image array). The lens must form a sharp, even, undistorted image for consistent measurements to be achieved. A brief discussion of the subject is presented here.

How a Lens Works

Lenses bend light rays as they move from air into glass and then emerge from the glass into the air. (The degree the light rays bend depends on the angle of incidence and the indices of refraction for glass and air.) Figure 4.4 shows two simple lenses focusing light onto the image plane. In solid-state video cameras, the image plane is a photosensitive array on an integrated circuit. A broad selection of lenses is available to meet a variety of application requirements.

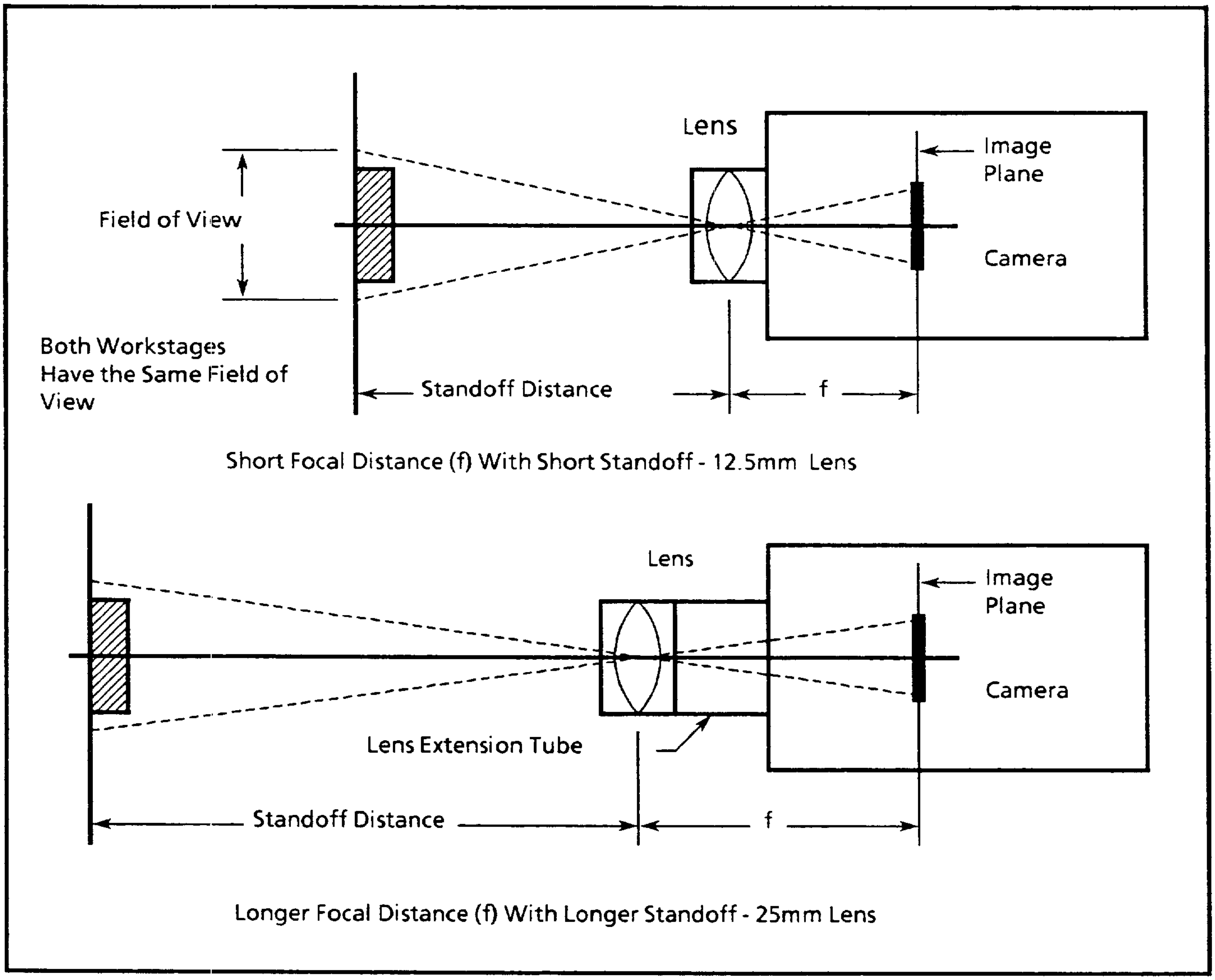
How a Lens Works (continued)

Field of View

Lens selection is largely determined by the field of view (FOV) required to see the full area of interest in the work stage (see page 4-10 to determine FOV size). When considering the required FOV, allow room for variation in part position from inspection to inspection, if applicable.

The FOV is the area (field) seen by the camera and viewed on the video monitor. There may be several lenses capable of meeting your FOV requirements. Each of the lenses has a different standoff distance for the same size FOV (see table 4.A or 4.B).

Figure 4.4
Relationship of the Focal Length of a Lens to Standoff Distance Given a Constant Field of View



How a Lens Works
(continued)

Lens Standoff Distance From the Workpiece

Standoff is the ideal distance between the lens and the item being inspected. The standoff for a given field of view is determined by the fixed array size and the focal length of the lens. The focal length is the distance between the lens center and the image plane (image array in this case) when objects in the field of view are in focus. Lenses are measured by their focal length. A lens with a short focal length, such as a 12.5 mm lens, has a shorter standoff for a given field of view than a 25 mm lens. This is illustrated in Figure 4.4.

The camera should be mounted far enough away so that it does not interfere with the process or with the workstage lighting. Other considerations might include keeping away from parts that are hot or that emit vapors or dust; all of which degrade the performance of the system.

Lens Aperture (F-stop) Settings

You can control the amount of light collected by the lens using the F-stop ring located around the camera lens. The F-stop acts much like the pupil of your eye, controlling the amount of light that comes in contact with the lens.

The F-stop number indicates the relative amount of light that passes through the lens aperture. As the F-stop setting *increases*, the aperture and the brightness of the image *decrease*, but the depth of field (depth of focus) *increases* (see Figure 4.5). A typical set of lens F-stop values is 2.8, 4, 5.6, 8, 11, and 16. For each step up in F-stop value (e.g., going from 2.8 to 4), the image brightness decreases by 1/2.

Adjust the F-stop to obtain the best possible image contrast, *after* the lens has been focused.

Note: Always switch the monitor display to the "live" analog image before adjusting the camera focus or F-stop. To do this, touch the light pen to the top half of the monitor screen; repeat until the live analog image appears (assuming your VIM2 system is connected and powered up, of course – see Chapter 5 for connection and power-up information).

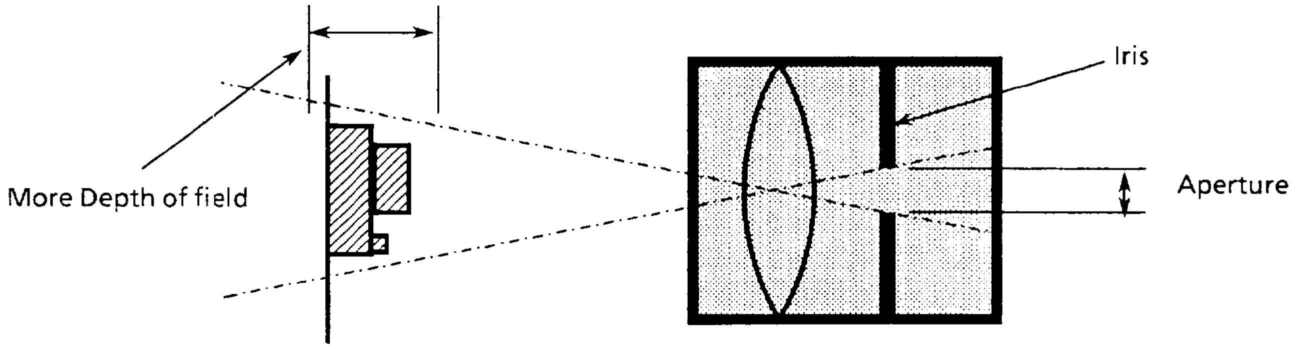
If your workpiece has features at different distances from the camera, you may need to increase the depth of field. To do this, increase the F-stop setting. If the inspected surface of the workpiece is basically two-dimensional (as in most cases), a large depth of field is not necessary, and you can use smaller F-stop settings.

Note: When you change F-stop settings, you will have to adjust the brightness compensation range and threshold image settings (see Chapter 8), and/or adjust the lighting intensity in the workstage to compensate.

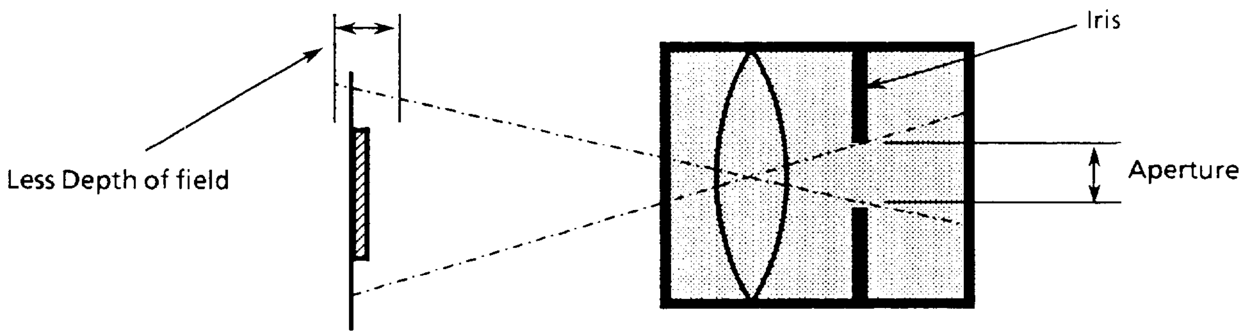
How a Lens Works (continued)

Lens Aperture (F-stop) Settings

Figure 4.5
Control of Brightness and Depth of Field Using the F-stop



Higher F-stop setting – Aperture is smaller, decreasing brightness and increasing depth of field.



Lower F-stop setting – Aperture is greater, increasing brightness and decreasing depth of field.

Lens Section View

Selecting the Lens for Your Application

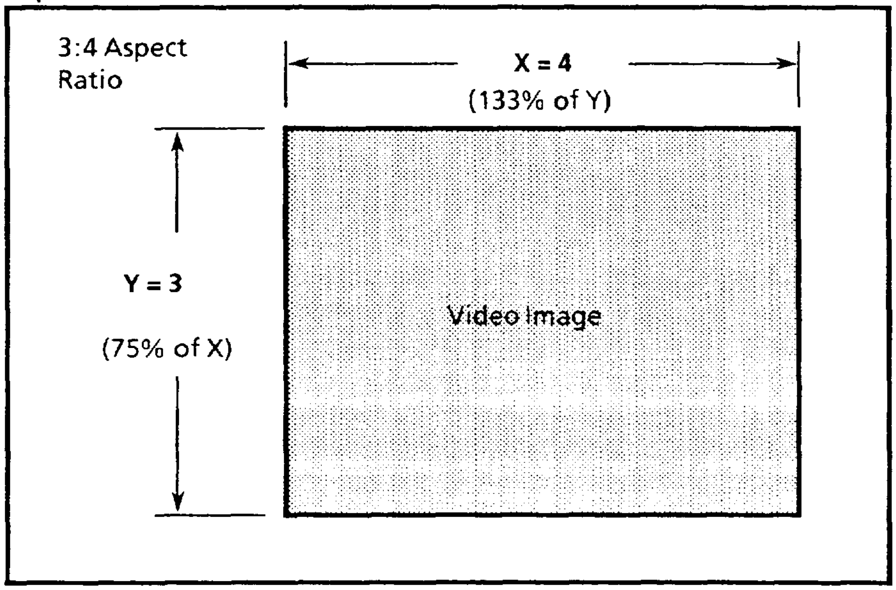
The following instructions aid you in the selection and setup of lenses to suit your applications.

- **Determine the field of view (FOV) size requirement**
  The FOV is the area that the camera "sees." The FOV is rectangular in shape (see Figure 4.6); its width is greater (× 1.33) than its height (this is due to the 3:4 "aspect ratio" of the camera and monitor).

  In setting the FOV size, the FOV should be large enough to include all the workpiece features to be inspected, and to accomodate for positional variation of the workpiece during inspections. Beyond this, a margin of 20% should be added to the FOV. Also, in setting the FOV size, the FOV should not be so large that the features of interest cannot be distinguished clearly in the video monitor.

Selecting the Lens for Your Application
(continued)

Figure 4.6
Aspect Ratio



Example FOV calculation: Suppose the features of interest on a carton are located on a 5 × 5 inch (127 × 127 mm) area. Suppose that the position variation is ± 1 inch (25.4 mm) – or 2 inches (50.8 mm) – for both horizontal and vertical positioning.

Approximate height of the FOV would be:

(5 in. + 2 in.) × 1.2 (20% margin) = 8.4 inches, or
(127 mm + 50.8 mm) × 1.2 = 213.4 mm

Approximate length of the FOV would be (since the length is greater (× 1.33) than the height):

8.4 inches × 1.33 = 11.2 inches, or
213.4 mm × 1.33 = 283.8 mm.

Thus the required FOV would approximately be 8.4 × 11.2 inches, or 213.4 × 283.8 mm (**Note:** Multiply the FOV by .75 when using a Cat. No. 2801-YD camera).

- **Determine the Standoff Distance**
Determine your ideal camera and lens standoff (distance away) from the part. This is often dictated by a clearance requirement to stay out of the way of moving machinery or workpieces. Sometimes it is limited by available floor space or ceiling height, or by a requirement to shroud the workstage. It might also be the standoff distance that is simply the most convenient for setup and maintenance.

Selecting the Lens for Your Application *(continued)*

Note: The size of the FOV is determined by two factors: the type of lens, and the standoff distance. Generally, for a specific lens type, the greater the standoff, the larger the FOV.

- **Determine Accuracy**
  Determine the accuracy to which the object must be measured.

- **Determine the Need for Extension Tubes**
  Some lenses need an extension tube to focus at close distances. The extension tube is placed between the camera and lens. If an extension tube is needed at a given standoff, the length of the tube is listed in Table 4.A.

  Note: There are techniques for folding and enlarging the standoff distance. For example, the camera can view the workstage through a mirror or prism. This allows considerable freedom in camera placement and angle.

Using the Lens Selection Tables

Table 4.A can aid you in selecting the best lens for your application. The first column of the tables lists the height, then width, for FOVs. The second column gives the accuracy to which an object will be measured at each FOV. The third column gives the pixel size. You may use the table based upon a known FOV or a desired accuracy.

Lens Selection if FOV is Known

To use the lens selection table, if the FOV is known, use the first FOV (listed in the first column of the table) that is larger than the FOV required.

On the right side of the table, the standoff distance for each lens is shown for each FOV.

Lens Selection if Accuracy is Known

To use the lens selection table if the desired accuracy of the measurement is known, find the desired accuracy listed in the second column of the table. The required FOV is listed in the first column, and the lens standoff distance is listed in the right columns.

Lens and Camera Set-up
(continued)

Table 4.A *(continued on next page)*
Lens Selection Table

Field of View* in inches (millimeters)		Desired Accuracy in inches (millimeters)		Pixel Size in inches (millimeters)		12.5mm Lens 2801-NL2 in inches (millimeters)		25mm Lens 2801-NL1 in inches (millimeters)		50mm Lens 2801-NL6 in inches (millimeters)		55mm 2801-NL3	60mm Lens 2801-NL9 in inches (millimeters)		105mm 2801-NL4 in inches (millimeters)	
Height	Width	Height	Width	Height	Width	Dist. to Obj.	Exten-sion	Dist. to Obj.	Exten-sion	Dist. to Obj.	Exten-sion	Dist. to Obj.	Dist. to Obj.	Exten-sion	Dist. to Obj.	Exten-sion
0.3 (8)	0.4 (10)	0.0039 (0.1)	0.0048 (0.12)	0.0013 (0.03)	0.0016 (0.04)	-	-	-	-	-	-	-	5.1 (129)	None	8.9 (226)	40mm
0.4 (10)	0.5 (13)	0.0051 (0.13)	0.0063 (0.16)	0.0017 (0.04)	0.0021 (0.05)	-	-	-	-	-	-	5.5 (139)	6.0 (152)	None	10.5 (267)	10mm
0.5 (13)	0.7 (17)	0.0063 (0.16)	0.0078 (0.2)	0.0021 (0.05)	0.0026 (0.07)	-	-	-	-	-	-	6.3 (161)	6.9 (175)	None	12.1 (307)	None
0.6 (15)	0.8 (20)	0.0078 (0.2)	0.0093 (0.24)	0.0026 (0.07)	0.0031 (0.08)	-	-	-	-	-	-	7.2 (182)	7.8 (198)	None	13.7 (347)	None
0.8 (20)	1.1 (27)	0.010 (0.25)	0.012 (0.3)	0.0034 (0.09)	0.0041 (0.1)	-	-	-	-	-	-	8.8 (224)	9.6 (244)	None	16.9 (428)	None
1.0 (25)	1.3 (34)	0.013 (0.33)	0.016 (0.41)	0.0043 (0.11)	0.0052 (0.13)	2.4 (61)	2mm	4.8 (121)	5mm	9.5 (241)	10mm	10.5 (266)	11.4 (291)	None	20.0 (509)	None
1.2 (30)	1.6 (41)	0.016 (0.41)	0.019 (0.48)	0.0052 (0.13)	0.0063 (0.16)	2.8 (70)	2mm	5.5 (140)	5mm	11.1 (280)	10mm	12.2 (309)	13.3 (337)	None	23.2 (590)	None
1.5 (38)	2.0 (51)	0.019 (0.48)	0.023 (0.58)	0.0064 (0.16)	0.0078 (0.2)	3.3 (85)	2mm	6.7 (169)	2mm	13.3 (339)	5mm	14.7 (373)	16.0 (406)	None	28.0 (711)	None
1.8 (46)	2.4 (61)	0.023 (0.58)	0.028 (0.71)	0.0077 (0.2)	0.0094 (0.24)	3.9 (100)	1mm	7.8 (198)	2mm	15.6 (396)	5mm	17.2 (436)	18.7 (476)	None	32.8 (832)	None
2.1 (53)	2.8 (71)	0.027 (0.69)	0.033 (0.84)	0.0090 (0.23)	0.011 (0.28)	4.5 (114)	1mm	8.9 (227)	1mm	17.9 (452)	5mm	19.7 (500)	21.4 (545)	None	37.5 (954)	None
2.4 (61)	3.2 (81)	0.031 (0.79)	0.038 (0.97)	0.010 (0.25)	0.012 (0.3)	5.0 (128)	1mm	10.1 (256)	1mm	20.1 (512)	2mm	22.2 (563)	24.2 (614)	None	42.3 (1075)	None
2.7 (69)	3.6 (91)	0.035 (0.89)	0.042 (1.07)	0.012 (0.3)	0.014 (0.36)	5.6 (142)	0.5mm	11.2 (285)	1mm	22.4 (569)	2mm	24.7 (627)	26.9 (684)	None	47.1 (1196)	None
3.0 (76)	4.0 (102)	0.039 (0.99)	0.047 (1.19)	0.013 (0.33)	0.016 (0.41)	6.2 (157)	0.5mm	12.3 (314)	1mm	24.7 (627)	2mm	27.2 (690)	29.6 (753)	None	51.9 (1317)	None
3.5 (89)	4.7 (119)	0.045 (1.14)	0.055 (1.4)	0.015 (0.38)	0.018 (0.46)	7.1 (181)	0.5mm	14.2 (362)	None	28.5 (724)	None	31.3 (796)	34.2 (869)	None	59.9 (1520)	None

* If you are using a Catalog No. 2801-YD camera, multiply the field of view by 0.75.
- A hyphen in a column indicates that the standoff distance is not possible with the specified lens.

Lens and Camera Set-up (continued)

Table 4.A *(continued)*
Lens Selection Table

Field of View* in inches (millimeters)		Desired Accuracy in inches (millimeters)		Pixel Size in inches (millimeters)		12.5mm Lens 2801-NL2 in inches (millimeters)		25mm Lens 2801-NL1 in inches (millimeters)		50mm Lens 2801-NL6 in inches (millimeters)		55mm 2801-NL3	60mm Lens 2801-NL9 in inches (millimeters)		105mm 2801-NL4 in inches (millimeters)	
Height	Width	Height	Width	Height	Width	Dist. to Obj.	Extension	Dist. to Obj.	Extension	Dist. to Obj.	Extension	Dist. to Obj.	Dist. to Obj.	Extension	Dist. to Obj.	Extension
4.0 (102)	5.3 (135)	0.052 (1.32)	0.062 (1.57)	0.017 (0.43)	0.021 (0.53)	8.1 (205)	0.5mm	16.1 (410)	None	32.2 (819)	None	35.5 (901)	38.7 (983)	None	67.7 (1720)	None
4.5 (114)	6.0 (152)	0.058 (1.37)	0.070 (1.78)	0.019 (0.48)	0.023 (0.58)	9.0 (229)	0.5mm	18.0 (458)	None	36.1 (916)	None	39.7 (1008)	43.3 (1099)	None	75.7 (1923)	None
5.0 (127)	6.7 (169)	0.064 (1.63)	0.079 (2.01)	0.022 (0.56)	0.026 (0.66)	10.0 (253)	0.5mm	19.9 (506)	None	39.9 (1013)	None	43.8 (1114)	47.8 (1215)	None	83.7 (2126)	None
5.5 (140)	7.3 (186)	0.071 (1.80)	0.086 (2.18)	0.024 (0.61)	0.029 (0.74)	10.9 (277)	None	21.8 (554)	None	43.6 (1108)	None	48.0 (1219)	52.3 (1329)	None	91.6 (2326)	None
6.0 (152)	8.0 (203)	0.08 (2.0)	0.09 (2.4)	0.026 (0.66)	0.031 (0.79)	11.9 (273)	None	23.7 (602)	None	47.4 (1204)	None	52.2 (1325)	56.9 (1446)	None	99.6 (2530)	None
7.0 (178)	9.3 (237)	0.09 (2.3)	0.11 (2.8)	0.030 (0.76)	0.036 (0.91)	13.7 (349)	None	27.5 (698)	None	55.0 (1396)	None	60.5 (1536)	66 (1676)	None	115.5 (2934)	None
8.0 (203)	10.7 (271)	0.10 (2.6)	0.12 (3.1)	0.034 (0.86)	0.041 (1.04)	15.6 (398)	None	31.3 (795)	None	62.6 (1590)	None	68.9 (1749)	75.1 (1908)	None	131.4 (3338)	None
9.0 (229)	12 (305)	0.12 (2.9)	0.14 (3.6)	0.039 (0.99)	0.047 (1.19)	17.5 (446)	None	35.1 (891)	None	70.1 (1782)	None	77.2 (1960)	84.2 (2138)	None	147.3 (3742)	None
10 (254)	13 (339)	0.13 (3.3)	0.15 (3.9)	0.043 (1.09)	0.051 (1.30)	19.4 (494)	None	38.8 (986)	None	77.7 (1974)	None	85.5 (2171)	93.2 (2369)	None	163.2 (4146)	None
12 (305)	16 (406)	0.16 (3.9)	0.19 (4.8)	0.052 (1.32)	0.062 (1.57)	23.2 (590)	None	46.4 (1180)	None	92.9 (2359)	None	102.2 (2595)	111.4 (2831)	None	195.0 (4954)	None
15 (381)	20 (508)	0.19 (4.9)	0.23 (5.9)	0.064 (1.63)	0.078 (1.98)	28.9 (734)	None	57.8 (1468)	None	115.6 (2936)	None	127.2 (3230)	138.7 (3524)	None	242.8 (6166)	None
18 (457)	24 (610)	0.23 (5.9)	0.28 (7.1)	0.077 (1.96)	0.094 (2.39)	34.6 (878)	None	69.2 (1757)	None	138.3 (3514)	None	152.2 (3865)	166.0 (4216)	None	290.5 (7379)	None
21 (533)	28 (711)	0.27 (6.9)	0.33 (8.3)	0.090 (2.29)	0.109 (2.77)	40.3 (1023)	None	80.5 (2045)	None	161.1 (4091)	None	177.2 (4500)	193.3 (4909)	None	338.2 (8591)	None
24 (610)	32 (813)	0.33 (8.4)	0.37 (9.5)	0.103 (2.62)	0.125 (3.18)	45.9 (1167)	None	91.9 (2334)	None	183.8 (4668)	None	202.2 (5135)	220.5 (5602)	None	385.9 (9803)	None

* If you are using a Catalog No. 2801-YD camera, multiply the field of view by 0.75.
- A hyphen in a column indicates that the standoff distance is not possible with the specified lens.

Lens and Camera Set-up

When mounting your camera and lens assembly, set the front of the lens of the camera at the standoff distance listed in the table for your field of view. This distance is an approximation; it will get your camera close to the ideal location. From there, focus the lens and set the F-stop for maximum clarity and brightness.

Example One
The desired FOV is 3 by 4 inches (76×102 mm). Referring to Table 4.A, you can use a variety of lenses, depending on the required standoff distance. For a standoff distance of 27.2 inches (690 mm), for example, you would use the 55mm lens.

Example Two
The desired accuracy of measurement is 1/16 of an inch, or 0.063 inch (1.6 mm). The closest value in the table is 0.052 × 0.070 (1.37 × 1.78 mm) with an FOV of 4.5 × 6.0 inches (114 × 152 mm). Referring to Table 4.A, for a standoff distance of 18 inches (458 mm), for example, you would use the 25mm lens.

Workpiece Positioning

The presentation of the workpiece within the VIM2 module FOV can fall into into one of two categories:

1) The workpiece is not moving (still) during inspection.

2) The workpiece is moving during inspection.

If the workpiece is still during inspection, the staging of an inspection application is generally simpler than if it were moving. Both categories of positioning are described below.

Still Workpiece

When the workpiece is still, the optimum set-up would be to position the workpiece in front of the camera, with a tolerance of less than 1% of the FOV. For example, in a FOV of 3 × 4 inches (76×102 mm), the tolerance would be ± 1/32 inch (0.79 mm). If the object cannot be positioned accurately, the VIM2 tools (X and Y float gages) can be used to compensate for position variations. It is desirable to fixture the workpiece so that the variation in workpiece location will be less than 25% of the FOV. (Note that the use of these float gages tools slightly increases the time required to inspect each workpiece).

Moving Workpiece

If the workpiece is moving, a strobe light is probably needed to "freeze" the workpiece's motion in order to eliminate blur in the image. A strobe light is positioned in the same manner as a fluorescent or incandescent light. The VIM2 module's *Strobe* output, located on the VIM2 swingarm, must be used to trigger the strobe light at the correct time.

In order for the strobe light to eliminate blur in the image, it must be much brighter than the ambient lighting. A cover or shroud may be needed around the workstage to reduce the ambient light. The shroud can also prevent the strobe light flashes from becoming a distraction to nearby workers.

To determine whether a strobe light is needed, first calculate the FOV necessary for the inspection. Then find the size of a pixel for that FOV using table 4.A or 4.B. Divide the speed of the workpiece in inches (or mm) per second by 60 seconds. If the result is greater than one-half the pixel size, a strobe light should be used.

- **Example**

Suppose the workpiece is a ring with a diameter of 2.5 inches (63.5 mm) moving at a speed of 6 inches (152.4 mm) per second. You select an FOV of 3 × 4 inches (76.2 × 101.6 mm). From table 4.A or 4.B, the pixel size is 0.013 × 0.016 inches (.330 × .406 mm).

Divide the speed of the workpiece, 6 inches (152.4 mm) per second, by 60 seconds:

6 in./sec. ÷ 60 sec. = 0.1 inch, or
152.4 mm/sec. ÷ 60 sec. = 2.54 mm

The result is greater than one-half the pixel size; therefore, a strobe light is needed. The workpiece speed would have to be less than 0.5 inch (12.7 mm) per second to eliminate the need for a strobe light in this case.

Using Filters With the VIM2 Module

Filters are devices used to suppress interference which would appear as noise in an image. Lens and lighting filters may be used in a VIM2 module application situation.

Lens Filters

A colored lens filter can be useful in an application situation that requires a certain colored item to stand out. If you are inspecting shiny, transparent, or translucent workpieces, polarizing filters can be valuable. Neutral density filters can be used to restrict the focus or depth of field of a lens setup. Use an infrared pass or cut filter if your workpiece is heated or if you want to exclude ambient visible light.

Photographers have used lens filters for more than a century. The techniques they have developed are all applicable to machine vision. It is beyond the scope of this manual to discuss the use of lens filtering in detail.

Lighting Filters

Lighting filters can also be used, or a combination of lighting and lens filters can be used. For example, you can light a fluorescent workpiece with ultraviolet light and observe the visible light it gives off. You can also illuminate the workpiece with horizontally polarized light and view through a vertical polarizing filter.

Workstage Shielding

To control contrast and shadows you need to enclose (shroud) the workstage. This prevents stray reflections and shadows from interfering with the automatic operation of your system. Shrouding also provides a dark black background which can increase the contrast.

It is also good practice to have any fixtures close to the workstage finished in a flat black color. This prevents reflections and the resulting uneven lighting. This is the reason that camera lenses are black.

(This page intentionally left blank)

Chapter 5 Connection and Power-up

Chapter Objectives

This chapter provides guidelines for installation and connection of the VIM2 module and peripheral system components, describes power-up indications and checks, and describes the VIM2 LED indicators and swingarm I/O.

VIM2 System Components

The procedures in this chapter relate only to those components that are part of the VIM2 machine vision system, namely:

- VIM2 module, Cat. No. 2803-VIM2.
- 1771 I/O rack:
  - PLC 1771 I/O rack
  - 1771-PSC rack (standalone)
- Chassis power supply (any of a number of suitable power supplies)
- External 12.5 VDC power supply (one of several suitable power supplies, depending on line input voltage)
- Monochrome video monitor:
  - 9-inch, 115 VAC, Cat. No. 2801-N9.
  - 9-inch, 230 VAC, Cat. No. 2801-N20.
  - 12-inch, 115 VAC, Cat. No. 2801-N6.
- Light pen, Cat. No. 2801-N7
- Camera:
  - Cat. No. 2801-YC
  - Cat. No. 2801-YD
- Interconnecting wires and cables

Additional components may be required for some installations. These will be identified where appropriate in this chapter.

Connecting the VIM2 System Components

Before you install your VIM2 system at its factory-floor site, you may find it useful to connect the basic components temporarily on a workbench or tabletop, where you can perform a connection and power-up check. There you can also get acquainted with the VIM2 user interface and the functional features, and review the set-up techniques presented in this manual.

Requirements for Installation of the VIM2 Module

This section discusses requirements for installing the VIM2 module into a 1771 I/O rack, or a standalone rack, such as the Allen-Bradley Cat. No. 1771-PSC.

Installation Into an Installed 1771 I/O Rack

Installation of a VIM2 module into an installed 1771 I/O rack is dependent upon two requirements: availability of slot space and availability of sufficient power.

Availability of Space
The VIM2 module requires a module group (two adjacent vacant slots) in which to be mounted. If your existing 1771 I/O rack does not have two adjacent empty slots, you have two options:

1) acquire a larger rack;
2) acquire an additional rack.

See your local Allen-Bradley representative for details.

Availability of Sufficient Power
If your existing I/O rack has the required space, next determine whether the existing power supply has enough current to satisfy the VIM2 module. The VIM2 module requires 3 amps of current (maximum) in order to operate.

To determine if your power source has sufficient current, subtract the total amount of current consumed by each individual component in the rack (see individual unit documentation), other than the VIM2 module, from the total amount of current output from your power source (see power supply documentation). If the resultant number is greater than or equal to 3 amps, sufficient current is available.

Total Current Output - Total Current Consumed = Available Current

If your existing 1771 I/O rack does not have enough power, you have two options:

1) employ a larger power supply;
2) acquire an additional rack with power supply.

See your local Allen-Bradley representative for details.

If your I/O rack meets power and space requirements, you can install your VIM2 module (see "VIM2 Module Installation" in this section).

Requirements for Installation Into a 1771 Standalone I/O Rack

A standalone I/O rack (such as catalog number 1771-PSC) has four slots, two for the VIM2 module and two for an in-rack power supply (catalog number 1771-P3). Since the standalone unit is self-contained, there will always be enough space and current. For additional rack installation information refer to the documentation accompanying your standalone rack.

I/O Rack Installation

For information on I/O rack installation, see documents 1771-4.5, "Mounting Instructions for 1771 I/O Chassis and Power Supply," and 1770-4.1 "Grounding and Wiring Guidelines." Read these documents before installing rack.

CAUTION: Ground loops can seriously impair the performance of the VIM2 module. Ensure that proper grounding procedures are followed. Refer to Cat. No. 1770-4.1, "Grounding and Wiring Guidelines" for correct grounding procedures.

Power Supply Installation

Many PLC system power supplies are available for use by the VIM2 module. For information on the installation of specific power supplies, refer to the documentation accompanying that power supply as well as Cat. No. 1770-4.1, "Grounding and Wiring Guidelines."

VIM2 Module Installation

This section provides guidelines for installing the VIM2 module.

WARNING: Remove system power before attempting installation of VIM2 module. Failure to do so may result in electrical shock.

Use of Keying Bands

Keying bands are shipped with each I/O chassis. Use of keying bands is recommended. Use of keying bands helps to prevent installation of a module in the wrong slot. Keying bands can be installed and replaced with needle-nose pliers.

VIM2 Module Installation

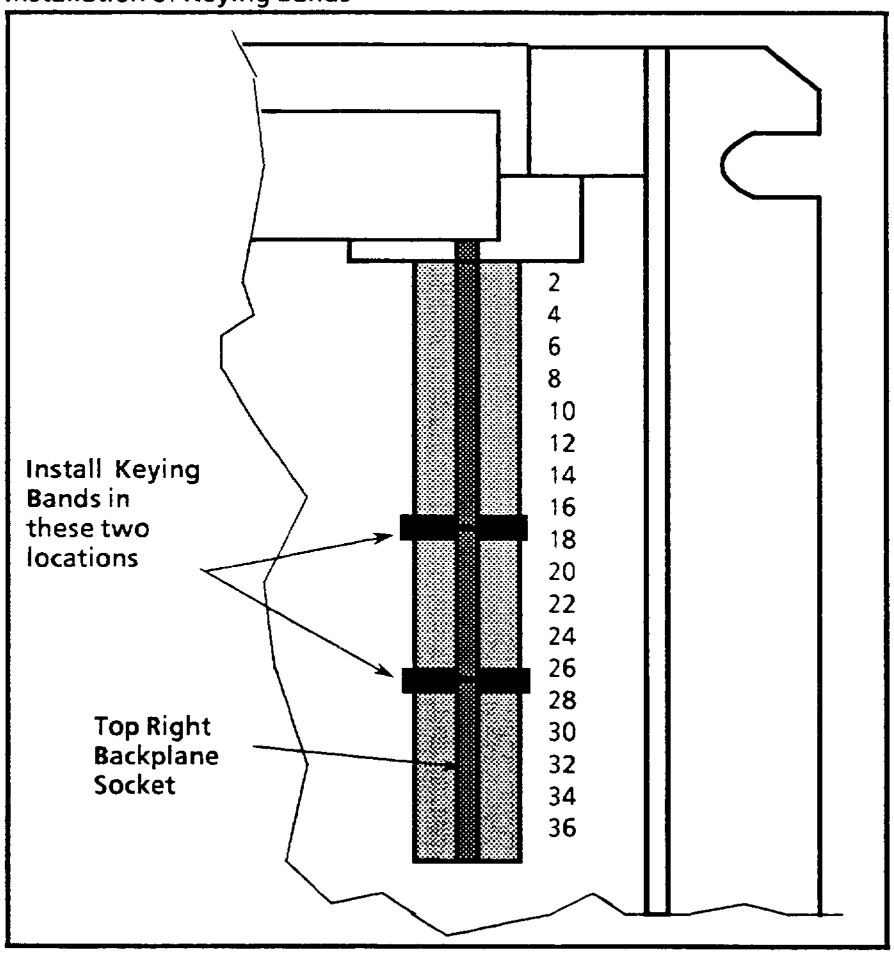
(continued)

Install keying bands before installing the VIM2 module, as follows:

1. Each VIM2 module plugs into two addressing slots on the backplane (two per slot) – determine which two slots the VIM2 module will occupy.

2. Install two keying bands on the top right socket of the two slots used by the VIM2 module. The socket has guide numbers along the right side to aid in the positioning of the keying bands. Install keying bands between numbers 16 - 18 and 26 - 28 for the VIM2 module (see Figure 5.1).

Figure 5.1
Installation of Keying Bands



VIM2 Module Installation
(continued)

Setting the I/O Chassis Configuration Plug

Many 1771 I/O chassis contain a "configuration plug." This is a stake-pin jumper located on the chassis backplane near the PLC controller slot. Set the configuration plug to the Y (left) position – covering the middle and left prongs – or to the N (right) position – covering the middle and right prongs – depending on the presence of a PLC and power supplies, as follows (refer to Figure 5.2):

Set to Y when either –

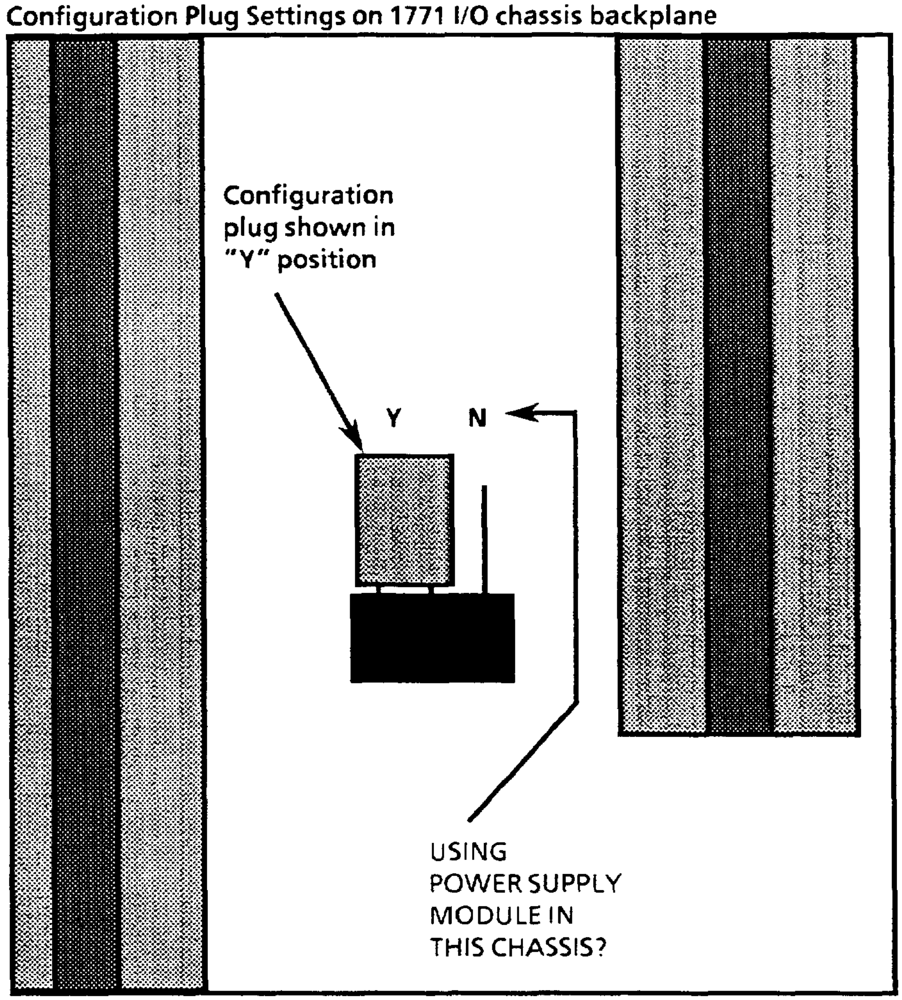
- You are using an in-rack power supply module, or

- You are using the VIM2 module(s) in the chassis with no PLC processor present.

Set to N when –

- You are using both an external power supply and a PLC controller.

Figure 5.2
Configuration Plug Settings on 1771 I/O chassis backplane



VIM2 Module Installation (continued)

VIM2 Module Installation

To install the VIM2 module into the 1771 chassis (see Figure 5.3):

1. Lift open the module locking latch at the top of the chassis slot.

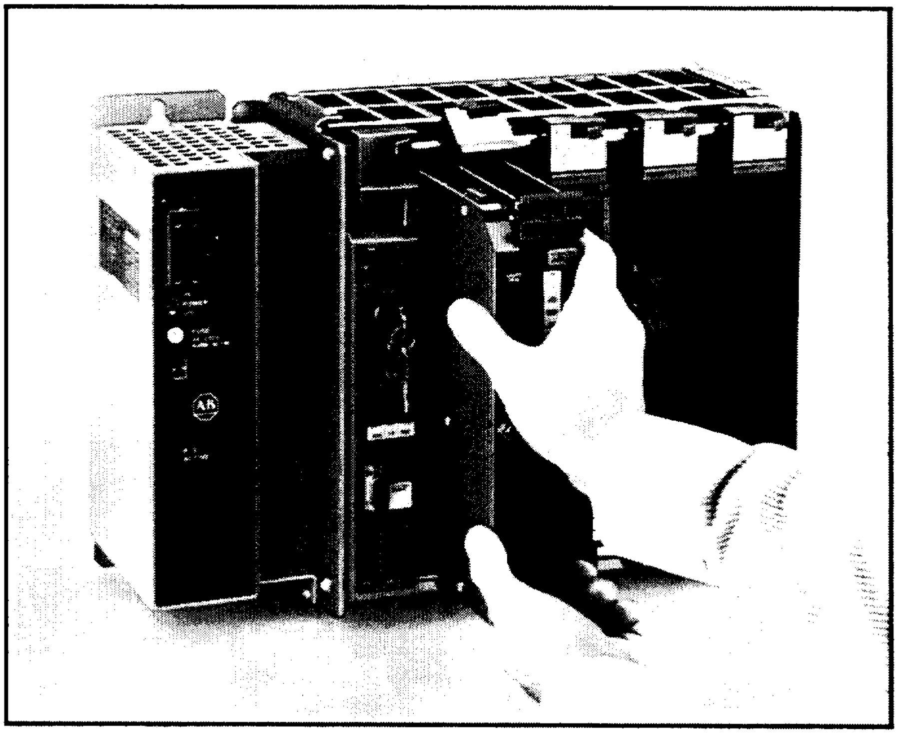
2. Insert the module into the slots, sliding the module along the plastic guides on the top and bottom of each slot.

   Note: Do not force the module into its backplane socket. Apply firm and even pressure to seat it into its sockets.

3. After the VIM2 module is seated in place, close the module locking latch.

At this point the VIM2 module has been installed. Next you will connect the video camera, video monitor, camera power supply (to the swingarm), and swingarm.

Figure 5.3
Installation of the VIM2 Module



87-283-1

Camera Component Installation

This section describes installation procedures for the video camera, cable, lens, and extension tube.

Lenses

You have the option of using any of several lenses depending upon your application needs. See Chapter 4 for information regarding lens characteristics and specifications; also refer to the Product Data sheet, Publication No. 2801-2.6.

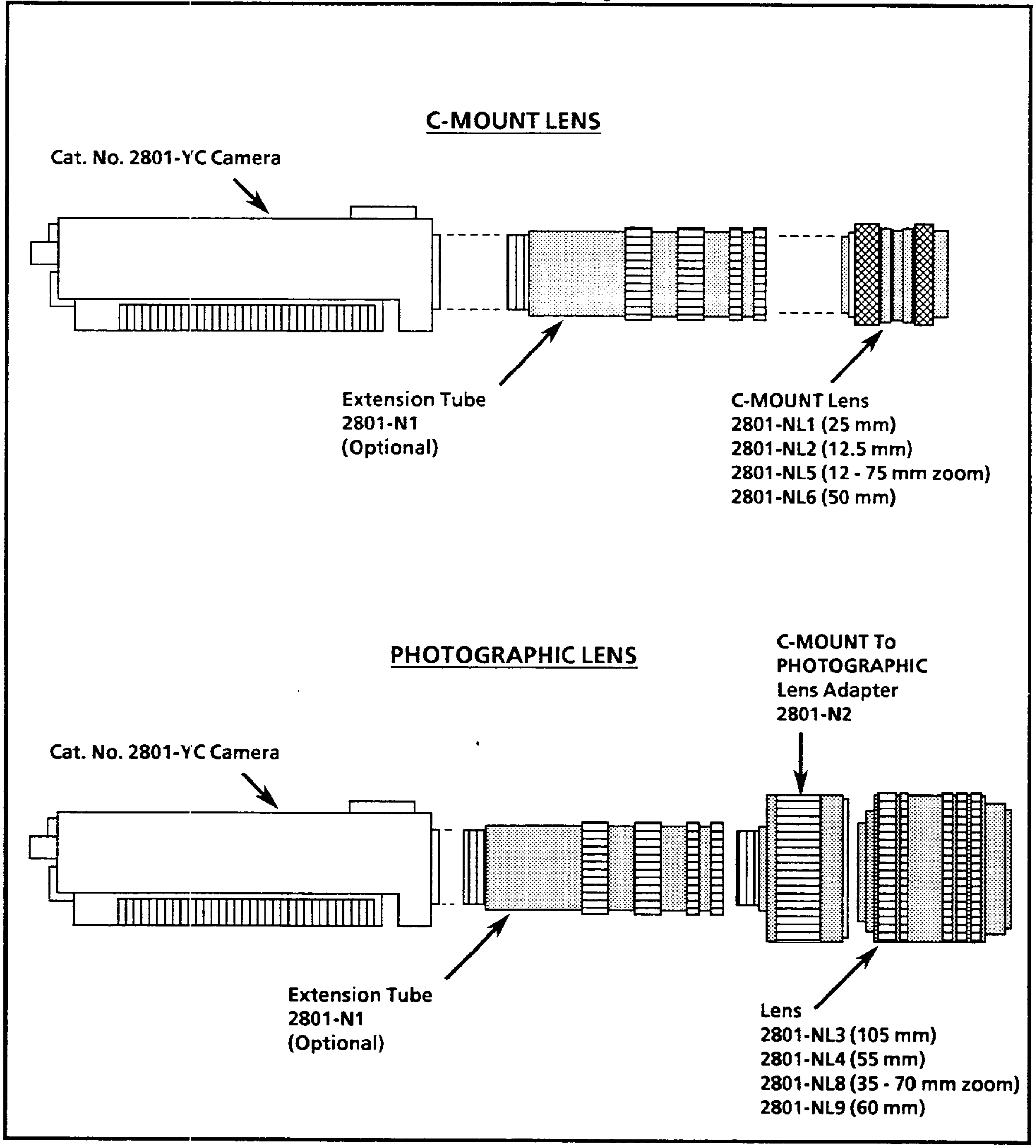
See Figure 5.4 for an illustration of camera and lens configurations. All lenses include an installation and maintenance instruction booklet. Consult your local Allen-Bradley representative for additional information.

Camera Extension Tubes

The optional camera extension tubes (see Figure 5.4) are used to alter the image focal length. This allows you to use the lenses at shorter distances. See Chapter 4 for lens extension information. Installation instructions are provided with each unit.

Camera Component Installation
(continued)

Figure 5.4
Typical Camera Configurations



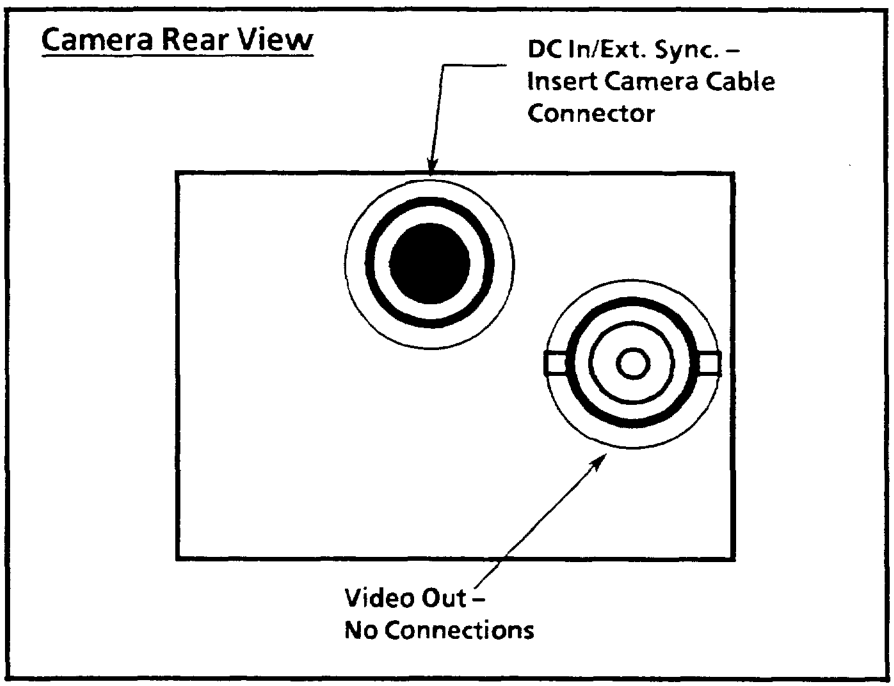
Camera Component Installation (continued)

Connecting the Camera to the VIM2 Module

Connect the camera to the VIM2 module through the 12-pin camera connector cable. Connector cables are offered in various lengths; refer to the Product Data sheet, publication no. 2801-2.4, for connector cable information. To connect the camera cable:

1. Plug the male end of the cable into the *CAMERA* input jack on the face of the VIM2 module (see Figure 5.6).
2. Plug the other end of the cable into the 12-pin output jack on the back of the camera (see Figure 5.5) which is labeled *VISION SYSTEM*. If resistance is encountered realign the pins and try again.
3. To remove the cable from its connection, slide the collar of the connector back towards you while pulling the cable out of its jack.

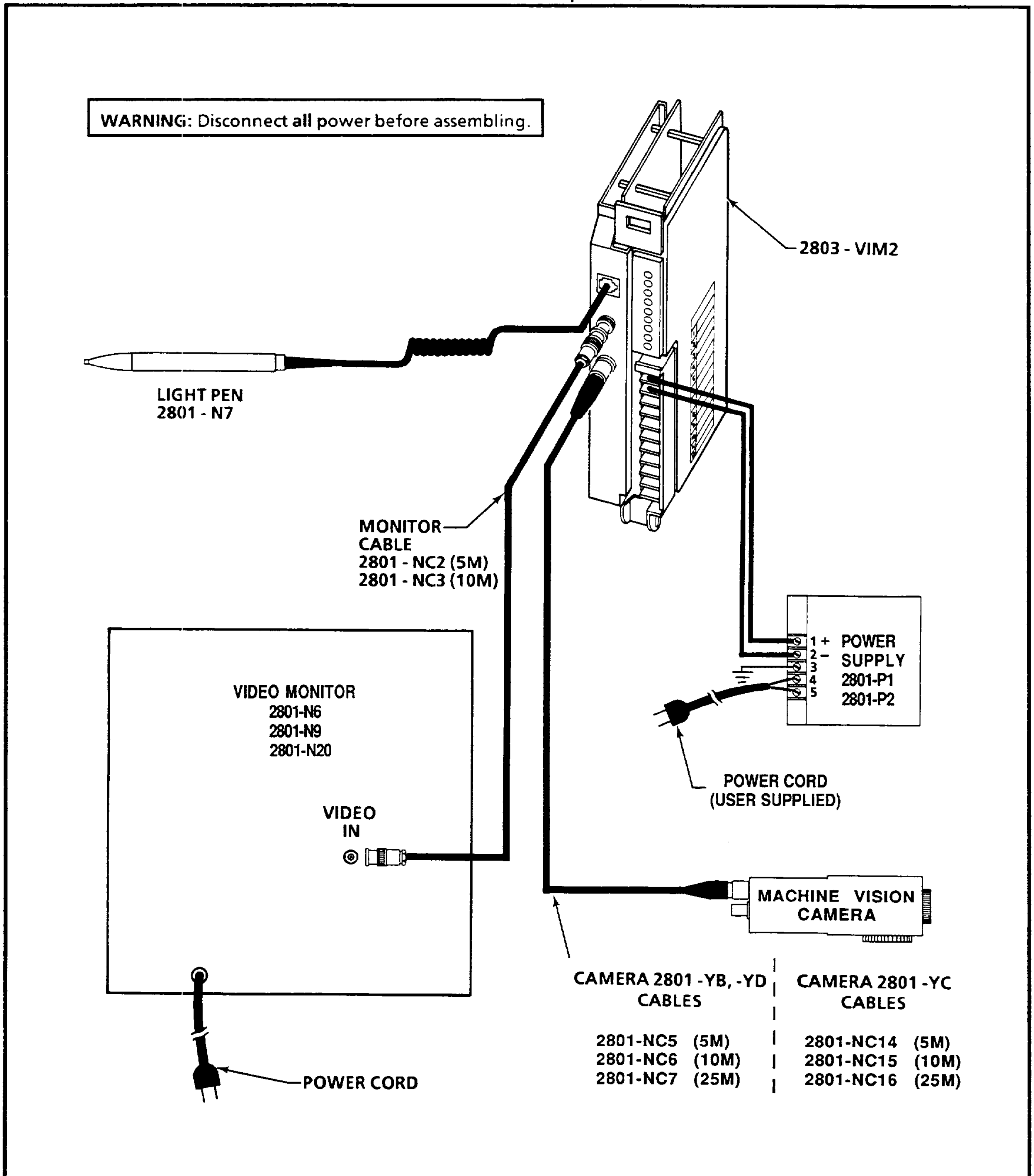
Figure 5.5
Camera I/O Locations



Note: In order to operate, the video camera requires an external 12.5 VDC power source, which must be connected to the VIM2 module swingarm. See the section covering connection of the camera power supply under "Swingarm Installation," this chapter.

Camera Component Installation
(continued)

Figure 5.6
The VIM2 Module, Peripherals, and cables



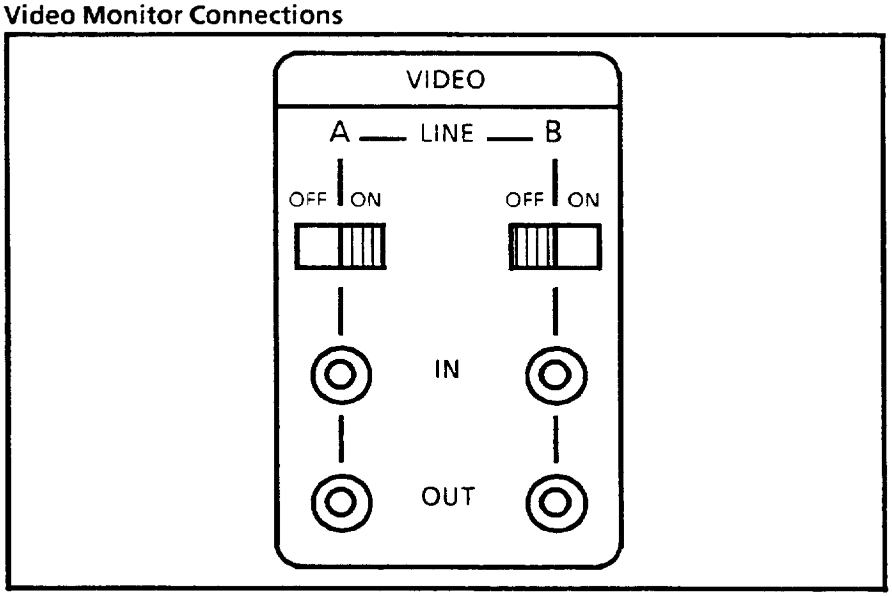
Video Monitor Connection

Connect the monochrome video monitor to the VIM2 module with a BNC type coaxial cable (see the Product Data sheet, Publication No. 2801-2.4 for more information on cables).

Note: Remove the outer plastic screen shield, if any, that may have come with your video monitor. The shield may adversely affect light pen operation.

1. Connect one end of the cable to the *MONITOR* connector on the face of the VIM2 module by aligning the slots on the connector with the cylindrical keys on the *MONITOR* input jack. Twist clockwise to lock.
2. Connect the other end of the cable to the Line A "IN" jack located on the rear panel of the monitor (see Figure 5.7).
3. Set the *LINE A* switch on the monitor back panel to "ON."
4. Set the monitor *LINE* button (on the front panel) to A.
5. Set the monitor *SCAN* button to underscan. This allows you to see all the way to the edges of the image (**Note:** Adjust the monitor brightness and contrast later as necessary, when viewing images on the monitor screen. Be sure to set the brightness to a level bright enough to allow the light pen to pick the icons on the screen).

Figure 5.7
Video Monitor Connections



Light Pen Installation

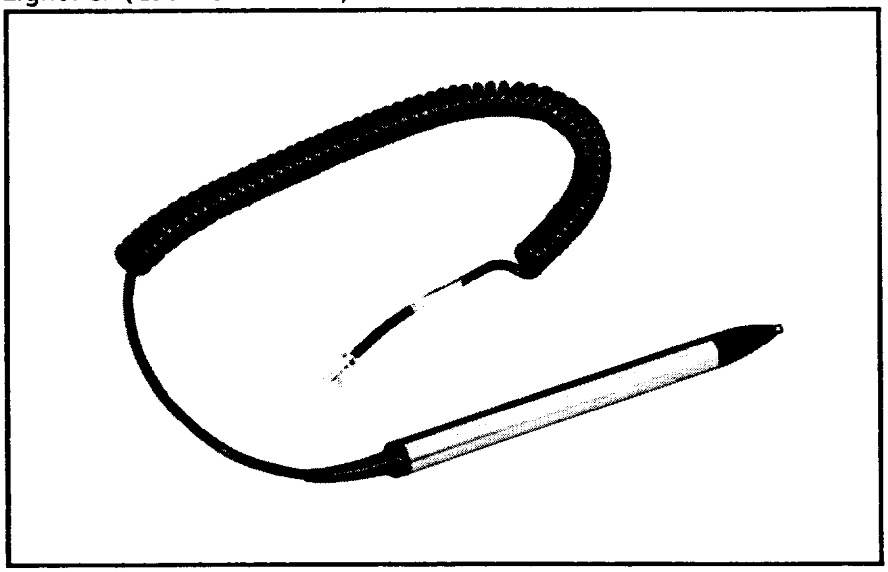
Attach the light pen (Cat. No. 2801-N7) to the VIM2 module see Figure 5.8:

1. Align the clear plastic plug on the end of the light pen cord with the *LIGHT PEN* jack located on the face of the VIM2 module (see Figure 5.6);

2. Insert the plug into the jack; if resistance is encountered, the release tab may not be properly seated in the slot.

3. To remove the light pen, press the release tab while pulling the plug out of the jack.

Figure 5.8
Light Pen (Cat. No. 2801-N7)



87-195-8

Swingarm

This section describes the swingarm I/O connection device. Swingarm connections are discussed as well as swingarm installation and removal procedures.

The swingarm (see Figure 5.9) can be disconnected and removed from the VIM2 module without disconnecting any connected wiring. This capability enables you to remove or replace the VIM2 module without having to disconnect and reconnect wiring from individual terminals.

Note: Use of shielded cables is highly recommended for connecting any and all I/O devices to the swingarm; shielded cables reduce susceptibility to electrical noise.

Swingarm Installation

To install the swingarm onto the VIM2 module:

WARNING: Remove system power before attempting installation, disconnection, or removal of the swingarm. Failure to do so may result in electrical shock.

1. Remove system power.
2. Snap the lower end of the swingarm (the C-shaped bracket) onto the horizontal bar of the I/O chassis in the area beneath the VIM2 Indicator LEDs (see Figure 5.9).
3. Pivot the swingarm upward and snap onto the front connector edge of the VIM2 module (see Figure 5.10).

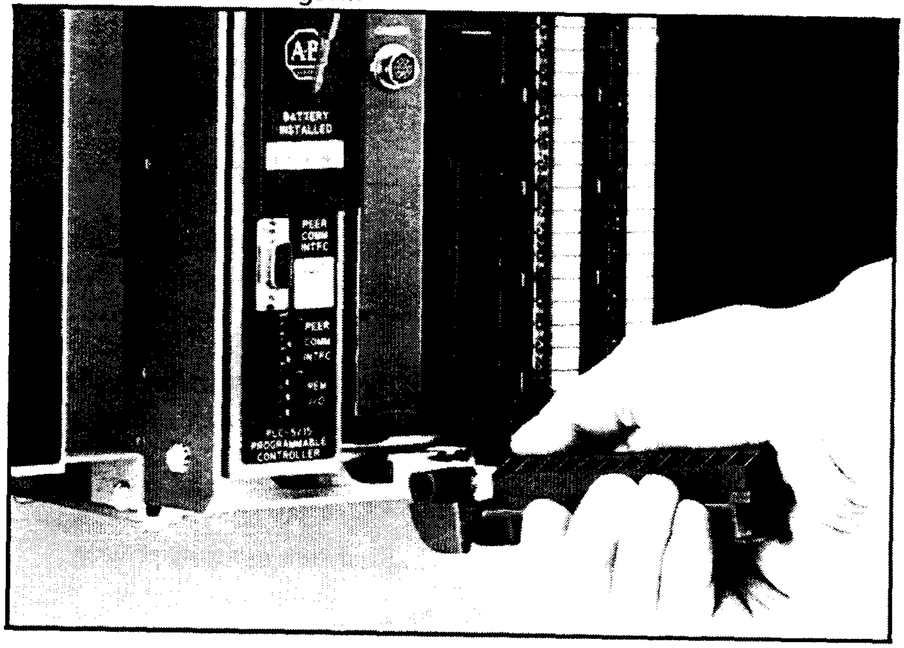
To disconnect and remove the swingarm:

1. Remove system power.
2. Lift the release tab located just beneath the VIM2 indicator LEDs.

   Note: The swingarm release tab requires a fair amount of pressure; be careful not to lift the tab too hard because damage may occur.
3. To disconnect the swingarm from the VIM2, pivot the swingarm downward and out of its seating in the VIM2.
4. To completely remove the swingarmfrom the chassis, carefully pull the swingarm from the chassis bar.

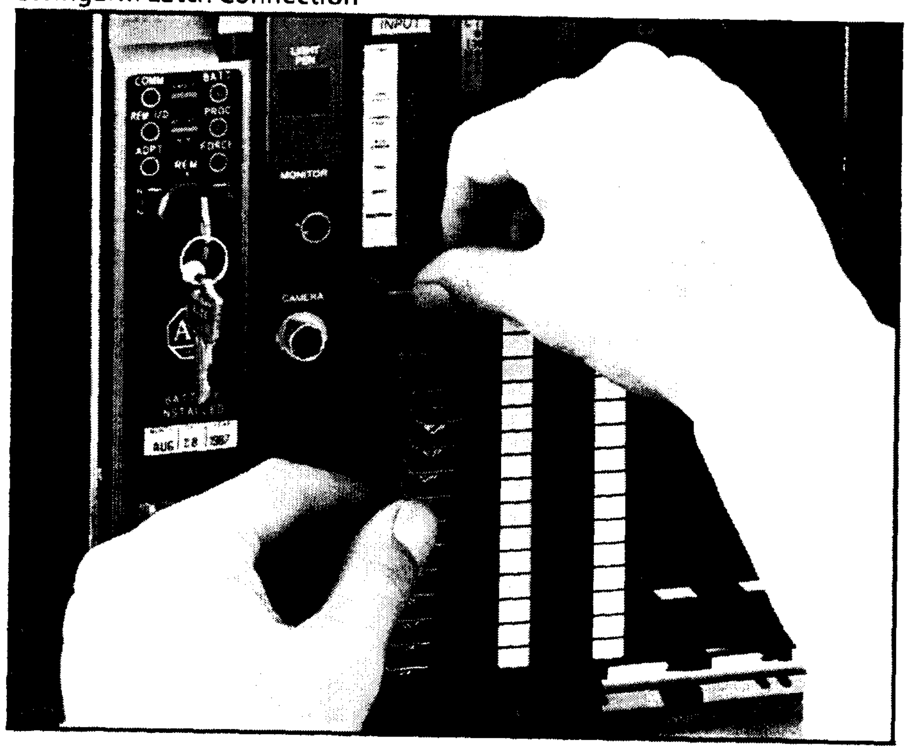
Swingarm Installation
(continued)

Figure 5.9
Installation of the Swingarm



87-283-2

Figure 5.10
Swingarm Latch Connection



87-283-5

Swingarm Connections

Swingarm terminal assignments are listed in Figure 5.11:

Figure 5.11
Swingarm - Field Wiring Terminals

Terminal Function		
1	12.5 VDC Camera Power Input	
2	Camera Power Common	
3	Trigger Input 3.3 – 32 VDC	
4	Trigger Input Common	
5	Strobe Output TTL	
6	Strobe Common	
7	Decision Output 3 – 32 VDC 1.0A	
8	Decision Output Common	
9	Busy Output 3 – 32 VDC 1.0A	
10	Busy Output Common	
11	CLL (Comm. Line Low)	THE DIFFERENTIAL VOLTAGE BETWEEN THESE TWO PINS MUST NOT EXCEED 10.5 VOLTS (MAX.)
12	CLH (Comm. Line High)	
	Use With 1771-WB Wiring Arm	

WARNING: Always disconnect the swingarm from the VIM2 module before connecting wires to or disconnecting wires from the swingarm. Remove system power before installation, disconnection, or removal of the swingarm. Failure to do so may result in electrical shock.

WARNING: Do not use the *Busy* output (terminals 9 and 10) to directly energize external equipment. The *Busy* output is energized during power-up and reset to indicate that the module is not yet ready to perform an inspection.

CAUTION: Take care not to miswire high voltage AC power directly to any swing arm terminal – connecting high voltage AC to any swingarm terminal can result in serious damage to the VIM2 module.

Swingarm Connections (continued)

The following describes each swingarm terminal in further detail:

Camera Power Supply Input - Terminals 1 and 2
In order to supply power for the camera used with the VIM2, an external 12.5 volt power is required. Connect the external power supply for the camera to terminals 1 and 2. The following power supplies can be used:

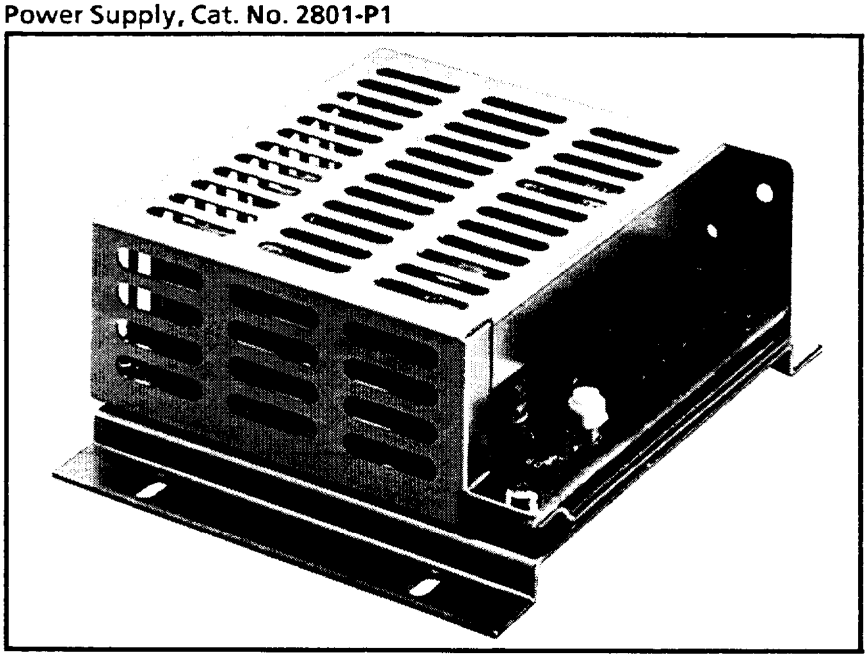
- The Cat. No. 2801-P1 power supply (see Figure 5.12). Input line voltage 85-132 VAC (47-63 Hz). Output is 12.5 VDC, 1.2 A nominal (enough power for four cameras).

- The Cat. No. 2801-P2 power supply. Input line voltage 170-264 VAC (47-63 Hz). Output is 12.5 VDC, 1.2 A nominal (enough power for four cameras).

Note: See Product Data, Publication No. 2801-2.3, for more detailed power supply specifications.

WARNING: Remove VIM2 system power and power to the external power supply before attempting installation. Failure to do so may result in electrical shock.

Figure 5.12
Power Supply, Cat. No. 2801-P1



87-282-5

Swingarm Connections
(continued)

Camera Power Supply Input – Terminals 1 and 2

To connect the Cat. No. 2801-P1 power supply to the VIM2 module, follow this procedure (see Figure 5.13):

1. Make sure the VIM2 module is not powered up.

2. Make sure the external power supply unit is disconnected from its power source.

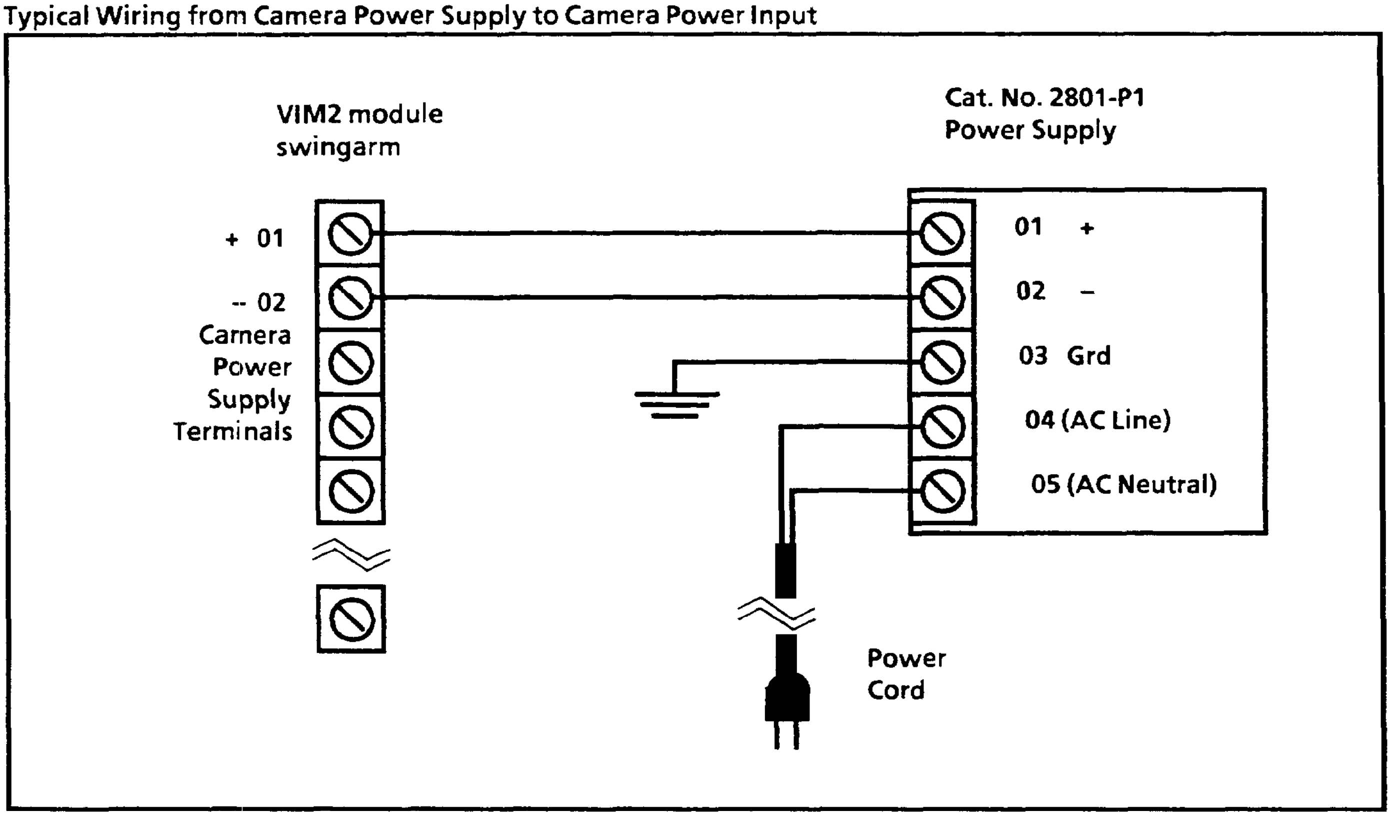
3. Disconnect the swing arm from the VIM2 module (see "Swingarm Installation" this chapter). If desired, remove the swingarm from the chassis bar completely.

4. Connect power supply terminals 3, 4, and 5 to power cord for line input VAC. Connect terminal 3 to ground. Connect terminal 4 to AC line. Connect terminal 5 to AC neutral.

5. Connect power supply terminal 1 to swingarm terminal 1. Connect power supply terminal 2 to swingarm terminal 2.

6. Reinstall the swingarm onto the VIM2 modules (see "Swingarm Installation" this chapter).

Figure 5.13
Typical Wiring from Camera Power Supply to Camera Power Input



Swingarm Connections (continued)

Trigger Input – Terminals 3 and 4

Connect external trigger input devices to terminals 3 and 4. The trigger signal must be held high for at least 50 microseconds for the trigger to be recognized.

The external trigger input signal must be from +3.3 VDC to 32 VDC (the internal resistance of the trigger input is equivalent to a 1kohm resistor).

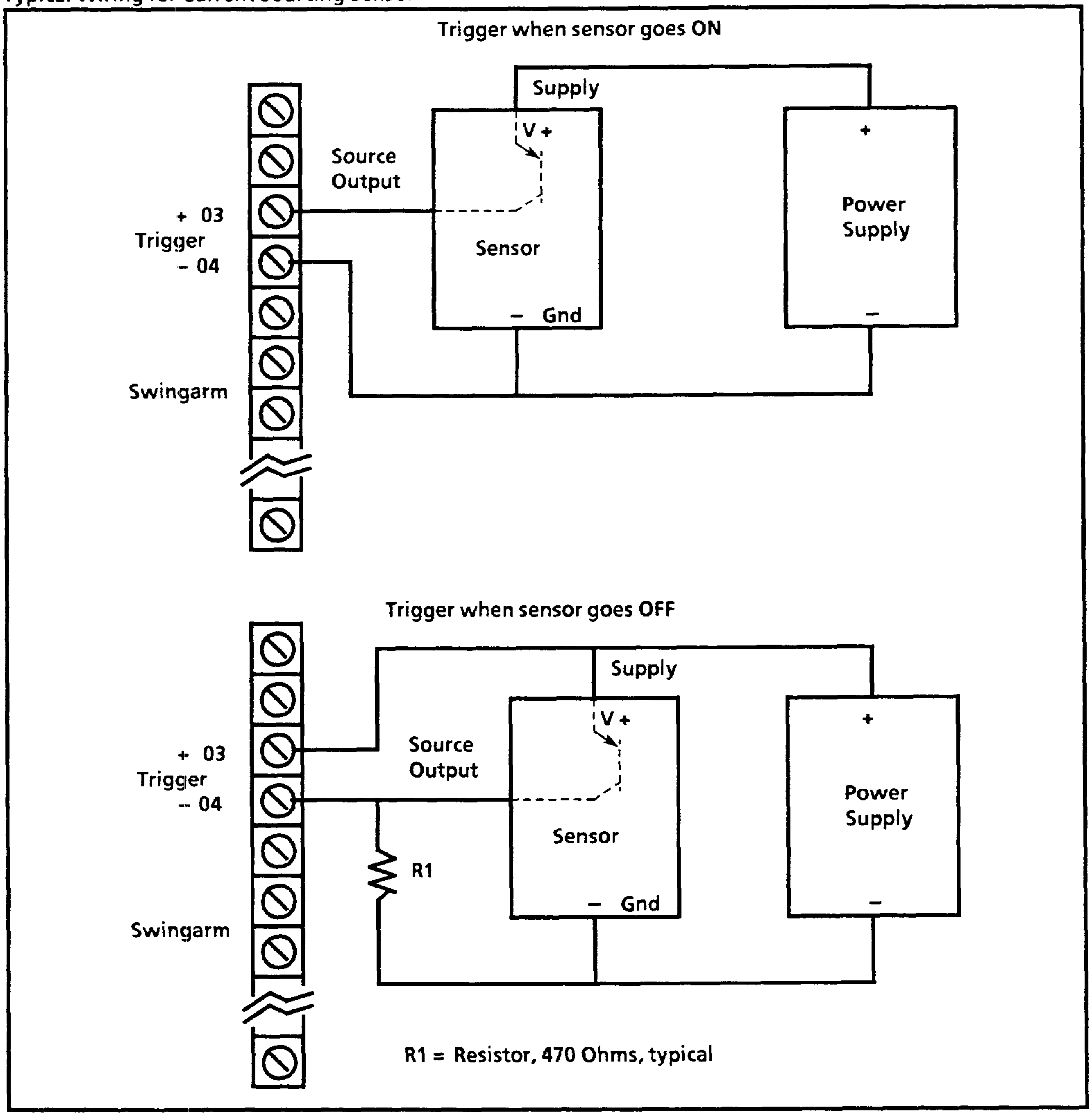
When the trigger input is high, the SWG TRIGGER LED goes *On* (provided the trigger input signal supplies at least 10 ma current; if the input current is less than 10 ma, the SWG TRIGGER LED does not go *On*).

Refer to Figures 5.14 and 5.15 for typical wiring for current source and current sink sensor input devices.

Note: By incorporating the camera power supply (terminals 1 and 2), you can use a pushbutton as a trigger device. See "Pushbutton Triggering" this chapter.

Swingarm Connections (continued) Trigger Input – Terminals 3 and 4

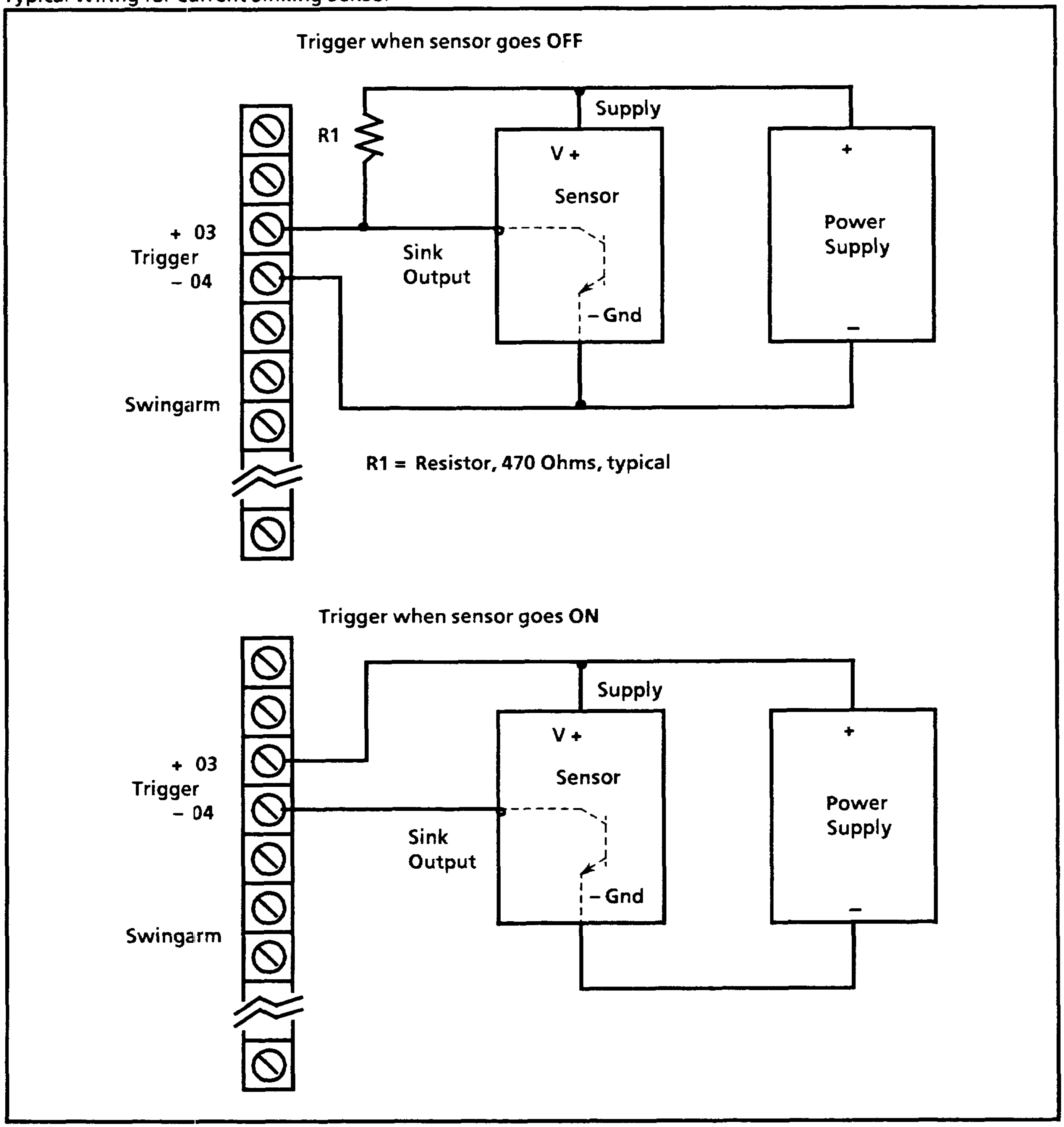
Figure 5.14
Typical Wiring for Current Sourcing Sensor



Swingarm Connections (continued)

Trigger Input – Terminals 3 and 4

Figure 5.15
Typical Wiring for Current Sinking Sensor



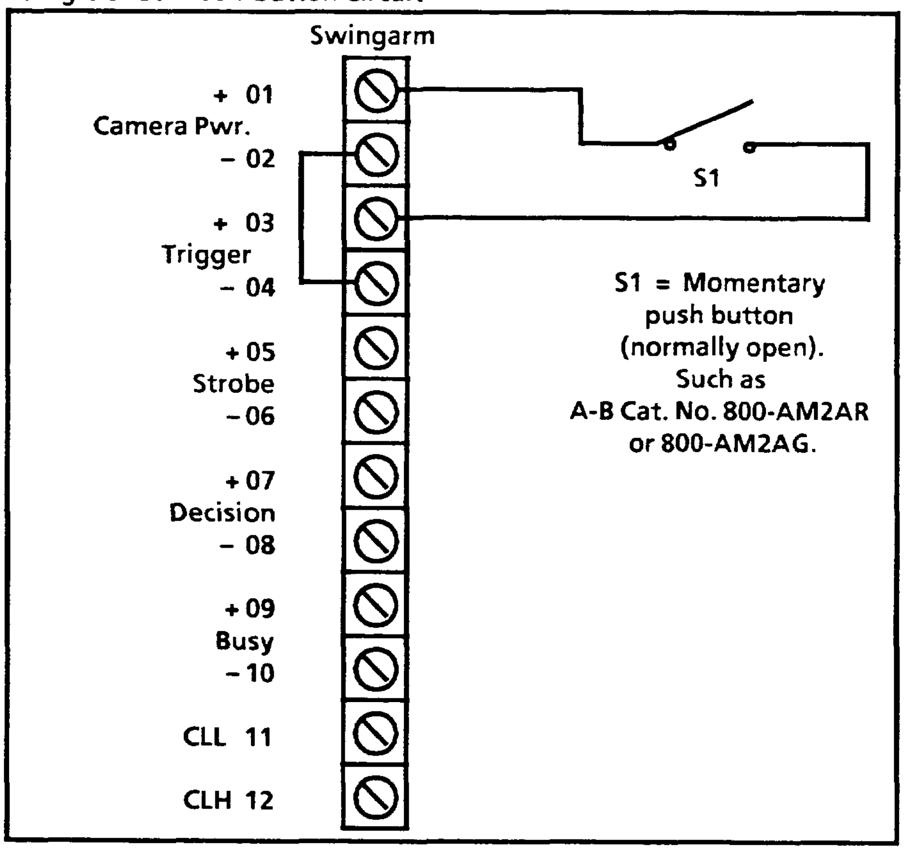
Swingarm Connections (continued)

Trigger Input – Terminals 3 and 4

Pushbutton Triggering: You may want to trigger the VIM2 module using a push button connected to the swingarm. This is useful for lab testing and system integration. Two examples of triggering circuits are provided in Figures 5.16 and 5.17. One is a "single shot" trigger which initiates a single inspection cycle and the other is a continuous trigger which repeats cycles as long as the button is pressed (**Note:** The module must be set to "SWG TRIG" mode – see Chapter 7 – to use the pushbutton trigger as shown).

"Single Shot" Push Button – This circuit diagram describes the connections for a push-button switch which you may attach to the swingarm to manually control triggering of the VIM2 module. The module must be set to "SWG TRIG" mode to use the pushbutton trigger as shown.

Figure 5.16
"Single Shot" Push-button Circuit

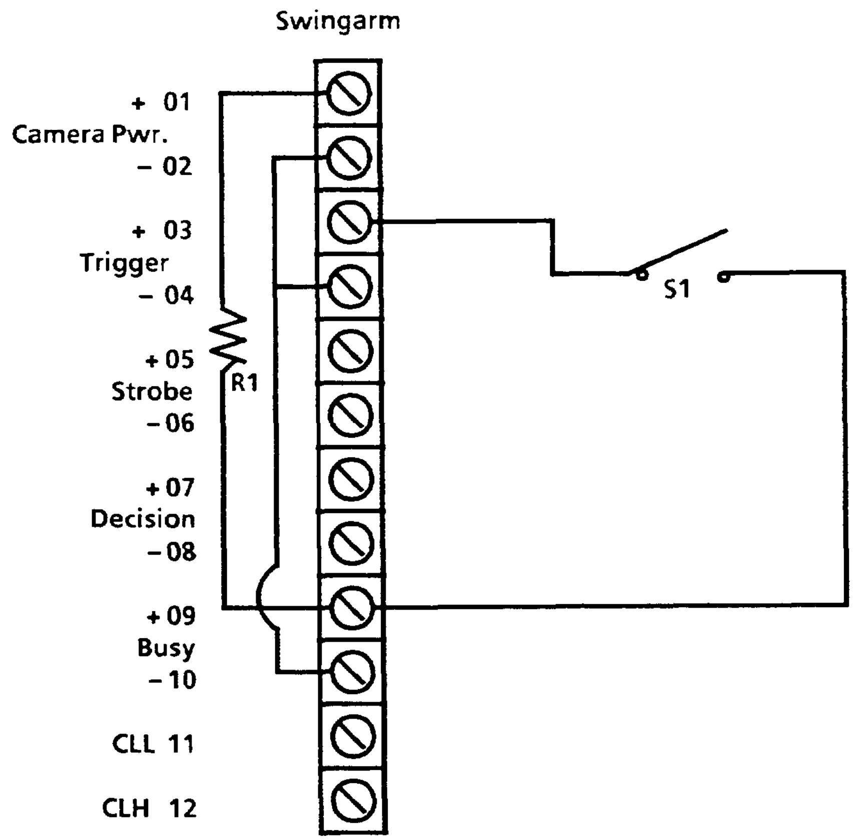


Swingarm Connections (continued)

Trigger Input – Terminals 3 and 4

"Continuous" Push Button – The continuous trigger will repeatedly cycle the VIM2 module as long as the push button is pressed. The *Busy* output is used to loop back a *Trigger* signal.

Figure 5.17
"Continuous" Push-button Circuit



R1 = Resistor, 560 Ohms, 20%
S1 = Momentary Push Button. Normally open.
A-B Cat No. 800-AM2AR or 800-AM2AG
Open = No Trigger
Closed = Trigger Oscillations

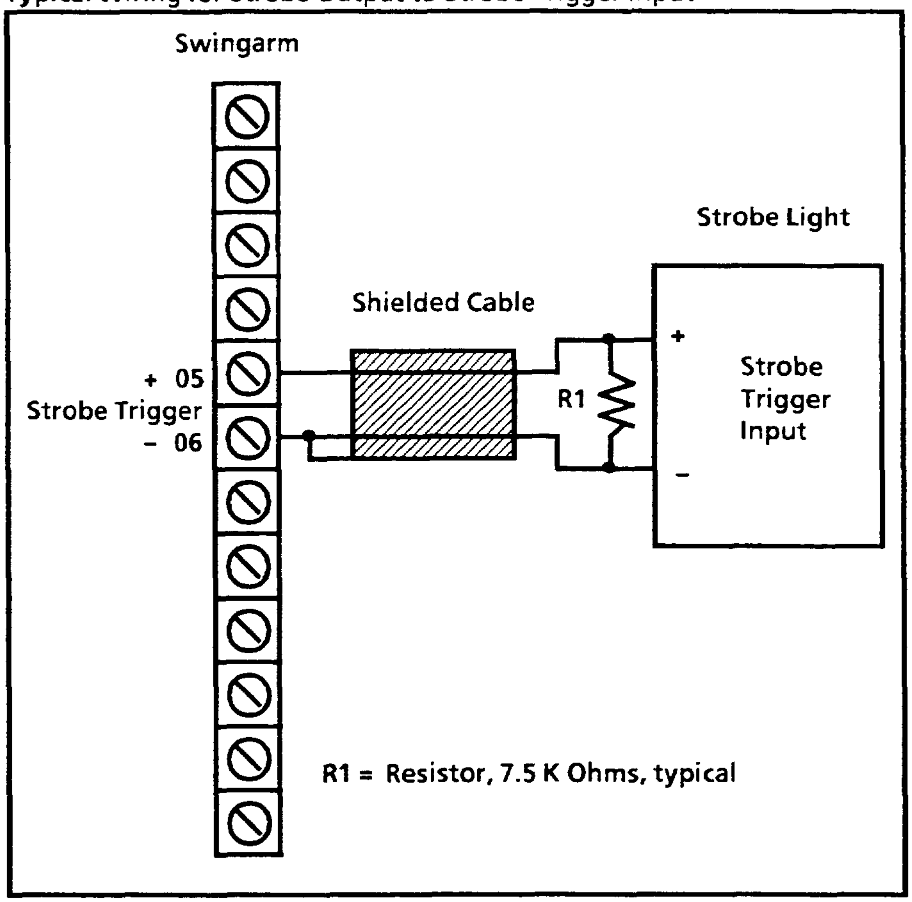
Swingarm Connections
(continued)

Strobe Light Output – Terminals 5 and 6

This output provides a 5 volt TTL strobe light (positive edge) trigger. The strobe trigger cable must be of shielded type and terminated (to 7.5k ohms, typical). Shielded cables reduce noise and interference and are highly recommended.

Note: Refer to Figure 5.18 for typical wiring to a strobe light trigger input.

Figure 5.18
Typical Wiring for Strobe Output to Strobe Trigger Input



Swingarm Connections *(continued)*

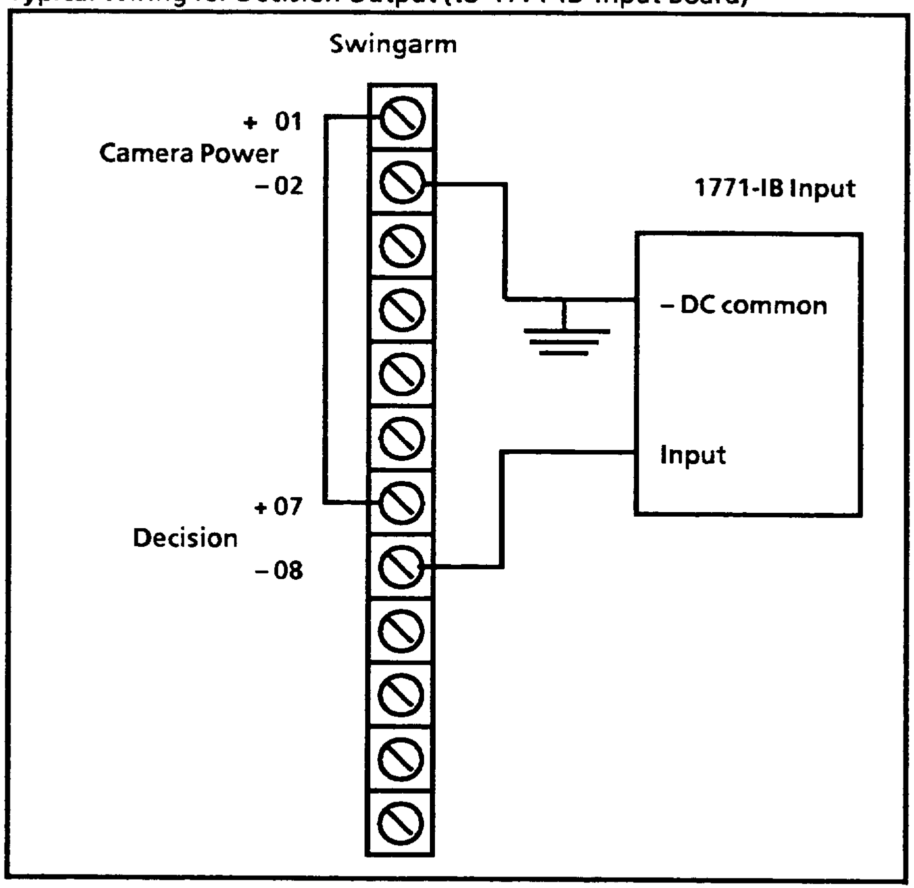
Decision Output – Terminals 7 and 8

The VIM2 module communicates summary accept/reject results through the *Decision* discrete output bit, and through the *Decision* terminal on the swingarm. The *Decision* transmits a *reject* signal if any of the acceptance range tests for the line gages, windows, probe, or math tools fail.

The swingarm *Decision* output toggle ON/OFF according to the *inspection result.* This is an open-collector transistor type output, rated for 3 to 32 VDC, 1 ampere. *Accept* = *Decision* LED OFF, high impedance (approximately 1 kohm). REJECT = *Decision* LED ON, low impedance to common.

Note: You can use the 12.5 VDC camera power as a current source for the *Decision* output. Refer to Figure 5.19 for typical wiring to a 1771 input board.

Figure 5.19
Typical Wiring for Decision Output (to 1771-IB Input Board)



Swingarm Connections (continued)

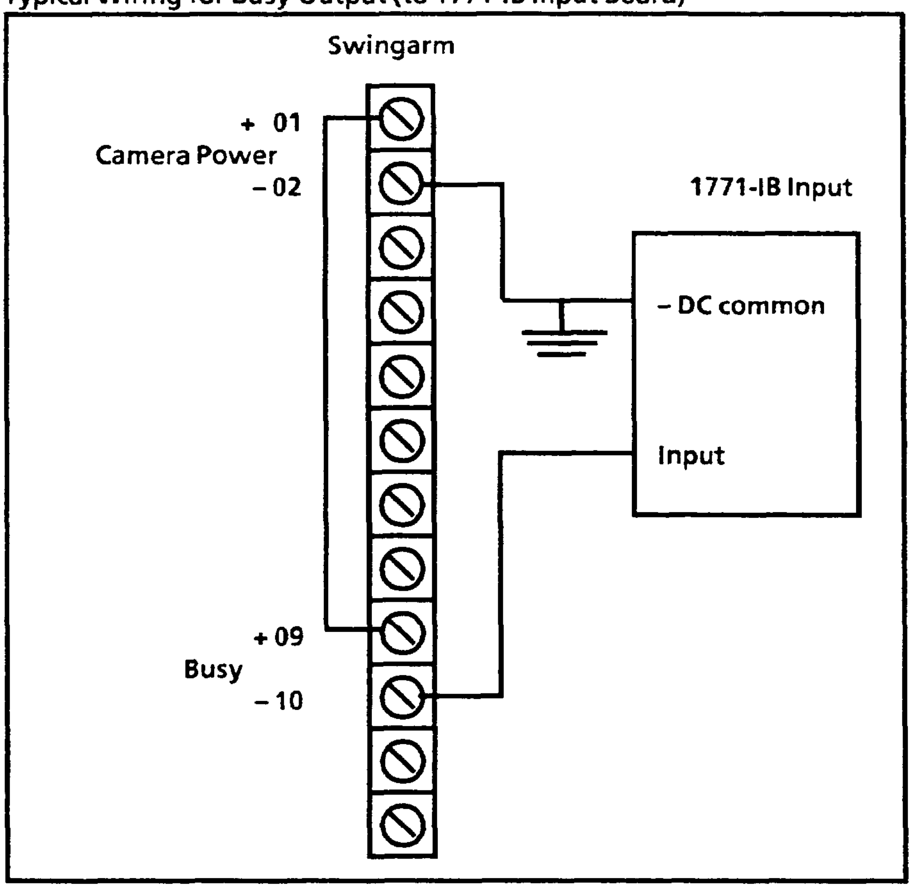
Busy Output – Terminals 9 and 10

The Busy signal indicates that the module is busy and a decision is pending. It comes on as soon as the trigger has been received and goes off after trigger has been reset and the *Decision* output is set/reset. This is an open-collector transistor type output, rated for 3 to 32 VDC, 1 ampere. *Busy* = low impedance to common and *Busy* LED ON. *Not Busy* or *Ready* = high impedance (approximately 1 kohm) and *Busy* LED OFF.

WARNING: Do not use the *Busy* output (terminals 9 and 10) to directly energize external equipment. The *Busy* output is energized during power-up and reset to indicate that the module is not yet ready to perform an inspection.

Note: You can use the 12.5 VDC camera power as a current source for the *Busy* output. Refer to Figure 5.20 for typical wiring to a 1771 input board.

Figure 5.20
Typical Wiring for Busy Output (to 1771-IB Input Board)



Swingarm Connections (continued)

RS-485 Port – Terminals 11 and 12

Connect your external host computer, if you intend to use one, to the RS-485 port at swingarm terminals 11 and 12. You will need to supply the communications cable to link your host device to the VIM2 module. Use shielded, twisted-pair cabling (such as Belden type 9841) for noise immunity.

The VIM2 RS-485 port has been designed to comply with Electronic Industries Association (EIA) standards for RS-485 balanced digital point-to-point systems. Consult the EIA RS-485 standards, if necessary, for additional information. Also see Chapter 15 for more information regarding the use of the RS-485 port.

Communication terminals (11 and 12) are marked as follows (see Figure 5.21 for typical wiring to the RS-485 port):

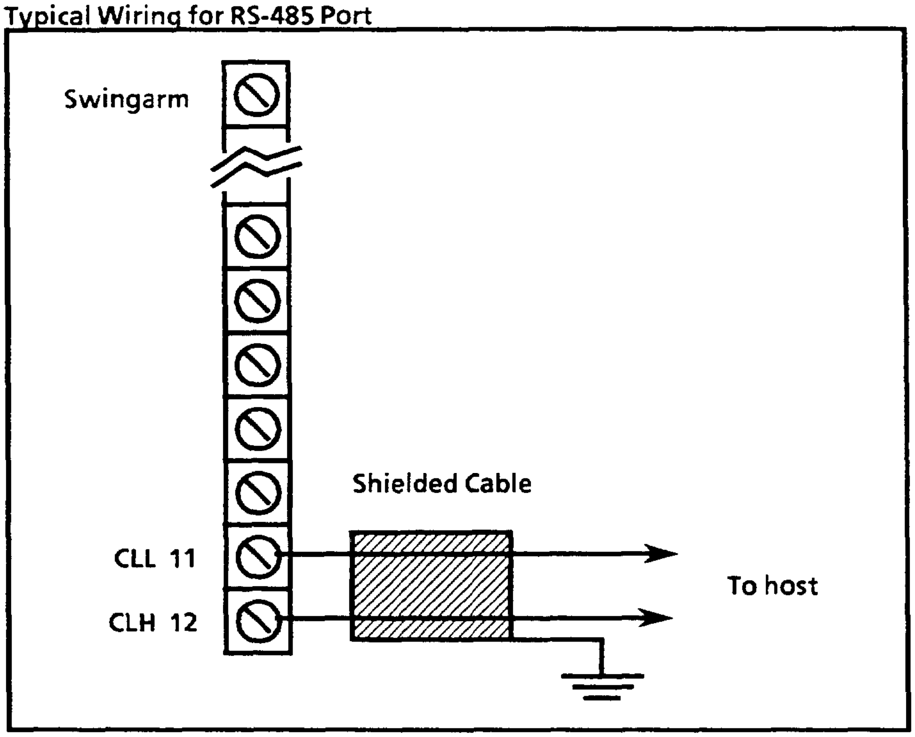
Terminal 11 – CLL (communication line low)

Terminal 12 – CLH (communication line high)

Note: Reversing the connections of the RS-485 port to the host will not damage the unit, but will prevent any communication from occurring.

For connecting host devices with RS-232 ports, use of an RS-232/RS-485 adapter is required.

Figure 5.21
Typical Wiring for RS-485 Port



Indicator LEDs

The VIM2 module is equipped with seven indicator LEDs. The LEDs are located on the front panel of the VIM2 module (see Figure 5.6) and depict the current status of the module.

PWR – This *Power Indicator* LED goes ON when the chassis power is *on*, OFF when the chassis power is *off*.

CPU FAULT – This LED goes ON when a possible hardware failure has been detected. During normal operation it will be OFF. When this light is ON, the module does not respond to triggers, and the *Decision* output goes OFF (low).

Note: At power-up, the CPU FAULT LED goes ON until the reset sequence is complete.

CONFIG FAULT – This *Configuration Fault* LED indicates that the module is not properly configured. This can be caused by a loss of memory, an invalid configuration download, a power loss during configuration transfer or by the loss of a "train-through-the-lens" mask due to a power outage. When this light is ON, the module will not respond to triggers, and the *Decision* output goes OFF. Clear the configuration fault by either of two means:

- Configure the VIM2 module with the light pen, or
- Download a configuration from the master host device (PLC or serial host), if there is one.

The **CONFIG FAULT** LED goes ON at power-up until the reset sequence is complete.

ACQ ERROR – This *Acquisition Error* LED indicates that the camera is not acquiring a satisfactory image. The LED goes ON when any of these conditions occur:

- The brightness probe is out of range. This can be caused by a disconnected camera or loss of illumination, such as a burnt-out light bulb.
- Either the X or Y float gage values are out of range or in error, such as if, for example, the X or Y-gage fails to find a blob edge.
- Any window or line gage is floated too far so that it collides with the edge of the screen. "Through-the-lens" windows will not cause an ACQ ERROR.

The "ACQ ERROR" conditions cause a *reject* decision, and cause the *Decision* LED to go ON.

Indicator LEDs (continued)

SWG TRIGGER – This LED goes ON when a trigger request is made from the swingarm. It remains ON as long as the input is held high.

> **Note:** This LED does not indicate trigger requests from the master host (either PLC or serial host).

BUSY – This LED goes ON when the module is actively performing an inspection. This LED is yellow and goes OFF when the inspection cycle is complete and when the *Trigger* input is reset. Also, this LED is on during software download operations and configuration setup.

DECISION – The *Decision* LED goes ON when a *reject* decision is delivered. It remains on until an *accept* decision is made. The output is valid only when the *Busy* LED is OFF and *Busy* output is low.

Typical Inspection Handshake Sequence

The handshake sequence for triggering an inspection (upon which the PLC or serial host communications can be based, for example) is as follows (see Figure 5.22):

1) The master host sets the *Trigger* bit (PLC master host) or issues a command (serial master host) to initiate an inspection cycle (see Figure 5.22).

2) The host (if PLC master host) monitors the *Busy* output bit, and waits for the *Busy* bit to go high, indicating the VIM2 module has begun the inspection.

3) The VIM2 module receives the *Trigger* signal or command, and responds by setting the *Busy* bit. Then the VIM2 module begins the inspection of the image.

4) The host receives the *Busy* signal (the PLC, if master host, responds by clearing the *Trigger* bit). The host waits for the *Busy* bit to go low.

5) The VIM2 module completes the inspection. Since the host *Trigger* is now low (if PLC is master host), the VIM2 module places the *Decision* output to the appropriate level (either low (*accept*) or high (*reject*)). Then the VIM2 module sets the *Busy* bit to low to indicate the *Decision* bit is properly set, and that the VIM2 module is ready for the next inspection.

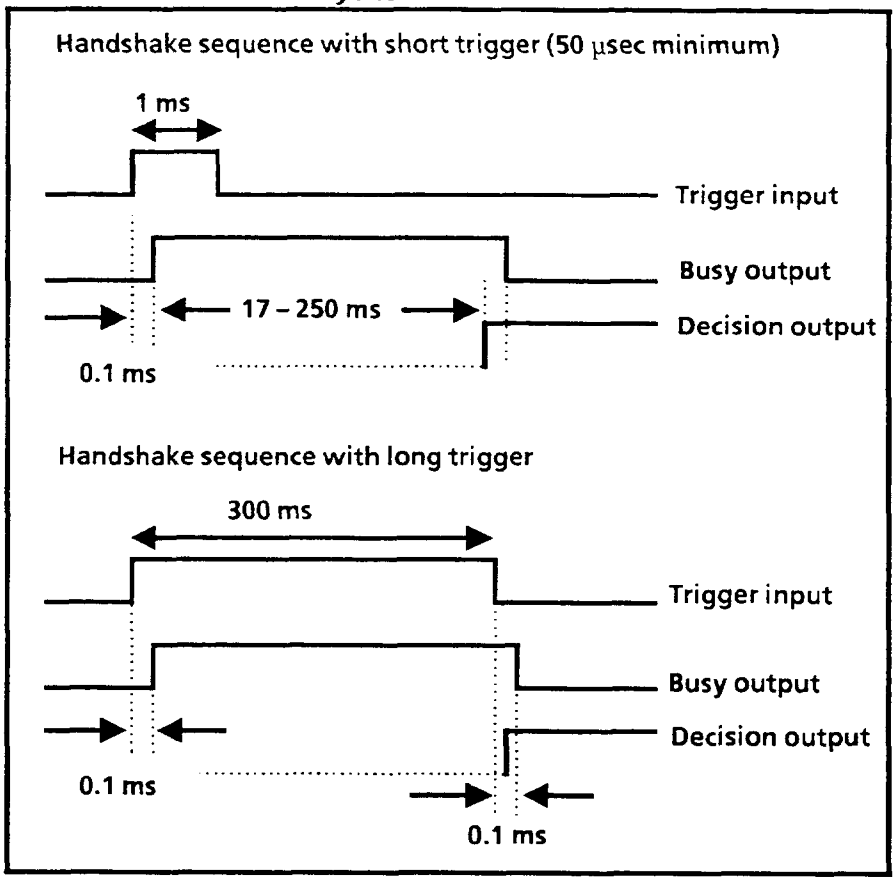
6) When the host sees the *Busy* bit go low, it reads the *Decision* bit for the *accept/reject* result of the inspection.

Typical Inspection Handshake Sequence (continued)

7) (Optional) – If a PLC is used, the PLC can perform a block transfer read the results block (after having set the discrete bits necessary to specify the read request). If a serial host is used, it can send a "read results" command. The host then uses the results to make more complex decisions or to make process control updates. (**Note:** Neither host device – PLC or serial host – has to be the master host in order for it to read results).

8) The inspection cycle is complete. The next cycle can now begin.

Figure 5.22
VIM2 Module Handshake Cycles



Inspection Cycle Time

"Inspection Cycle Time" is the elapsed time from the trigger until the inspection is complete. The inspection cycle time depends upon the configuration setup, the controller, and the image.

At a minimum, it takes 1/60th of a second to receive and process a new image from the camera.

Inspection Cycle Time *(continued)*

The module typically can capture an image from the camera and analyze it in less time than it takes for the next workpiece to advance into position on the production line. The following section provides timing information which may help you in high-speed situations where the inspection cycle time is critical.

The fastest VIM2 module cycle time attainable is achieved using TRIGGER, BUSY, and DECISION signals via the swingarm. With all four (4) windows enabled and anchored and all line gages and math tools disabled, the inspection cycle time is 1/60th of a second.

Enabling the X-gage and Y-gage and floating the four (4) windows in X and Y results in a cycle time of approximately 1/15th of a second. Employing filtering to eliminate transitions of length 1 pixel for both the X-gage and Y-gage does not change this number appreciably unless there are many white to black and black to white transitions along these gages.

Inspection cycle times increase as more line gages are enabled, as the lengths of the line gages are increased, as math tools are enabled, and/or as statistics are computed. See Figure 5.23 for example inspection times.

In the worst case (without using math tools, or computing statistics), enabling and floating all four (4) windows, enabling all twenty-two (22) line gages as 128 pixels in length, floated in X and Y and using Filter Selection 2 results in a cycle time of approximately 1/6-1/4th second .

When you must know precisely what the cycle time is with your particular configuration, the best way to obtain that information is to measure it directly. This may be done by connecting an oscilloscope to the TRIGGER and BUSY terminals on the VIM2 module's swingarm. Trigger the scope scan with the rising edge of the TRIGGER signal. Measure the time from this trigger until the BUSY signal drops. This is the *inspection cycle time.* Make sure to measure it under a variety of image conditions – this may affect the cycle time somewhat.

Inspection Cycle Time
(continued)

Figure 5.23
Inspection Cycle Times

Inspection Rate With All Windows Active, Swingarm Trigger Activated And No Block Transfers to PLC.

	SPEED Inspect/Cycles Per Second	XY Position Compensation	Lighting Compensation	Strobe Light Enabled	# of "Standard" Line Gages
		(1)	(4)		(2)
(BEST CASE) -FASTEST	60 (3)	N	N	N	0
	30	N	Y	N	0
	30	N	N	Y	0
	30	N	Y	Y	0
	30	Y (6)	Y	Y	4
	20	Y	Y	N	4
	15	Y	Y	Y	4
	12	Y	Y	Y	10
	10	Y	Y	Y	16
	8	Y	Y	Y	22
(WORST CASE) -SLOWEST	4	Y	Y	Y	22 (5)

(1) At least 1 window has XY float option.
(2) "STANDARD" line gage is about $\frac{1}{4}$ screen length -64 pixels long, medium (1-PIXEL FILTER) filter option and simple image. The two X and Y source-of-float gages are not included in number.
(3) Case only applies to continuous inspections.
(4) Mode 2 Lighting Compensation: FULL IMAGE threshold adjust. No compensation or Mode 1 (Immediate Brightness Compensation) causes no speed penalty.
(5) General worst case (SLOWEST). All lines maximum length, heavy (2-PIXEL FILTER) filter option and complex (VERY BUSY) image.
(6) No XY floating windows allowed. All 4 line gages are XY-floating.

Note: The use of math tools, to the degree they are used, and/or collecting statistics in the *learn* mode, will impact cycle times.

Powering Up the VIM2 System

After you have completed the installation procedures, and all the components are connected together, you can connect power to the system in order to check the power-up indicators, and prepare for the next chapter, "Quick Start – An Introduction to the User Interface."

Follow these steps in powering up the VIM2 module:

1. Plug the power cord for the video monitor into the AC outlet.
2. Set the monitor POWER switch to ON.
3. Connect the external camera power supply to AC power.
4. Plug the VIM2 system power cord into the AC outlet.
5. Turn the *POWER* switch on the rack power supply to ON.

Successful Power Up Indications

Upon completing the above steps, look for the indications listed below for a successful VIM2 module powerup:

1. Three VIM2 module LEDs go ON – the *CPU FAULT*, *CONFIG FAULT*, and *POWER* LEDs.
2. Next, the *ACQ ERROR* and *BUSY* LEDs go ON.
3. The *CPU FAULT*, *CONFIG FAULT*, and *ACQ ERROR* LEDs go OFF.
4. The video monitor sequentially displays the four window / threshold images, then displays a thresholded video image from the camera.
5. The *BUSY* LED goes OFF.
6. If the LEDs and monitor follow this sequence of indications, the powerup is successful; continue on to Chapter 6, for an introduction to the user interface.

Note: The first time you power up a new VIM2 module, the *CONFIG FAULT* and *BUSY* LEDs stay ON at the end of the powerup sequence (this is because the VIM2 modules comes from the factory without a stored configuration). To clear the *CONFIG FAULT* and *BUSY* indications, place the VIM2 module in setup mode using the light pen and monitor (see Chapter 6), or download a configuration from the PLC (see Chapter 14) or from the serial host (see Chapter 15).

Power Up Failure Indications

If the *CPU FAULT* LED remains ON after powerup, the power up sequence failed, indicating a malfunction. Try turning the rack power supply switch OFF, then switching it back to ON. If this fails to bring about a successful powerup sequence, the VIM2 module must be replaced.

Powering Up the VIM2 System *(continued)*

Power Up Failure Indications *(continued)*
If, after powerup, the *CPU FAULT*, *CONFIG FAULT*, and *POWER* LEDs remain on, check the I/O Chassis configuration plug (see page 5-5 for plug placement guidelines).

VIM2 Powerup Host Check Sequence

When the VIM2 module is powered up, regardless of the master host selection (PLC, serial, or no master host – see Chapter 7) it first conducts a number of self-diagnostic tests. The VIM2 module also checks the two configurations stored in the EEPROM; if both are valid, the selected Starting Configuration is downloaded from EEPROM to RAM to become the working configuration.

The VIM2 module then checks to see if a host is present. If no PLC host is present, and the PLC is the master host, the VIM2 module reverts to operation in the standalone mode. If a PLC is present, then the VIM2 module begins operation in run mode, unless the *Unlock* bit is asserted (see Chapter 14).

If the serial line has been selected as the master host, the VIM2 module waits for five seconds for a command from the serial host after the diagnostics are complete. If a command from the serial host *is not* received within five seconds, the VIM2 will automatically be "unlocked," and the VIM2 module will go to *standby** mode. The selection of serial line as the master host is maintained by the VIM2 module.

If a command from the serial host *is* received within five seconds, the VIM2 module will be locked and will go to run mode (unless the command is the *Unlock* command, in which case the VIM2 module goes to *standby** mode).

***Note:** *Standby* mode is an interim mode of the VIM2 module which occurs upon leaving *run* mode, and before entering *setup* mode. When the VIM2 module is in *standby* mode, the title banner is displayed on the monitor. The VIM2 module enters *standby* mode when:

- The VIM2 module, in *run* mode, is unlocked by the master host.
- The VIM2 module is in *setup* mode and a master host is present, and you exit *setup* mode by picking the *Run* icon.
- The VIM2 module is in *run* mode and operating as a standalone unit, and you pick the monitor screen once with the light pen.

(This page intentionally left blank)

Chapter 6 Quick Start – Introduction to the User Interface

Chapter Objectives

The objective of this chapter is to provide a "quick start" self-teach procedure which introduces you to the light pen and video monitor (the "user interface") you will use to set up the VIM2 module.

After completing this chapter, you will understand the basics of accessing icon menus and selecting icons to configure the VIM2 module (**Note:** For more self-training material, refer also to the *VIM Self-Teach Manual*, Cat. No. 2803-SM2, which was written for the Cat. No. 2803-VIM1).

The User Interface

The term "user interface" refers to the specific devices you use to interact with the VIM2 module. The user interface consists of two main parts:

1. The light pen.
2. Graphic figures on the video monitor representing VIM2 module's functional features.

You use the light pen and monitor to configure the VIM2 module for vision inspection tasks. The VIM2 module has been designed for easy operation using "icons." Icons are small symbols that can be selected to activate a function, perform a task, or move through selections of options. You select the icons by "picking" them with the tip of the light pen (see Figure 6.1).

Figure 6.1
"Picking" an Icon Using the Light Pen

87-283-7

Picking with the Light Pen

The light pen has a switch in the pen tip. You activate the light pen by depressing the pen tip, which activates the switch. When you point the light pen at the video screen, and depress the tip against the screen, this is called "picking." When you place the light pen tip over an icon shown on the monitor, and depress the light pen tip against the icon, you "pick," or select, that icon.

Other Light Pen Functions

There are other ways to use the light pen besides picking icons on the monitor screen. These include:

- **Run Mode Display Selection** – You can depress the light pen tip away from the monitor, while the VIM2 module is in "run" mode (see Chapter 13), to cycle through six different monitor display options, and select an appropriate display. The six display modes are –

  (1) Show threshold image #1 and window #1

  (2) Show threshold image #2 and window #2

  (3) Show threshold image #3 and window #3

  (4) Show threshold image #4 and window #4

  (5) Show threshold image #1 and windows #1-4

  (6) Show "live" analog image (direct – camera to monitor)

  The threshold images 1-4 are high contrast images; each pixel on the screen is either white or black – there are no intermediate gray levels, as with a "live" analog image.

- **Setup Display Selection** – If you pick a bright area* in the top one-half of the monitor screen, while the VIM2 module is in setup mode, you can cycle through three image options:

  (1) Show the current thresholded image, but do not show the icon menu strip on the monitor screen (this is helpful during set up if the menu strip is blocking a feature of interest on the monitor image).

  (2) Show the "analog" image direct from the camera (this is helpful for aiming and focusing the camera).

  (3) Return the menu strip and current thresholded image to the screen.

***Note:** If there is no bright area on the top part of the screen the light pen may not be able to "pick." In this case, turn up the *BRIGHT* knob on the video monitor and try again.

Quick Start Procedure

This section guides you through a "quick start" self-teach procedure, which demonstrates the VIM2 module user interface and also some of the VIM2 module functions.

Note: Before powering up and using the VIM2 system components as described in this section, make sure the VIM2 system is not currently incorporated in a vision inspection process, or otherwise being utilized, and that you have proper permission to use the VIM2 module for demonstration and/or setup purposes.

Setting up the Quick Start Workplace

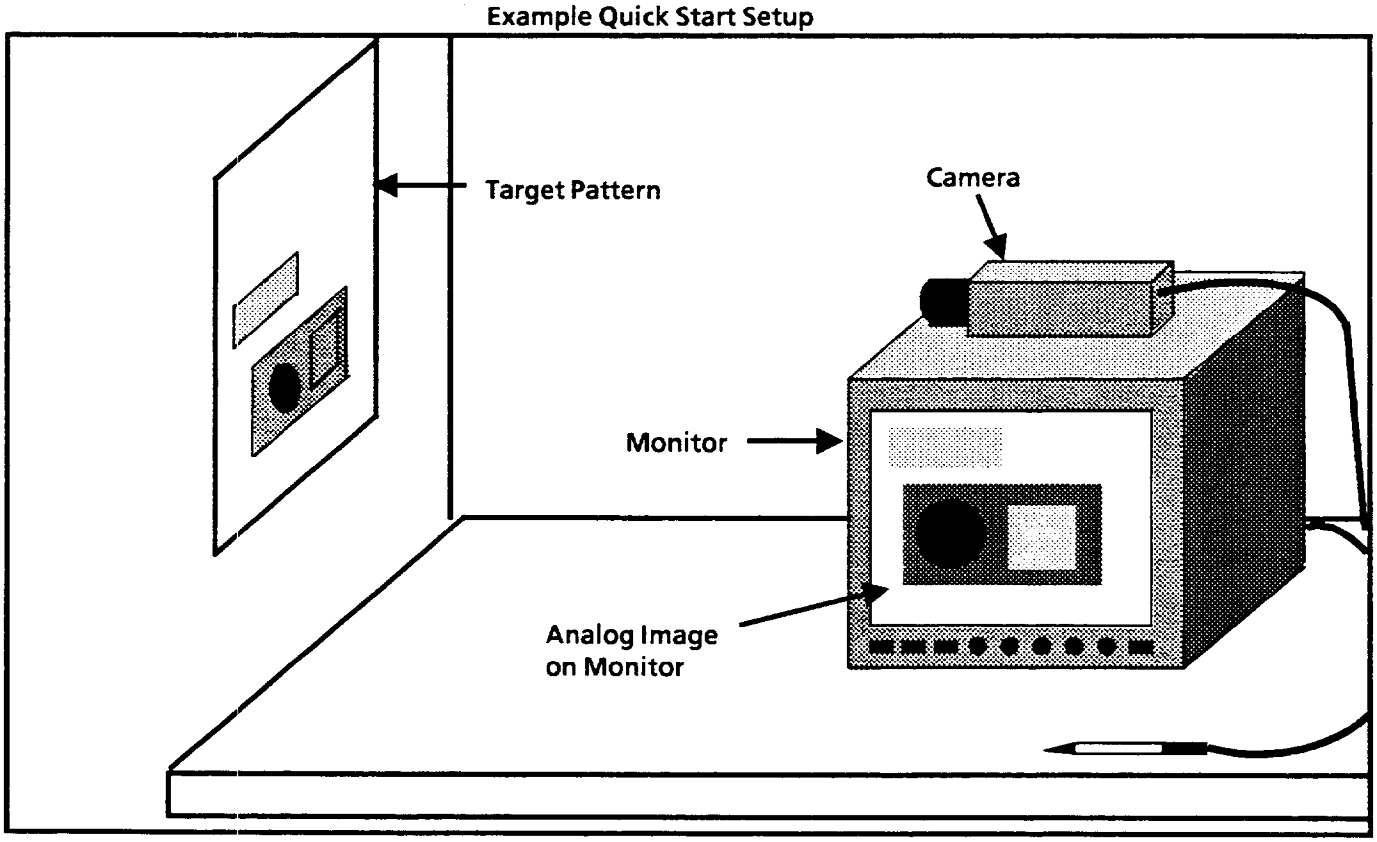
In order to demonstrate some of the VIM2 capabilities, the VIM2 system must first be connected and powered up (see Chapter 5). Next, the video camera must aimed and focused at the target pattern supplied (see Figure 6.3). In order to accomplish these tasks, refer to the steps below:

Your Action	Comments
Secure a location to serve as your Quick Start workplace.	Find a location for connecting and using the VIM2 system module, rack, monitor, power supply, and light pen, which has the following available for use: • A desk top, table top, or other flat horizontal surface capable of supporting the video monitor and camera. • A nearby vertical surface (wall or partition) for mounting the target pattern (or, if necessary, use the side of the video monitor). • Exposure to normal ambient room lighting.
Connect and power up the VIM2 system.	Once you have secured an appropriate location, connect and power up the VIM2 system (see Chapter 5), placing the video monitor and camera on a table top or other flat horizontal surface (see Figure 6.2 for an example setup).
Remove the target pattern from inside the back cover.	The target pattern shown in Figure 6.3 is also printed on card stock and located on the inside of the back cover.
Mount the target pattern.	Using tape or other adhesive, mount the target pattern on a vertical surface near the video monitor (or use the side of the video monitor). Position the target pattern with the top side up, as shown in Figure 6.2.
Position and aim the camera toward the target pattern.	Place the camera in front of the target pattern, about 20 inches away from the pattern at first (the final distance will vary, depending on the type of lens). Aim the camera toward the pattern so that the pattern is shown on the monitor.

Setting up the Quick Start Workplace
(continued)

Your Action	Comments
Depress the tip of the light pen to select the "live" analog monitor image.	Depressing the light pen tip with your finger or thumb changes the monitor display; cycle through the different monitor displays until the "live" analog image of the pattern is shown on the video monitor.
Turn the focal ring on the camera to focus the image.	Turn the focal ring on the camera until the monitor image of the target pattern is sharp. If the image does not come into focus, adjust the camera-to-target distance, and try again.
Correct the size of the target pattern in the monitor image.	Compare the video image with Figure 6.4. If the target pattern takes up more of the video monitor screen than is indicated in Figure 6.4, move the camera back; if the target takes up less of the video monitor screen than is shown in Figure 6.4, move the camera toward the target.
Aim and refocus the camera.	Aim the camera to center the pattern in the monitor image. Refocus the camera as necessary.
Secure the camera in position.	With the camera in position and focused, secure the camera in position.

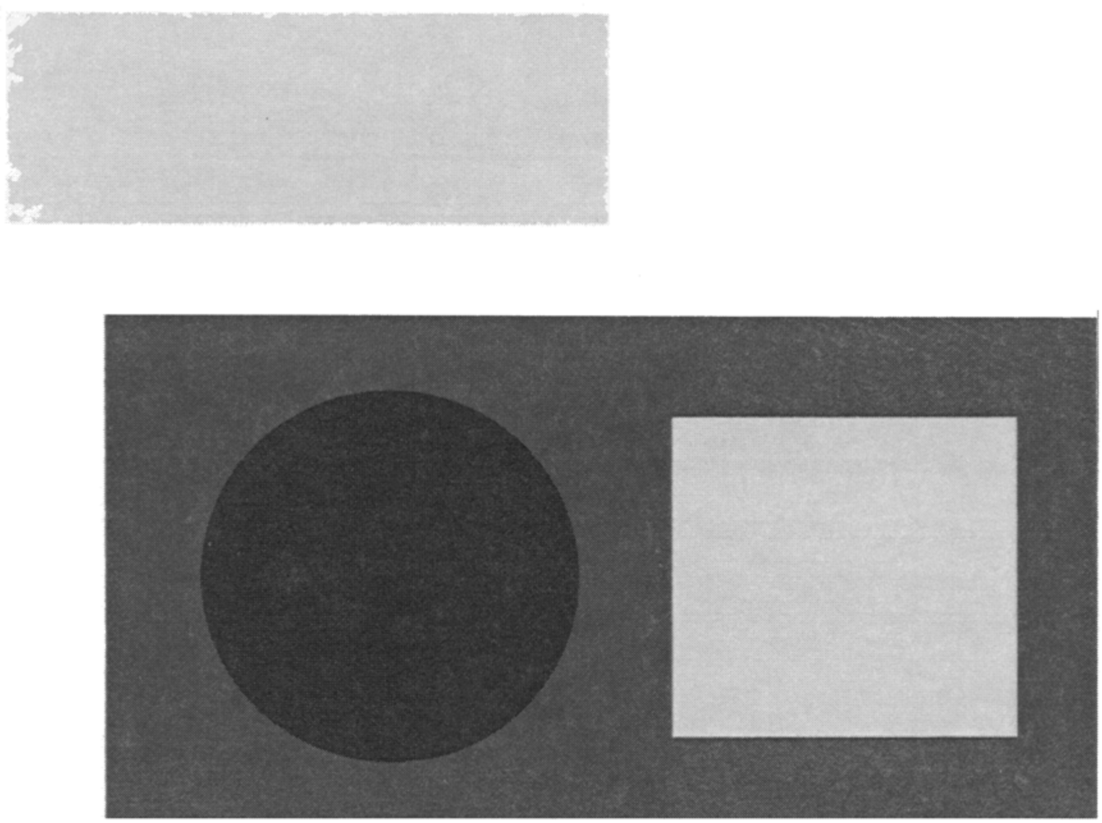
Figure 6.2
Example Quick Start Setup



Setting up the Quick Start Workplace
(continued)

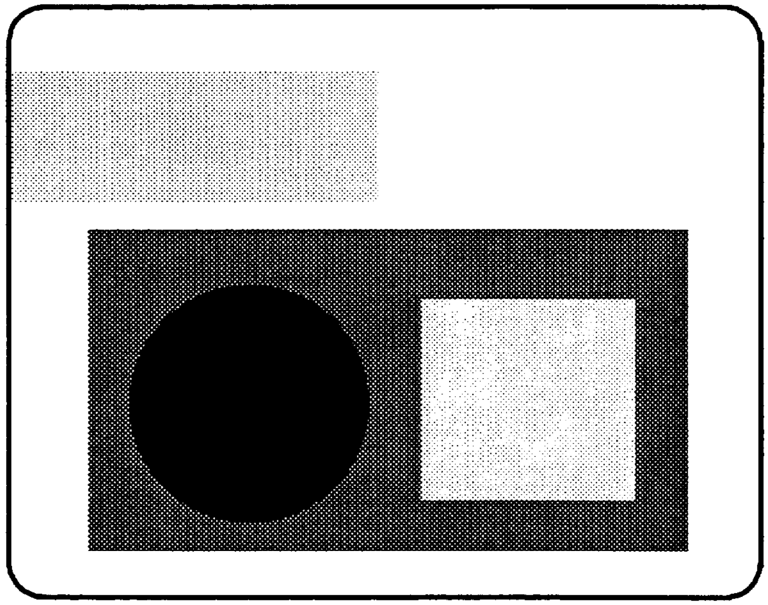
Figure 6.3
Quick Start Target Pattern

Note: Do not remove this page from the manual for use as the target pattern – instead use the target pattern included inside the back cover of this manual.



Setting up the Quick Start Workplace *(continued)*

Figure 6.4
Monitor Image of the Target Pattern



Accessing the Setup Mode and Main Menu

With the focused target pattern on the monitor, begin to use the light pen and monitor to access the icons and menus, starting with the *Main* menu. In the following steps, you will place the VIM2 module in *setup* mode, allowing you to access the icons and menus, and configure the VIM2 module.

Your Action	Comments
Pick the monitor screen to display the title banner.	Depress the tip of the light pen against the screen once; the Allen-Bradley VIM2 module banner is displayed.
Pick the screen again to activate the "setup" mode.	When you pick the screen again, the VIM2 Module goes into the setup mode; the *Main* menu* appears, and you can begin to configure the VIM2 module:

Main Menu

*Note: If the *Main* menu does not appear when the screen is picked, the VIM2 module may be "locked" – if the VIM2 module is integrated with a PLC, or connected to a host through the RS-485 port, and one of these devices is the "master host." To "unlock" the VIM2 module and allow configuration, see Chapter 14 or 15, as appropriate.

Using the Icons and Menus

The *Main* menu is a series of individual icons, each representing a specific function or acting as a gate to other menus. Before using the light pen and *Main* menu icons, read the information below:

Icons
The icons represent the functional options and tools which the VIM2 module uses to conduct vision inspections. Each icon is designed to symbolically represent the function which it controls. The icons are presented in sets called "menus" (the *Main* menu is typical of the menus you'll see on the VIM2 module video monitor).

Menus
A menu is a group of icons associated with a particular vision task. Each menu appears in the black bar displayed near the bottom of the screen. A menu consists of a set of icons related to a specific task or tool, and also contain icons which take you to other menus, such as the three described below.

- **Menu Access Icons** – The "menu access" icons are identified by an *angular cut on the lower-right corner*. Picking a menu access icon displays the menu represented by the menu access icon you picked.

Setup Icon

Angular Cut

- **ETC Icons** – ETC appears on a menu when all of the icons related to a tool cannot be displayed on one monitor screen. Picking ETC (et cetera) displays an additional set of icons relating to the current menu selected. The additional set of icons also includes an ETC icon; picking it returns the original menu.

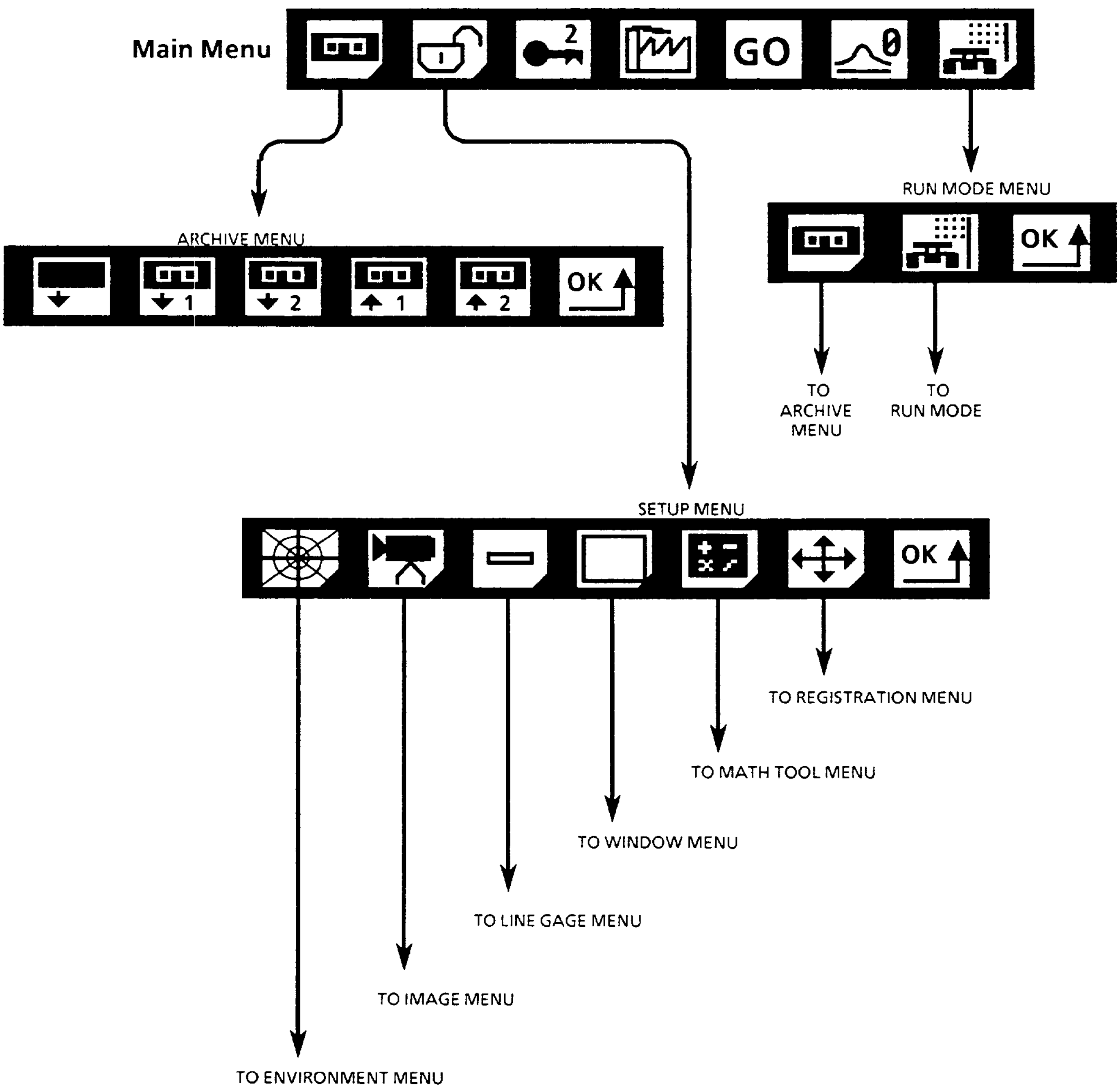
ETC (et cetera) Icon

- **OK Icons** – The "OK" icon is used to exit a selected menu and return to the previous menu. Repeated picking of the "OK" icons eventually returns you to the *Main* menu.

OK Icon

The Menu Branching Map

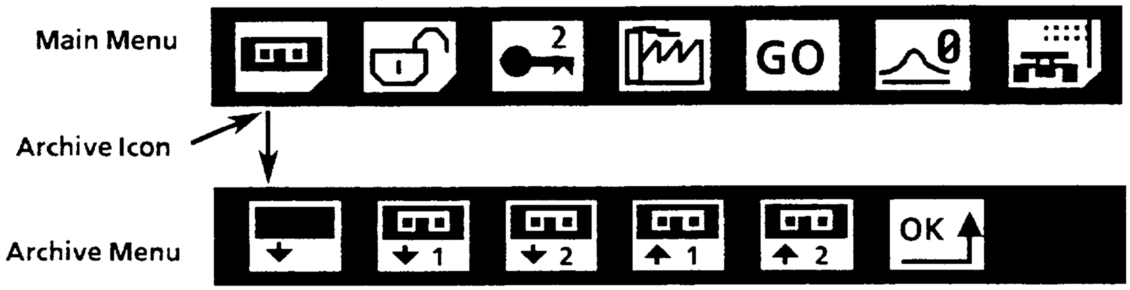
The menus and icons are strategically situated to allow you to access the necessary icons within a single menu to complete one setup task or subtask per menu. The result is a menu "branching" structure that can be represented as a "branching tree chart." The diagram below shows the major menu access icons and the major branches in the menu structure. See Appendix A for a complete menu branching chart.



Using Main Menu Icons

The following is a brief demonstation of some of the *Main* menu icons. You can, at your option, pick each icon as it is discussed in order to illustrate the use of each.

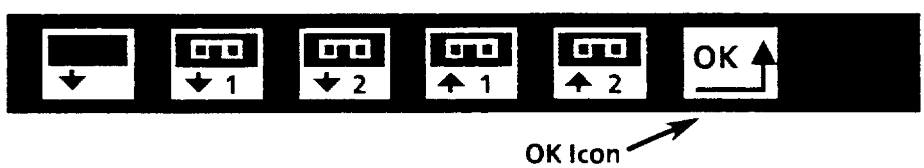
Your Action	Comments
Pick the Archive icon.	Notice that the *Archive* icon has an angular cut in the lower right corner; this indicates that picking this icon will display a subsequent menu.



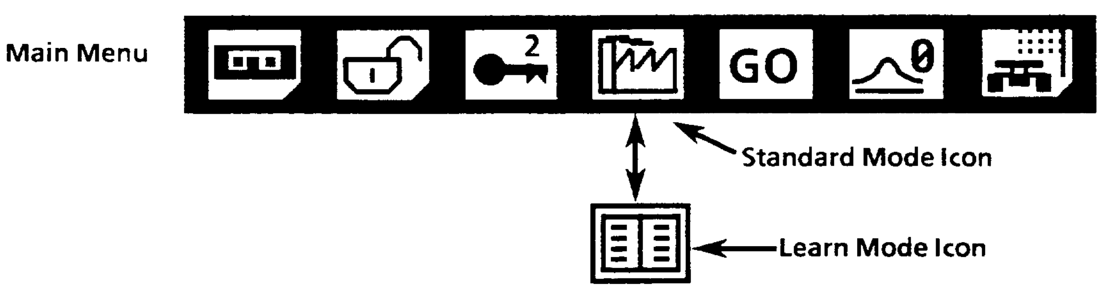
When you pick the *Archive* icon, the *Archive* Menu appears, *replacing the* Main *menu in the icon strip.*

Use the *Archive* menu after you configure the VIM2 module to save the RAM (working) configuration to one of two EEPROM stored configurations. Or, you can load a previously stored EEPROM configuration as the working configuration (Chapter 13 discusses archiving in more detail).

Pick the OK icon. Picking "OK" on the *Archive* menu returns the *Main* menu:



Pick the Standard / Learn icon several times. Notice there is no angular cut in the lower right corner of this icon. When you pick this icon, it toggles between two icons:

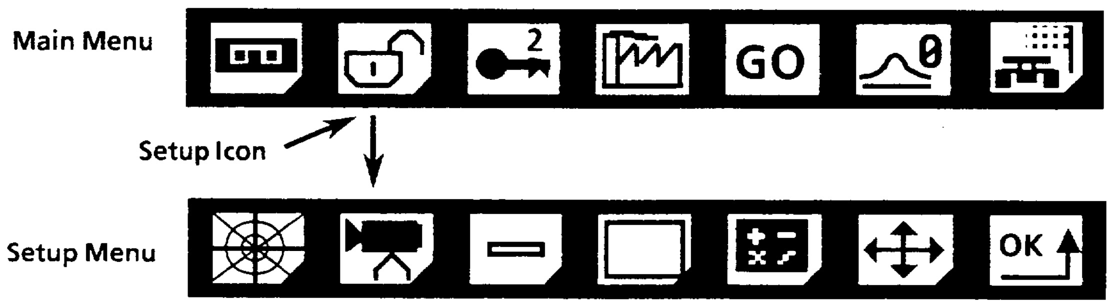


Select Standard mode. For now, pick the *Standard/Learn icon* so the *Standard* mode icon is displayed (Chapter 13 discusses the *Learn* and *Standard* modes in more detail).

The Setup Menu

The majority of the icons used for configuring the VIM2 module are accessed through the *Setup* menu (**Note:** Each icon on the *Setup* menu is a menu access icon, except for the *OK* icon; picking any *Setup* menu icon displays a subsequent menu). To see the *Setup* menu icons, access the *Setup* menu:

Your Action	***Comments***
Pick the Setup icon.	When you pick the *Setup* icon, the *Setup* menu appears:



Here are brief descriptions of the *Setup* menu icons:

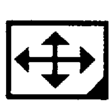
Environment – Pick this icon to select operating environment parameters such as standalone or hosted operation, master host device (serial port vs. PLC), baud rate (for serial port), trigger source (host or swingarm), and setup trigger source.

Image – Pick this icon to set such parameters as brightness compensation, threshold images, and strobe or non-strobe light usage.

Line Gage – Pick this icon to set up line gages and/or float gages for use by the VIM2 module in the inspection process. Select orientation, adjust length and position, select function, and select threshold image for each.

Window – Pick this icon to set up inspection windows for use by the VIM2 module in the inspection process. Select shape, adjust size and position, and select function.

Math Tool – Pick this icon to set up math tools for use in the inspection process. Select from eight math tool operations, and reference inspection windows, gages and/or other math tools for use in the math tool formula.



Registration – Pick this icon to automatically shift all inspections tools according to a new nominal part position, and/or automatically shift all threshold image settings according to a new nominal brightness level.

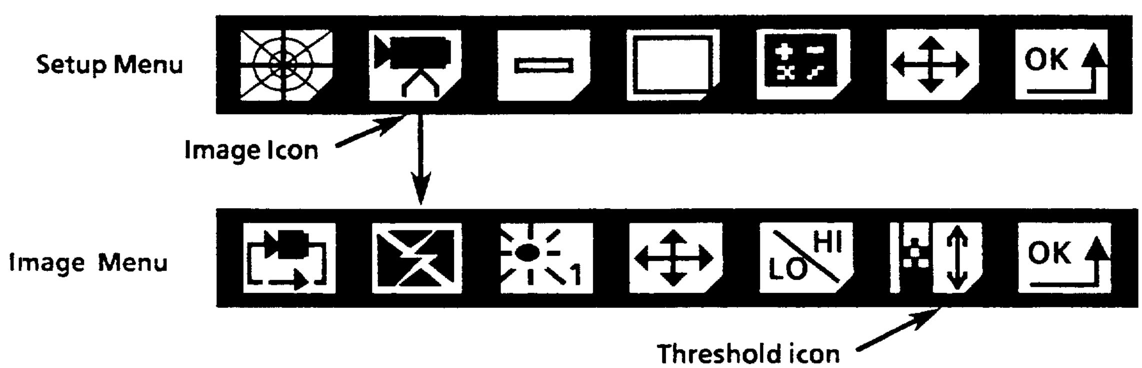
Using Setup Menu Icons

Picking any of the *Setup* menu icons accesses a subsequent menu represented by that icon. You then use the icons on the subsequent menu as necessary in order to select or adjust the operating parameters controlled by that menu.

To illustrate the use of the *Setup* icons and subsequent menus, perform example procedure below.

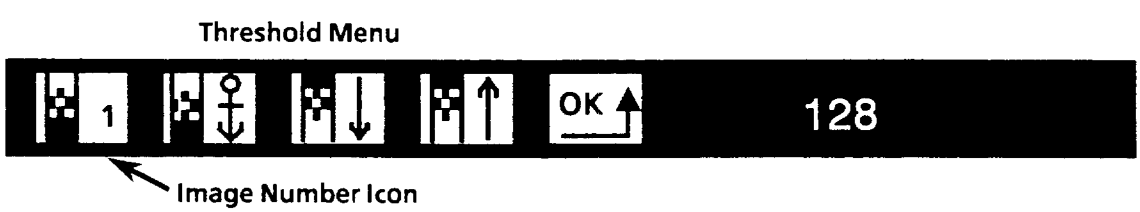
Example Procedure: Threshold Image Demonstration
This section steps through a procedure for setting up two of the threshold images. Recall from Chapter 3 that the VIM2 module produces four threshold ("binarized") images for use with line gages and windows. Also note that this section assumes the VIM2 system is connected and powered up, and that the video camera is aimed and focused upon the target pattern as described earlier in this chapter.

Your Action	*Comments*
Pick the Image *icon.*	Picking the *Image* icon displays the *Image* menu:



The *Image* menu consists of a number of icons used for adjusting the way the VIM2 module acquires images for processing. One of these is the *threshold* icon.

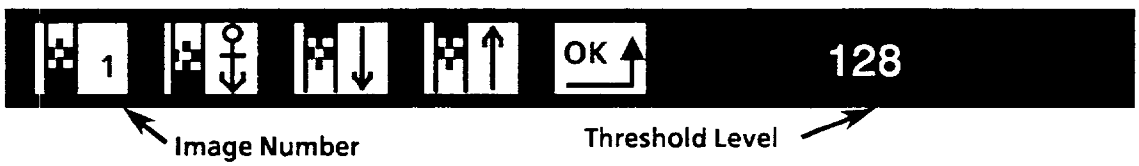
Pick the Threshold *icon on the* Image *menu to display the* Threshold *menu.*	Picking the *threshold* icon (see diagram above) displays the *threshold* menu (shown below). The *threshold* menu consists of icons used for adjusting each of the threshold images. One of these is the *image number* icon:



Using Setup Menu Icons
(continued)

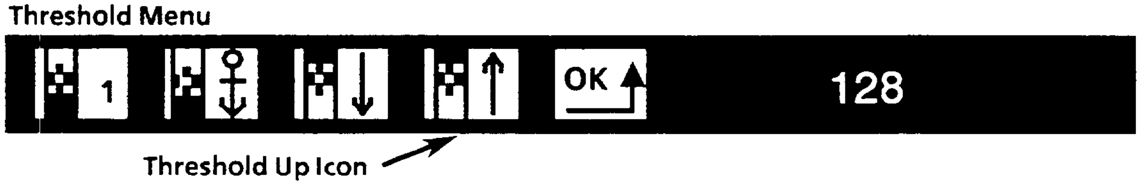
Your Action	Comments
Pick the Image Number *icon several times.*	The *Image Number* icon determines which of the four available threshold images is selected and displayed on the monitor screen. When a specific threshold image is selected, you can change the threshold level for that threshold image.
Select image number 1.	Pick the *Image Number* icon again until the number "1" is shown on the *Image Number* icon. With "1" selected, threshold image 1 is shown on the monitor.

Threshold Menu



Your Action	Comments
Observe the threshold level number.	Note the number displayed on the right side of the menu. This is the threshold level setting for threshold image 1. Recall from Chapter 3 that this threshold number has to do with the "binarization" of the video image. Pixels in the video image have brightness values ranging from 0 to 255. Within binarized, or thresholded, images, any pixel with a gray level above the threshold level number changes to 1 (white), while any pixel with a gray level lower than the threshold changes to 0 (black). The result is a picture with only two shades – white or black.
Pick the Threshold Up *icon several times to increase the threshold level.*	Picking the *Threshold Up* icon increases the threshold level number, one unit at a time.

Threshold Menu

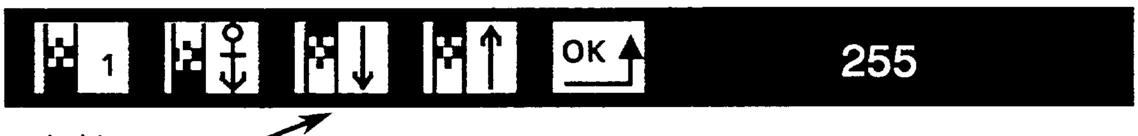


Your Action	Comments
Hold the light pen against the icon to increase the threshold level rapidly.	Holding the light pen against the icon so the tip is held down for a second or two – causes the threshold number to increase rapidly, until you lift up the light pen, or the upper threshold limit (255) is reached. Notice that as the threshold number increases, the screen darkens – more and more pixels have values lower than the threshold number.

Using Setup Menu Icons
(continued)

Your Action	Comments
Pick the Threshold Down *icon several times to decrease the threshold level.*	Picking the *Threshold Down* icon decreases the threshold level number, one unit at a time.

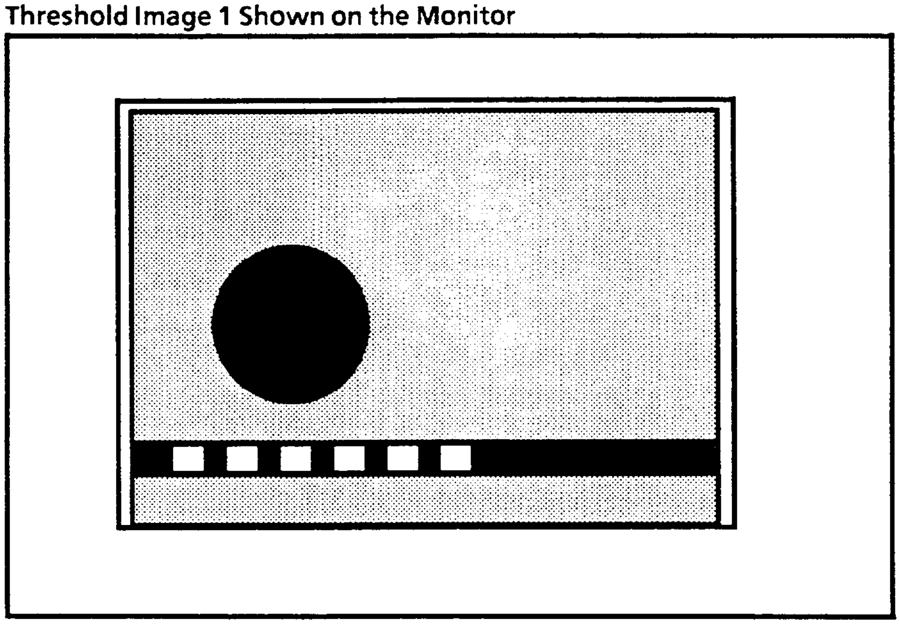
Threshold Menu



Threshold Down Icon

Your Action	Comments
Decrease the threshold level rapidly to 0.	Hold the light pen against the *threshold down* icon so the tip is held down for a second or two – the threshold number begins decreasing rapidly. Keep the light pen pressed against the icon until the lower limit (0) is reached. Notice that as the threshold number decreases, the screen has more white area – more and more pixels have values higher than the threshold number, until finally, at level 0, all pixels are white, and the screen is blank.
Set the threshold level of image number 1 so only the dark round spot is shown on the monitor.	Hold the light pen tip against the *threshold up* icon, increasing the threshold number until the dark spot on the left of the target pattern begins to appear. Then, pick the *threshold up* or *down* icons as necessary until the dark spot appears solid, and no other dark spots or "noise" (random black or white pixels) appear in the image (see Figure 6.5).

Figure 6.5
Threshold Image 1 Shown on the Monitor

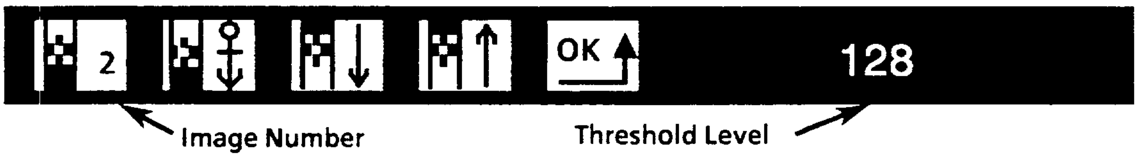


Using Setup Menu Icons *(continued)*

With threshold image 1 set as shown in Figure 6.5, now set threshold image 2.

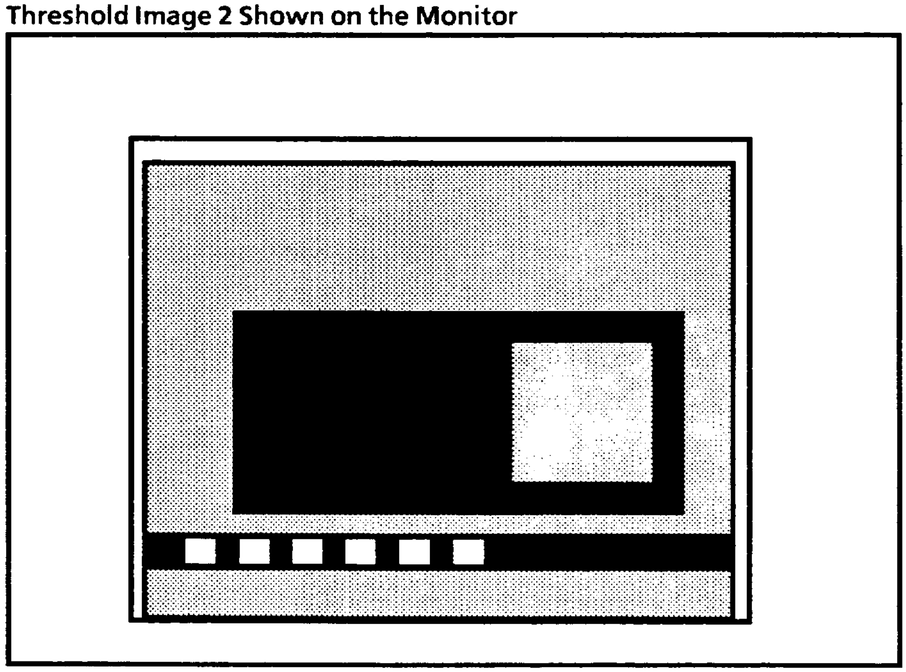
Your Action	Comments
Pick the image number *icon to select image number 2.*	Picking the *image number* icon again changes the number to "2." With "2" selected, threshold image 2 is shown on the monitor.

Threshold Menu



Change the threshold number so that the threshold image matches Figure 6.6.	Using the *threshold up* or *down* icon as necessary, change the threshold number of threshold image 2 until the image on the monitor matches Figure 6.6. **Hint:** Start by setting the threshold number to about 30 units greater than the number set for image 1. Then adjust the number up or down until the image matches that shown in Figure 6.6.

Figure 6.6
Threshold Image 2 Shown on the Monitor



Using Setup Menu Icons (continued)

Note that *threshold image 1* (Figure 6.5) shows only the black spot. This is because the threshold is set so that only the very dark pixels in the spot have a gray level lower than the threshold number for threshold image 1. Thus threshold image 1 has effectively isolated this one feature from all the rest of the target pattern. Selecting threshold image 1 for a given gage would allow you to measure attributes of this single dark object.

Also, note that *threshold image 2* (Figure 6.6) shows a white square surrounded by the larger black rectangle. This is because the threshold is set so that both the dark spot, and the medium gray rectangle, have gray levels lower than the threshold number set for threshold image 2. The light gray square has a gray level greater than the threshold number. Thus threshold image 2 has effectively highlighted two features, the larger dark rectangle, and the smaller bright square inside the rectangle. Selecting threshold image 2 for a given gage would allow you to measure attributes of *both* the dark rectangle, and the bright square.

With threshold images 1 and 2 set as described above, return to the *Setup* menu.

Your Action	Comments
Pick the OK icon to return to the Setup menu.	Picking the *OK* icon returns you to the previous menu (that is, the *Setup* menu). **Note:** The threshold image you last selected remains on screen while the VIM2 module is in setup mode.

Points to Remember When Using the Menus and Icons

- Picking the *OK* icon returns you to the previous menu.
- Picking the *ETC* icon displays additional icons which are part of the same menu branch.
- Picking any icon with an *angular cut* on its lower right corner displays a subsequent menu for working with the function indicated by that icon.

This chapter has introduced you to the basics of using the icon interface. The remaining chapters describe in detail the specific functions and operations of each of the icon menus in the VIM2 module user interface.

(This page intentionally left blank)

Chapter 7 Operating Environment

Chapter Objectives

The objective of this chapter is to show how to use the *Environment* menu icons to configure the VIM2 module operating environment. For the most part, the operating environment parameters define the relationship between the VIM2 module and external host devices and trigger sources.

Note: For more information on the interfacing with the VIM2 module, see Chapters 14 and 15. See Chapter 14 for more information regarding the PLC/ VIM2 module interface. See Chapter 15 regarding the RS-485 port interface on the VIM2 module.

Operating Environment

Using the *Environment* menu icons, you can select the following operating environment parameters:

Master Host or Standalone Mode: Select whether the VIM2 module operates as a standalone unit, or is controlled by a master host – either a PLC, or a host device connected to the RS-485 serial port.

Note: The RS-485 port on the VIM2 module is point-to-point, half-duplex only. See Chapter 15 for more information on the RS-485 interface.

Enable serial host/protocol: Enable VIM2 module communication to a host device connected to the RS-485 serial port, and select either DF1 or ASCII protocol for serial communication.

Select baud rate: Select the baud rate for communication with a host device connected to the RS-485 port.

Select run mode trigger: Select run mode trigger source – either the VIM2 module's swingarm trigger input, or the trigger signal from the host.

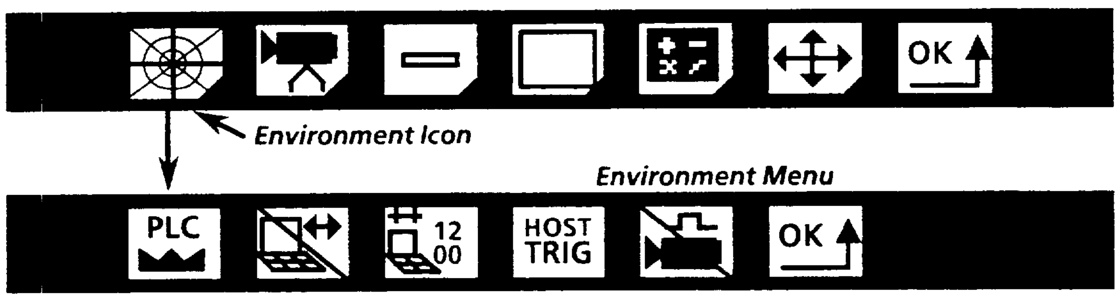
Select setup mode trigger: Select triggering to be used during setup – either automatic internal triggering and image acquisition, or the run mode trigger (either host or swingarm trigger source, whichever has been selected).

Accessing the Environment Menu

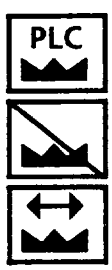
To begin to set up the operating environment, first access the the *Environment* menu as described below.

Your Action	*Comments*
Pick the Environment *icon on the* Setup *menu to access the* Environment *menu.*	If the *Main* menu is displayed, first access the *Setup* menu by picking the *Setup* icon (the "open padlock"). Then, with the *Setup* menu displayed, pick the *Environment* icon.

Setup Menu

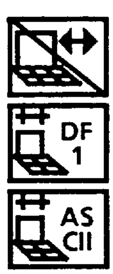


Here are brief descriptions of the *Environment* menu icons:



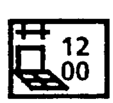
Master Host / Standalone – Pick this icon to select the *standalone* mode, or to select a master host (PLC or the device connected to the RS-485 port). This icon, when picked, toggles between the *PLC host* icon and the *Standalone* icon (indicated by diagonal slash across the icon).

The *third* icon, the *Serial host* icon, is available also, *if* the VIM2 module's RS-485 port is enabled (see the next icon).

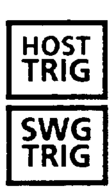


Serial Enable – Pick this icon to *enable* the VIM2 module's RS-485 port, and to *select the protocol* (*DF1* or *ASCII)* you will use in RS-485 communications.

This icon, when picked repeatedly, cycles through three selections: 1) RS-485 port *disabled* (indicated by diagonal slash across the icon), 2) RS-485 port *enabled*, using *DF1* protocol, and 3) RS-485 port *enabled*, using *ASCII* protocol.

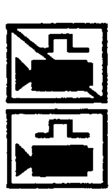


Baud Rate – When using the RS-485 port, pick this icon to select the baud rate for RS-485 communications, from a choice of 1200, 2400, 4800, or 9600.



Run Mode Trigger – Pick this icon to select the trigger source for VIM2 module inspections.

This icon, when picked, toggles between **HOST TRIG** (*host triggered)* and **SWG TRIG** (*swingarm triggered).*



Setup Trigger – Pick this icon to select the trigger source during setup. This icon, when picked, toggles between two choices; select either *internal trigger* (indicated by diagonal slash across the icon) or *Run Mode trigger* (*host* or *swingarm* triggered, as selected by the *Run Mode Trigger* icon).

Master Host vs. Standalone Mode

Select the *Master Host* or *Standalone* mode according to whether the VIM2 will be operated as a standalone unit, or a PLC or other device has control as the master host.

Note that you can install the VIM2 with the PLC as a host, and/or connect a host device to the VIM2 module through the RS-485 port. Where two host devices are used, only one can act as the master host *at any one time*; you can, in this case, select the master host using the *Master Host / Standalone* icon. The master host is exclusively permitted to perform certain functions, such as downloading configurations to the VIM2 module, or triggering inspections.

You can also choose to have *no* device serve as *master host* by selecting *standalone* mode, even though one or two host devices are used. In this case, no device operates as master host, although a subset of the master host functions can be performed by the PLC or serial device, such as collection of configuration or results information (for more specific command information for either the PLC and serial host, refer to Chapters 14 and 15, respectively).

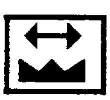
To select a *master host* or *standalone* mode, pick the *Master Host / Standalone* icon.

Master Host / Standalone Icon

Your Action	Comments
Pick the Master Host / Standalone *icon as necessary to select the appropriate mode.*	Here are descriptions of the *Master Host / Standalone* icon options:

PLC Host: Select this icon if you are configuring the VIM2 system with the PLC as the master host to the VIM2 module.

Standalone: Select this icon if you are configuring the VIM2 system without a PLC or other host device connected to the VIM2.



Serial Host: Select this icon if you are configuring the VIM2 system with device connected to the VIM2 RS-485 lines as the master host to the VIM2 module.

Note: The *Serial Host* icon is not available unless you first *enable* the RS-485 communication using the *Serial Enable* icon (described on the following page).

Enabling the RS-485 Serial Port

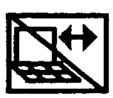
The VIM2 module can communicate to a device through the RS-485 communication port, which is available through the VIM2's swingarm interface. A device connected to the RS-485 port must communicate with the VIM2 module using one of two protocols – DF1 or ASCII.

Note: For more information regarding use of these protocols, see Chapter 15. See also Appendix E for a description of DF1 protocol.

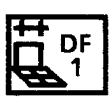
Use the *Serial Enable* icon to select whether or not the VIM2 module will communicate to a host device connected to the serial port, and to select either DF1 or ASCII protocol for serial communciation.

Serial Enable Icon ↗ ↖ Baud Rate Icon

Your Action	Comments
Pick the Serial Enable *icon as necessary to select the appropriate mode.*	Here are descriptions of the *Serial Enable* icon options:



Serial Disabled: Select this mode if you are not connecting any device to the VIM2 RS-485 lines. **Note:** You cannot select this icon if the serial host has been selected as the master host (see previous page).



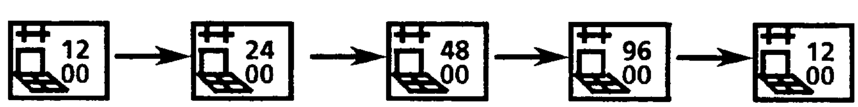
Enabled / DF1: Select this mode, if you are connecting a device to the VIM2 RS-485 lines, and if you intend to use DF1 protocol to communicate with the VIM2 module.

Enabled / ASCII: Select this mode, if you are connecting a device to the VIM2 RS-485 lines, and if you intend to use ASCII protocol to communicate with the VIM2 module.

Selecting the Baud Rate

If you are connecting a device to the RS-485 port of the VIM2 module, pick the *baud rate* icon to select the baud rate for RS-485 communications.

Pick the Baud Rate *icon as necessary to set the baud rate for communication.*	Picking this icon repeatedly cycles through the baud rate options – 1200, 2400, 4800, or 9600.



Selecting Run Mode Trigger

The VIM2 module requires a trigger signal to initiate inspections in *run* mode. You can supply the trigger signal to VIM2 module by either of two methods:

- Host trigger – communication from the master host (either PLC or a device connected to the RS-485 port).
- Swingarm trigger – direct hardware interface with the swingarm trigger input terminal.

If no master host is used, you must use the swingarm trigger method to trigger the VIM2 module. If you *are* using a master host with your VIM2 module, you can use either the host or the swingarm trigger method, but not both.

Select the trigger source by picking the *Run Mode Trigger* icon, for the trigger method you intend to use.

Run Mode Trigger Icon

Your Action	*Comments*
Pick the Run Mode Trigger *icon as necessary to select the appropriate mode.*	When you pick the *Run Mode Trigger* icon, it toggles between two options – *Swingarm trigger* and *Host trigger*. Here are descriptions of the two options:
HOST TRIG	**Host Trigger** – Select this icon if you intend to trigger the VIM2 module through communication from the host device.
SWG TRIG	**Swingarm Trigger** – Select this icon if you intend to trigger the VIM2 module through the swingarm terminal.
	Note: If you select *Host Trigger*, the VIM2 module ignores the swingarm trigger (however, the *SWG Trigger* LED is always activated when you supply a trigger signal to the swingarm). If you select *Swingarm Trigger*, the VIM2 module ignores any attempt to trigger through communication from the (master) host.

Selecting the Setup Trigger

When the VIM2 module is in *setup* mode and is being configured, the module must at times acquire a video image for analysis from the video camera, in order to complete the setup procedure in progress. For example, whenever you attempt change the threshold level of a threshold image, or change the *acceptance range* of a line gage or window, the VIM2 module must acquire an image for analysis.

You can select one of two methods for image acquisition during setup:

- Internal trigger – when the VIM2 needs to acquire an image during setup, it will do so automatically, without a trigger signal.
- Run Mode trigger – when the VIM2 needs to acquire an image during setup, it will do so only upon receiving the trigger signal from the selected *run mode trigger.*

Use the *run mode trigger* setting to synchronize the image acquisition to an external trigger source during setup; this is especially valuable if you preparing to use strobe lighting, for example, or if the workpiece is positioned periodically in the workstage.

Select the *setup trigger* by picking the *Setup Trigger* icon.

Setup Trigger Icon

Your Action	Comments
Pick the Setup Trigger *icon as necessary to select the appropriate mode.*	Here are descriptions of the *Setup Trigger* icon options:
	**Internal Trigger** – Select this icon if you *do not* intend to use the *run mode trigger* during setup; the VIM2 will automatically acquire an image as needed.
	**Run Mode Trigger** – Select this icon if you intend to use the *run mode trigger* during setup; the VIM2 will wait for a trigger signal to acquire an image.

Selecting the Setup Trigger *(continued)*

Note: If, during setup, the VIM2 *Setup Trigger* icon is set for *run mode trigger*, and an image is required, the VIM2 module will wait for a trigger signal. If no trigger signal is received within one second, this image is displayed across the monitor screen, while the VIM2 awaits a signal:

To complete the procedure in progress, supply a trigger signal (from the host or through the swingarm, as configured). If this is not possible, and you wish to continue your configuration process, depress the tip of the light pen for about one second. The image acquisition occurs, and then the process in progress is completed.

(This page intentionally left blank)

Chapter 8 Image Acquisition and Brightness Compensation

Chapter Objectives

The objective of this chapter is to show how to use the *Image* menu icons to set up the VIM2 module's brightness compensation and other image acquisition capabilities. This chapter also decribes using the *Registration* menu icons to automatically adjust threshold image settings.

Note: Before attempting to set the image acquisition parameters described in this chapter, you should first set up the workstage to establish the lighting and workpiece positioning during inspection, and establish the camera focus and F-stop setting. For information on the staging for VIM2 vision inspections, see Chapter 4.

Image Acquisition and Brightness Compensation

Using the *Image* and *Registration* menu icons, you can set the following parameters:

Brightness Compensation: Set up the VIM2 module's brightness compensation capability to account for lighting brightness variations. First select from two modes – mode 1 (*same frame* compensation) or mode 2 (*next frame* compensation). Then place the brightness probe on the monitor screen, and set an *acceptance range* for the brightness probe.

Strobe Enable: Set the VIM2 module to utilize the strobe light during setup and/or run mode.

Threshold Images: Set the threshold value for the threshold images you intend to use during inspection.

Halted Image: Set the VIM2 module to utilize a halted or "live" video image during setup.

Brightness Registration: Automatically adjust the threshold image settings to a new nominal brightness level.

Accessing the Image Menu

To begin to set up the image acquisition and brightness parameters, first access the *Image* menu as described below.

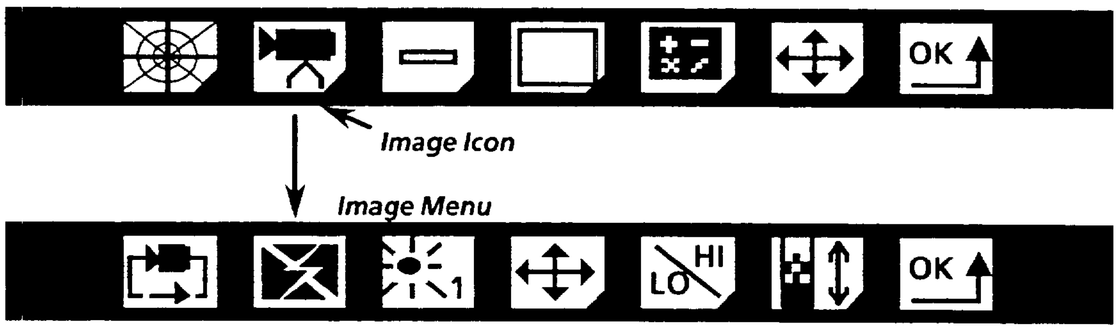
Your Action	Comments
Pick the Setup *icon on the* Main *menu to access the* Setup *menu.*	If the *Main* menu is displayed, access the *Setup* menu by picking the *Setup* icon (the "open padlock"). If the *Setup* menu is already displayed (as shown below), skip this step.

Setup Menu

Accessing the Image Menu
(continued)

Your Action	**Comments**
Pick the Image *icon on the* Setup *menu to access the* Image *Menu.*	Picking the *Image* icon causes the *Image* menu to appear. Notice that a *small rectangular window* also appears on the screen; this is the *brightness probe.*

Setup Menu



Here are brief descriptions of the *Image* menu icons:

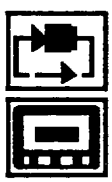
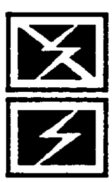


Image Acquisition – Pick this icon to select the *image acquisition mode* to be used by the VIM2 module during setup. This icon, when picked, toggles between two modes – *image acquisition*, and *halt image.*

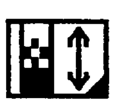


Strobe – Pick this icon to enable or disable the *strobe* capability of the VIM2 module. This icon, when picked, toggles between *enabled* or *disabled* (*disabled* is indicated by diagonal slash across the icon).

Mode Select – Pick this icon to select the *brightness compensation* mode. This icon, when picked, toggles between mode 1 (*same frame* compensation) and mode 2 (*next frame* compensation).

Probe Move – Pick this icon to access the *Probe Move* menu. Use the *Probe Move* menu to position the brightness probe on the monitor image.

Acceptance Range – Pick this icon to access the *Acceptance Range* menu. Use the *Acceptance Range* menu to set the upper and lower limits for the brightness probe, and to read the current probe result, and probe statistics.



Threshold – Pick this icon to access the *Threshold* menu. Use the *Threshold* menu to set the threshold level for any or all of the four threshold images.

Brightness Compensation

The VIM2 module has a brightness compensation capability which allows the VIM2 module to automatically adjust for changes in lighting intensity from inspection to inspection.

What does brightness compensation do?
Each time the VIM2 module conducts an inspection, it analyzes the threshold images with the vision tools which have been configured. The composition of these threshold images can change if the lighting intensity changes – that is, if lighting becomes dimmer, more black pixels may appear in the images; if the lighting becomes brighter, more white pixels may appear in the images. If this happens, the measurement accuracy of line gages and windows can be affected; this can cause mistaken *reject* or *accept* decisions.

When you use brightness compensation, if a change in lighting brightness occurs in the workstage area (for example, a light bulb fails), the VIM2 module automatically adjusts threshold levels to account for the change, so that the threshold images are relatively unaffected. The result is that VIM2 measurements are relatively unaffected.

How do you set up brightness compensation?
During setup, you place the *brightness probe* – a special type of window – in the video image. This brightness probe measures the average level of brightness in a small area of the gray-level image from the camera. During setup, the brightness probe learns the nominal brightness level in the image; the range for measurement values is a scale from 0 (completely dark) to 255 (very bright). During VIM2 operation, the probe detects the brightness level for each inspection, and compares this to the nominal level. Then, if any change in lighting is detected, the threshold levels of the threshold images are changed also.

When setting up the brightness probe, you also set an *acceptance range* for brightness. If the measured brightness value is beyond this preset range, an image *acquisition error* occurs. This allows you to limit the amount of acceptable lightning variation, and help distinguish between inspection failures due to bad parts, and those due to bad lighting.

Which threshold images are affected by brightness compensation?
Any or all of the threshold images can be set to use the brightness compensation capability; you *enable* or *disable* brightness compensation for each threshold image using the *Anchor / Float* icon on the *Threshold* menu.

Is brightness compensation the same for strobe lighting and non-strobe lighting?
No, there are two different brightness compensation modes –

Brightness Compensation *(continued)*

mode 1, for strobe or non-strobe lighting (*same frame* compensation), and mode 2, for non-strobe lighting only (*next frame* compensation). You select the mode using the *Compensation Mode* icon.

When do you set up brightness compensation?
Set up brightness compensation *after* you have set up the workstage, and have a workpiece in position (see Chapter 4), so that the workpiece features to be inspected are shown clearly in the field of view (and on the monitor).

Selecting the Compensation Mode

When setting up brightness compensation for your application, you must select the *compensation mode.* You can choose one of two modes – mode 1, (*same frame* compensation), or mode 2 (*next frame* compensation). You select the mode using the *Compensation Mode* icon on the *Image* menu.

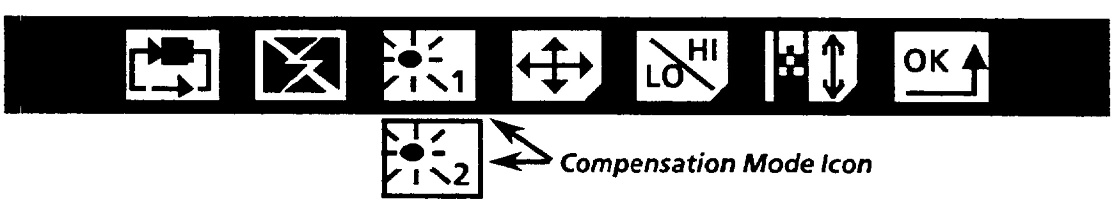
When *compensation mode 1* is selected, the brightness compensation is applied to the same image (the *same frame*) used to detect the lighting brightness. Thus it is ideal for strobe-lit inspections, although it can also be used in non-stobe lighting applications.

When using *compensation mode 2*, the brightness compensation is applied to the subsequent acquired image (the *next frame*). This mode is more suited to non-strobe light sources; *do not use mode 2 when using strobe lighting.*

Note: If you select *mode 1*, you can only place the brightness probe in the top 40 rows of the monitor image. And, you cannot place any part of a line gage or window in the top 48 rows of the monitor image.

Your Action	**Comments**
Pick the Compensation Mode *icon on the* Image *menu to set the* Compensation Mode.	The *Compensation Mode* icon is the icon third from the left on the *Image* menu. The *Compensation Mode* icon, when picked, toggles between mode 1 and mode 2.

Image Menu



Note: The VIM2 module does not allow you to select mode 1 if the brightness probe is placed somewhere below the top 40 rows of the monitor image, or if any window or line gage is positioned in the top 48 rows of the image.

Placing the Brightness Probe

In order to use a brightness probe for lighting brightness compensation, you must *place* that brightness probe in a suitable position in the monitor image.

Generally speaking, you should place the probe so that it window covers a *reference patch* in the video image.

What is a "reference patch?"
A reference patch is an area of consistent color and shading which you should place in the camera field of view when you are setting up the workstage, if you intend to use brightness compensation. The reference patch is necessary to assure that any detected change in brightness is due to variations in the lighting, and not due to changes of shading of objects in the workstage. When the brightness probe is used, any change in brightness from inspection to inspection is detected by the probe as a change in lighting. The purpose of the reference patch is to supply an area in the monitor image whose shading will not change from inspection to inspection, so that the brightness probe detects only a brightness change which is due to change in lighting intensity.

An example procedure in this section illustrates the placement of the brightness probe over a reference patch.

How do you position the reference patch?
The position of the reference patch should be fixed in the workstage area in relation to the camera, so that it appears in the same place in each inspection image. Also, the reference patch should be placed so the position of the workpiece will not obstruct the camera's view of the reference patch during inspection. This helps assure that the brightness probe readings will remain effective.

Note: If you are using *brightness compensation mode 1*, you must position the reference patch so that it appears in the top 40 rows of the monitor image.

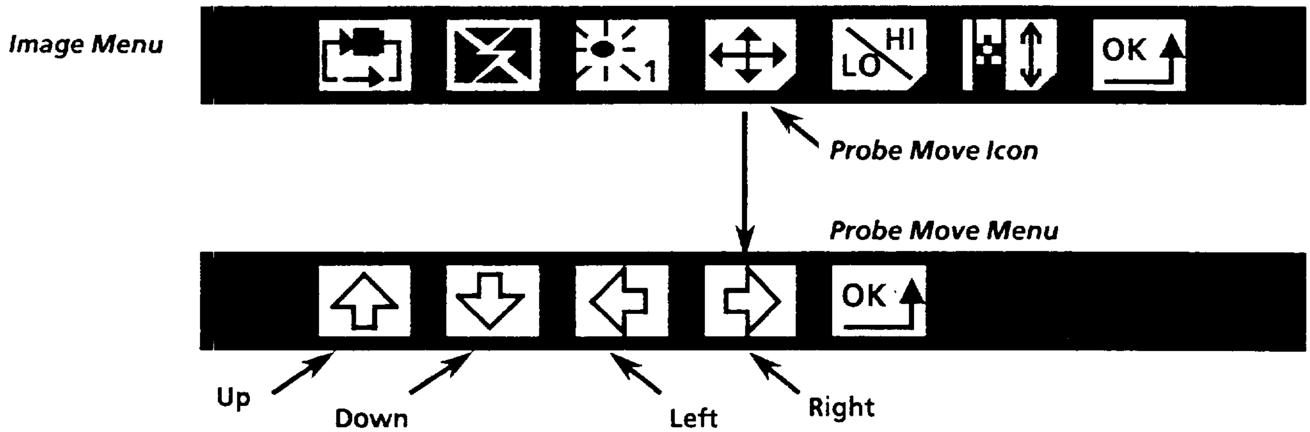
What shade should the reference patch be?
The appropriate shade of the reference patch will depend on the brightness of the lighting in the workplace. The shade of the reference patch should be such that, when the brightness probe is placed over it, a midrange reading results (the overall brightness range is 0 to 255; midrange would be approximately 70-180). This allows brightness compensation in either direction (for brighter or darker brightness changes).

Note: When using backlighting, it may be necessary to place a patch of somewhat transparent, yet shaded material over part of the backlit area to obtain a midrange reading.

Placing the Brightness Probe
(continued)

To place the brightness probe you need to use the *Probe Move* menu (**Note:** You cannot adjust size or shape of the probe). Pick the *Probe Move* icon to access the *Probe Move* menu, as described below:

Your Action	*Comments*
Pick the Probe Move *icon to access the* Probe Move *menu.*	The *Probe Move* icon is the fourth icon from the left on the Image menu (indicated by the four directional arrows).



The *Probe Move* menu includes four directional arrow icons. To *move* the probe into the desired position, pick the appropriate arrow icon(s), in one of two ways:

- *Pick* the arrow icon to move the probe one pixel in the direction indicated by the icon.
- *Pick and hold* the light pen down against the arrow icon to cause *continuous movement* in the indicated direction (*if you pick an arrow icon and it flashes*, this indicates that the window is at a boundary - such as the screen edge -and cannot be moved further in that direction).

  Note: Depending on the *brightness compensation* mode you have selected, placement of the probe may be restricted - if you have selected mode 1(*same frame* compensation), you can only place the probe in the top 40 rows (or about the top 17%) of the video image.

Placing the Brightness Probe (continued)

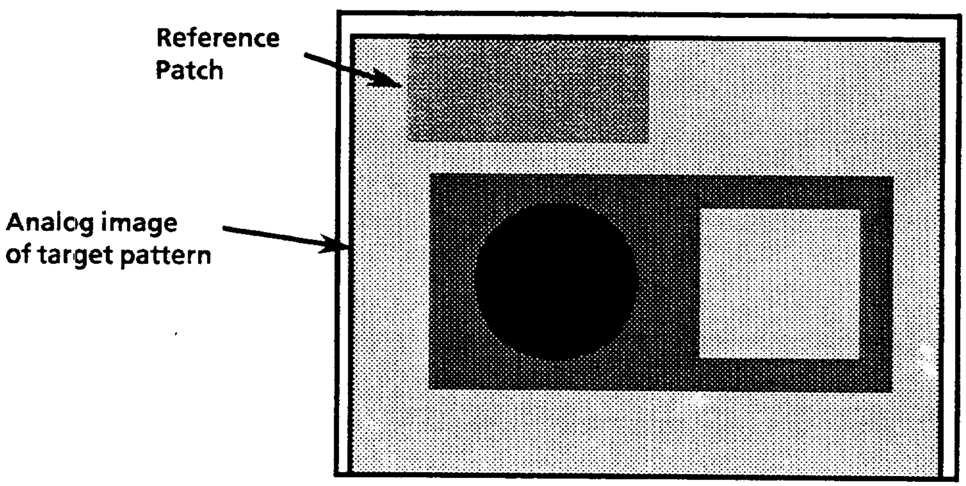
Example Probe Move

This demonstration places the brightness probe over the reference patch on the monitor image.

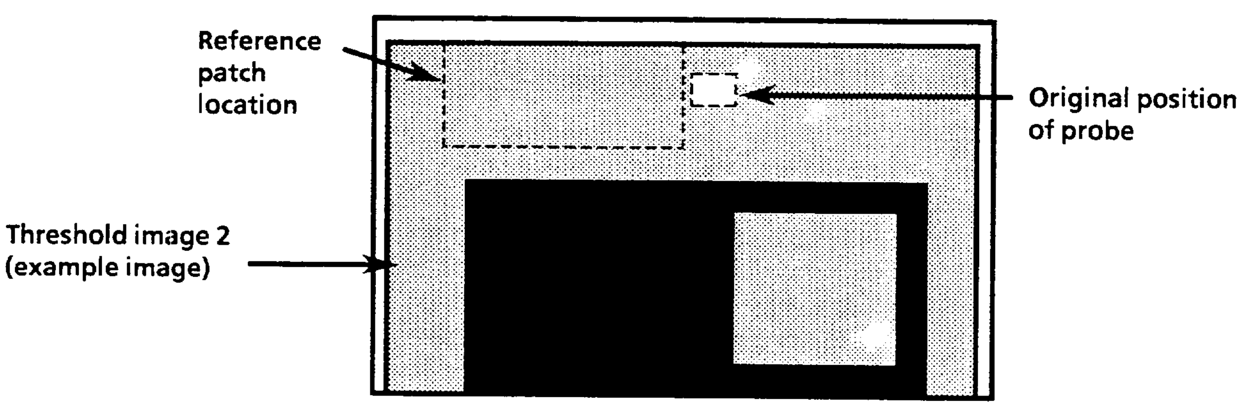
Note: For example purposes, these procedures refer to the target pattern included with this manual. See Chapters 5 and 6 to set up the VIM2 module, peripheral equipment, and target pattern.

In order to locate the reference patch in the monitor image, you may first need to briefly change the video display mode, so the monitor displays the analog image, instead of the threshold image.

Your Action	*Comments*
Pick the upper half of the monitor screen twice.	Picking the screen twice with the light pen displays the analog image (the probe is not displayed on screen).
Observe and remember the reference patch *position.*	The reference patch is the light gray rectangle in the upper half of the analog image (see illustration below).



Pick the upper half of the monitor screen once.	Picking the screen once redisplays the threshold image, the *Probe Move* menu, and the brightness probe.

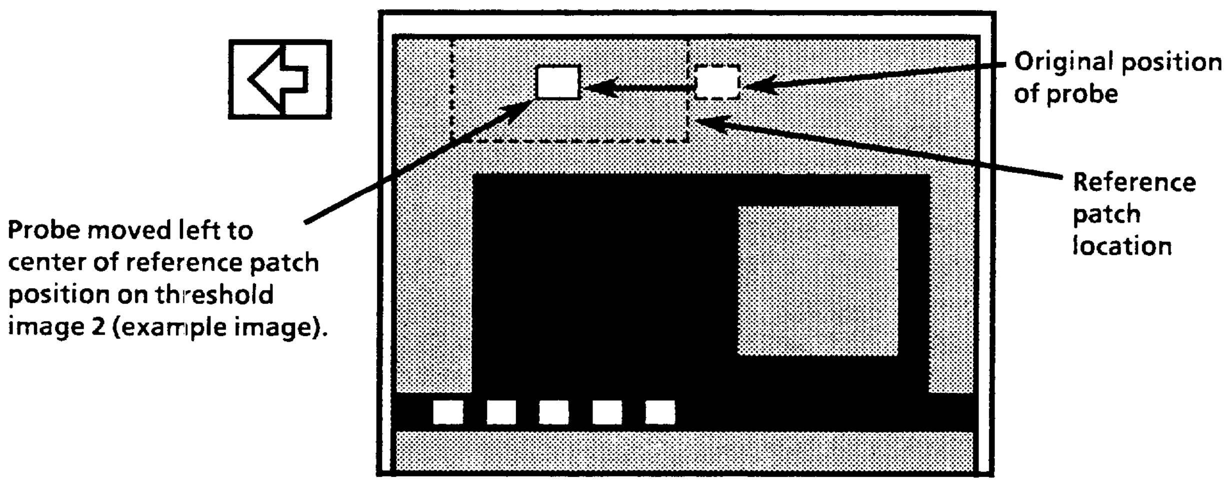


Placing the Brightness Probe (continued)

Example Probe Move *(continued)*

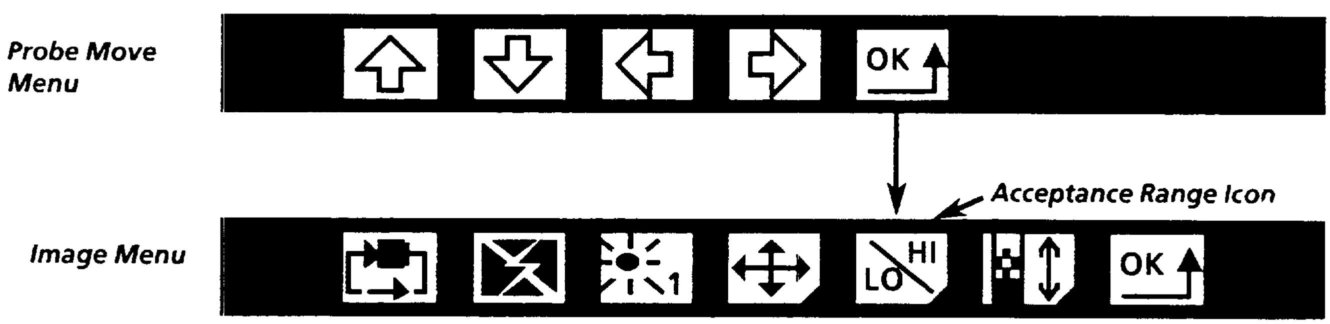
Pick the left *arrow until the probe is placed over the reference patch.*

Picking the *left* arrow moves the probe to the left.



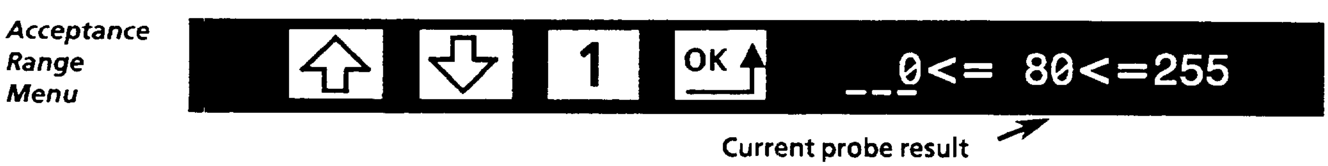
Pick the OK *icon.*

Once the brightness probe has been placed, you can return to the *Image* menu using the *OK* icon.



Pick the Acceptance Range *icon to access the* Acceptance Range *menu.*

The current measured result of the brightness probe is listed as the *middle number* of the three numbers on the right side of the *Acceptance Range* menu.



Observe the probe result.

Ideally, the probe result should be a midrange reading (the range is 0-255, midrange is approximately 70-180). **Note:** If the reading is not in midrange during setup, you may have to adjust either or both the workstage lighting or the F-stop setting of the lens (see Chapter 4). Or, if the lighting and F-stop setting are satisfactory, you can change the reference patch to a different shade to get a midrange reading.

Setting the Acceptance Range

Once you have placed the brightness probe, the next step is to set the *acceptance range* of the brightness probe.

What is the acceptance range?
When you set up you the brightness probe, you must set the range of acceptable measurement values. That is, you set an *upper* limit and a *lower* limit for the brightness probe results. When you set the acceptance range limits, you determine the amount of lighting variation allowed during inspections.

How does the acceptance range work?
When the VIM2 performs an inspection, the *brightness probe* returns results based on the lighting brightness detected; if the brightness probe results do not come within the upper and lower limits (or *acceptance range*), an *acquisition error* is generated; the inspection fails, and the *Acq Error* LED goes *On*. The VIM2 module indicates a *Reject* decision through the swingarm, the *Decision* LED, and the discrete bit, as well as through the results data blocks.

What is the acceptance range used for?
In each application, there is a point of lighting change beyond which the brightness compensation of the VIM2 is no longer effective; the threshold images remain distorted despite the brightness compensation. The acceptance range can be used to limit the allowable lighting variation, so that if this limit is surpassed, an *acquisition error* is generated. Thus you can use the acceptance range, and the *acquisition error* signal, to help distinguish whether failed inspections are the result of bad parts, or the result of bad lighting.

You can use the acceptance range limit to prevent inspections from passing when the lighting is inadequate. You can also use the *acquisition error* signal as an alarm indicating the lighting for the inspection is no longer adequate, and the light source must be adjusted or replaced.

How do you determine the acceptance range limits?
You can set the acceptance range limits to detect when lighting has failed entirely. Do this by observing brightness probe result with lighting at optimal level, then observing brightness probe result with lighting removed. Set the acceptance range limit so that the reading for the removed lighting is out of range.

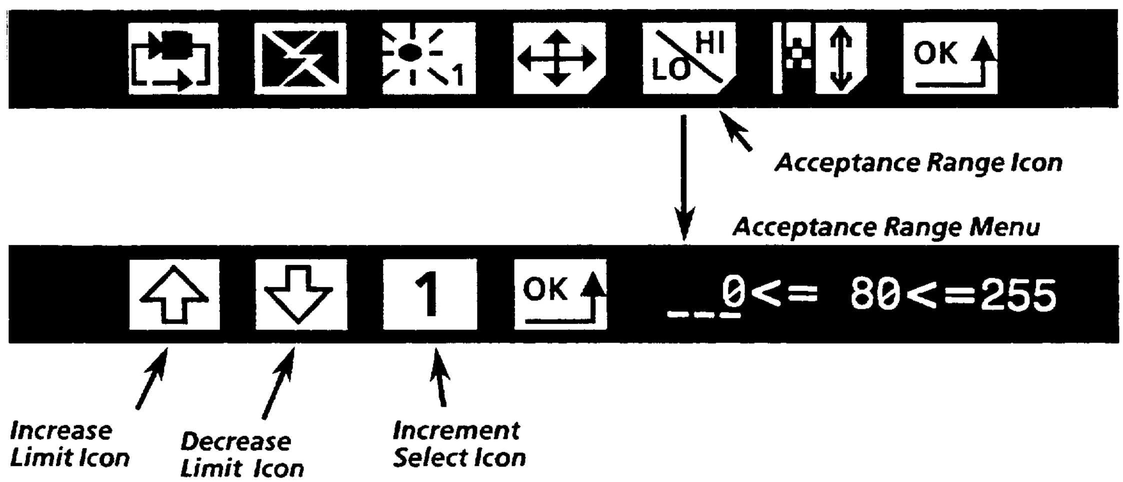
Or, you can experiment with different lighting levels during trial inspections with known good parts, to see how much lighting variance is necessary to begin to cause failures. You can then observe the brightness probe reading at the point where failures begin to occur due to lighting variance, and set acceptance range limits accordingly.

Setting the Acceptance Range (continued)

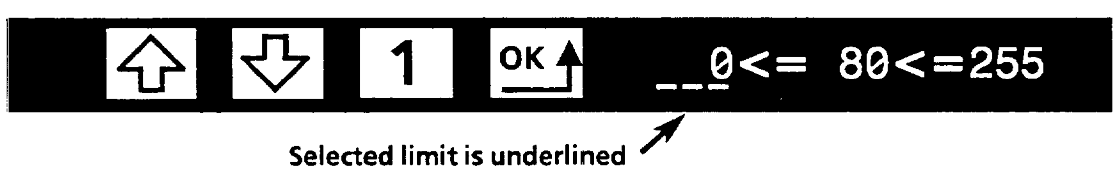
To set the acceptance range for the brightness probe, access the *Acceptance Range* menu.

Your Action	Comment
Pick the Acceptance Range *icon on the* Image *menu.*	

Image Menu



Your Action	Comment
Observe the probe result.	On the right side of the *Acceptance Range* menu are three numbers, separated by "< =" signs. The *middle* number is the current measured *result* of the brightness probe.
*Pick the limit (*upper or lower*) you want to change.*	The *lower limit* is the *left* number of the three numbers displayed on the right side of the menu, the *upper limit* is the number on the *right* (**Note:** Default values are displayed for the limits initially). To select either limit, pick that limit with the light pen. The selected limit is *underlined.*



Your Action	Comment
Pick the Increment Select *icon, if necessary, to change the increment.*	The *Increment Select* icon sets the amount by which the selected limit will change when you pick the *Increase Limit* or *Decrease Limit* icon. When picked, the *Increment Select* icon toggles between 1 and 10.
Pick the Increase Limit *icon or the* Decrease Limit *icon, as necessary, to increase or decrease the selected limit.*	To *increase* the selected limit by the amount shown on the *Increment Select* icon, pick the *Increase Limit* icon. To *decrease* the selected limit, pick the *Decrease Limit* icon. To change the limit value rapidly, pick and hold the light pen tip against the icon.

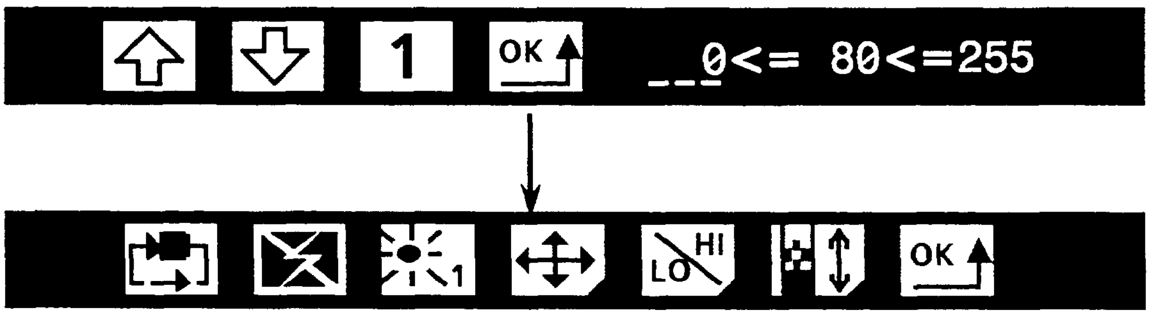
Setting the Acceptance Range (continued)

Note: The VIM2 module does not allow you to set the upper limit to a value lower than the lower limit, or to set the lower limit to a value higher than the upper limit.

Your Action	***Comment***
Pick the OK *icon to set the nominal brightness level and go to the* Image *menu.*	When you pick the *OK* icon on the *Acceptance Range* menu, the VIM2 *samples* the brightness level, and *sets the nominal brightness level* based on the sample. This setting is the value used for comparison during run mode, when lighting brightness compensation is in effect.

Acceptance Range Menu

Image Menu



Using the Learn Mode and other Run Mode Functions

The VIM2 module features a variation of its *run* mode called the *learn* mode. In *learn mode,* the VIM2 accumulates *statistics for each enabled inspection tool, and the brightness probe,* based on a number of trial inspections. Use of the *learn* mode can be helpful in determining the optimal *acceptance range* for inspection tools and probe.

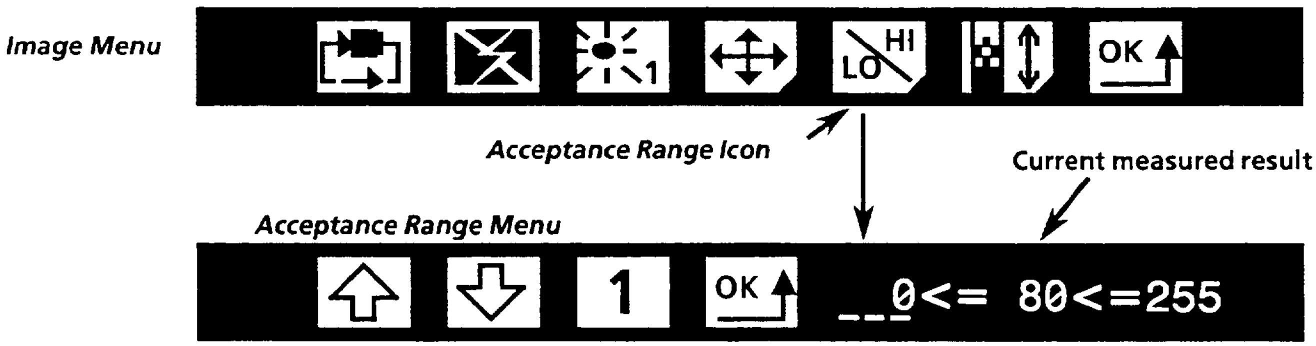
The VIM2 module also features a number of run mode *arm* options, which can be helpful in refining your configuration. See Chapter 13 for more information regarding use of the *learn* mode and other *run* mode options.

Accessing Statistics

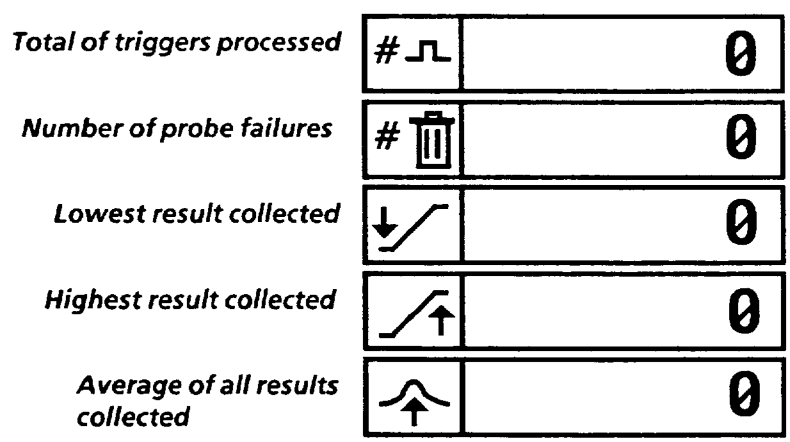
When you run the VIM2 in *learn mode,* you accumulate statistics for all enabled visions tools and probes. To observe the collected statistics for the brightness probe, first access the *Acceptance Range* menu on the *Image* menu, as described on the next page.

Accessing Statistics
(continued)

Your Action	*Comment*
Pick the Acceptance Range *icon on the* Image *menu.*	



Pick the current measured result.	On the right side of the *Acceptance Range* menu are three numbers, separated by "< =" signs. The *middle* number is the *current measured result* of the brightness probe. When you pick this number, the brightness probe disappears, and the statistics are displayed (see diagram below).



Note: The results of *failed* inspections for the probe are not included in the *lowest, highest*, and *average* values. These statistics are based on passing inspections only. Therefore, for preliminary trial runs, you may want to set the acceptance range to extreme limits initially, for the purpose of collecting statistics for all inspections.

Pick the current measured result again to remove the statistics display.	Picking the *current measured result* again redisplays the brightness probe and removes the statistics display.

Strobe Enable

The VIM2 module can operate using a strobe light as the light source. If you are going to use strobe lighting, either during setup mode or during run mode, you must first enable the VIM2 module's strobe capability using the *Strobe* icon.

Note: If you are using the strobe lighting during setup, see Chapter 5 for more information regarding the strobe light signal output on the VIM2 swingarm.

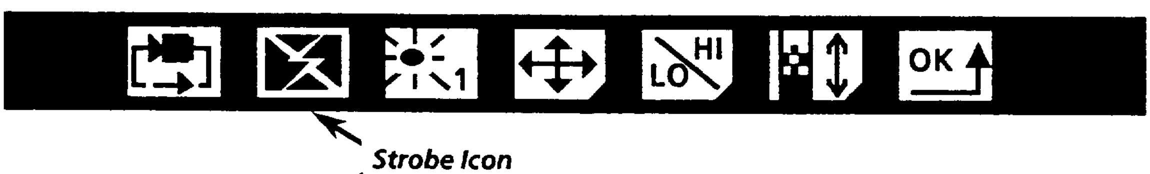
With *Strobe enabled* during setup mode, the monitor image appears frozen. Each time an image acquisition is required, such as after moving the brightness probe, the strobe output on the VIM2 module's swingarm is energized just before the image is acquired.

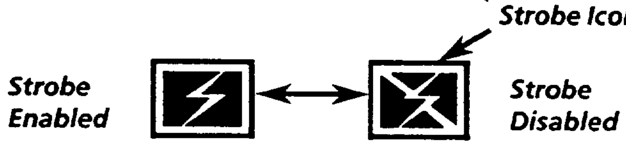
With *Strobe enabled* during run mode, the strobe output is energized just before the image is acquired, and the monitor image appears frozen after each inspection.

To enable operation using the strobe lighting, pick the *Strobe* icon.

Your Action	***Comment***
Pick the Strobe *icon on the* Image *menu to* enable *strobe operation.*	When picked, the *Strobe* icon toggles between *enabled* and *disabled* (*disabled* is indicated by the diagonal slash across the icon). With the *Strobe enabled*, the monitor image appears "frozen," but images are acquired by the VIM2 module as needed. When image acquisition takes place, the VIM2 module activates the strobe trigger signal on the swingarm (see Chapter 5 for information on strobe trigger connections).

Image Menu





Pick the Strobe *icon icon on the* Image *menu to* disable *strobe operation.*	With the *Strobe disabled*, the VIM2 module resumes normal "live" video display.

Threshold Images

The VIM2 module produces four threshold images for use with line gages and windows. You can use one, or up to all four threshold images in your configuration. Each line gage or window is assigned a threshold image to inspect; each tool inspects only the threshold image to which it is assigned.

What is a threshold image?
Recall from Chapter 3 that a threshold image is a high-contrast image consisting of pixels of only two colors – black and white. The VIM2 module allows you to set up a different *threshold level* for each of the four different threshold images it produces with each inspection.

What is a threshold level?
The threshold level determines which pixels will be black, and which will be white, in the threshold image. Recall from Chapter 3 that the image from the video camera contains an array of pixels, each of whose brightness levels, or "gray-levels," can vary along a range of values from 0-255. The value "0" indicates black, and "255" indicates white. The values from 1-254 represent the range of grays in between; the lower the number, the darker the gray.

In creating a threshold image from this gray-level image, the VIM2 module converts each pixel to one of two colors – black or white. To accomplish this, the VIM2 uses a "threshold level," which you must supply. This threshold level is the dividing line that determines which pixels are converted to black, and which are converted to white. All pixels having gray-level values below or equal to the threshold (the darker pixels) are converted to black, and those above (the brighter pixels) are converted to white. The result is a high-contrast "threshold" image with only black and white pixels.

Changing the threshold level setting can alter the image substantially. When the threshold is increased, the image becomes darker as more pixels fall below the threshold, and and are converted to black. When the threshold is decreased, the image becomes lighter, as more pixels are above the threshold, and are converted to white.

Why are different threshold levels used?
Varying the threshold can have the effect of enhancing certain image features, and/or detracting from the appearance of certain image features. The ability to vary the threshold allows you to enhance the features of interest on the workpiece you are inspecting.

The VIM2 module has four available threshold images. You must set up *at least one* threshold image. You can set up each of the four threshold images, each with a different threshold level, with each highlighting different features.

Threshold Images
(continued)

Or you can set up the threshold images with some high-lighting the same features. You *do not necessarily have to set up and use all four of the threshold images*; set up only as many of the threshold images as you need to use to highlight the different features of interest on the workpiece.

How are threshold images referenced?
Each time you configure a line gage, you assign a threshold image for that line gage; the line gage inspects only the threshold image assigned. When you configure any window, 1-4, each window that you configure is automatically assigned the threshold image of the same number (see Chapters 9 and 10 for line gage and window information).

When do you set up a threshold image?
Set up a threshold image:

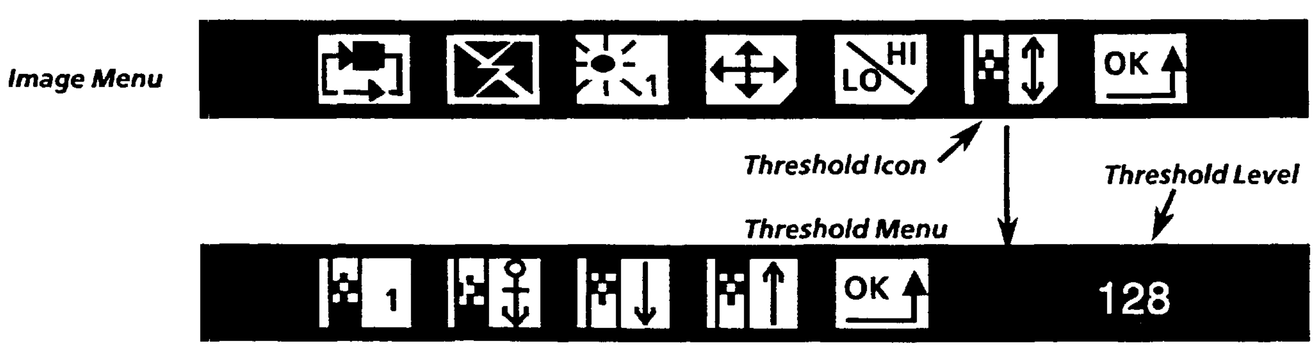
- *After* you have set up the workstage, and have a workpiece in position (see Chapter 4), so that the workpiece features to be inspected are shown clearly in the field of view (and on the monitor).
- *After* you have positioned the brightness probe.
- *After* you have selected the appropriate strobe setting.
- *Before* attempting to configure a line gage and/or window assigned to the threshold image.

Accessing the Threshold Menu

To set up a threshold image, first access the *Threshold* menu.

Note: The "Quick Start Procedure" in Chapter 6 includes a specific example of setting up threshold images 1 and 2. The two example threshold images are used in example procedures throughout this manual.

Your Action	***Comments***
Pick the Threshold *icon on the* Setup *menu to access the* Threshold *menu.*	The *Threshold* icon is the icon second from the right on the *Setup* menu. Picking the *Threshold* icon displays the *Threshold* menu.



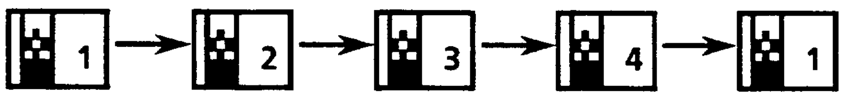
Selecting the Threshold Image

Select the threshold image you want to set up by using the *Image Select* icon.

Threshold Menu

Image Select Icon

Your Action	***Comments***
Pick the Image Select *icon to change the* image *number.*	Picking the *Image Select* icon (the first icon on the left) changes the *image* number. As you change the image number by picking the *Image Select* icon, the corresponding threshold image is displayed on the monitor. Picking the *Image Select* icon repeatedly cycles through the four threshold images. Pick the icon until you have selected the threshold image you wish to set up.



Note: When selecting the threshold image, remember – if you intend to use one or more windows in your inspection – each window, 1-4, is automatically assigned to inspect the threshold image of the same number. That is, window 1 always inspects threshold image 1, window 2 inspects threshold image 2, etc. (see Chapter 10 for more information on windows). For line gages, you can select any one of the four threshold images for each line gage you configure.

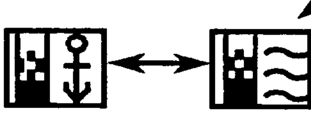
Select Anchor or Float Status

After selecting the threshold image number, set the *Anchor/ Float* status of the threshold image using the *Anchor / Float* icon. *Anchor* or *Float* status refers to the VIM2 module's *brightness compensation* capability (see "Brightness Compensation" earlier in this chapter).

Threshold Menu

Anchor / Float Icon (toggles when picked)

Anchor



Float

Select Anchor or Float Status *(continued)*

When setting up a threshold image, you select *Float* if you expect the lighting to vary, and you want to *enable* the threshold image to use the brightness compensation capability. With *Anchor* selected, brightness compensation is *disabled* for that threshold image.

Your Action	Comments
Pick the Anchor / Float *icon*	Picking the *Anchor / Float* icon toggles the threshold image status between *Anchor* (indicated by the *Anchor* icon) and *Float* (indicated by three wavy lines in the icon). As you toggle the icon, the appearance of the threshold image on the monitor does not change.

Anchor

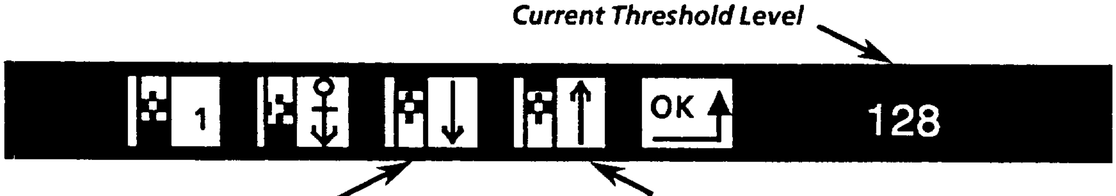
Float

Setting the Threshold Level

Notice the number on the right side of the *Threshold* menu; this number is the *current threshold level* of the selected threshold image. You can set this number to a value from 0 to 255. The default setting is 128.

Current Threshold Level

Threshold Menu



Decrease Icon

Increase Icon

To change the threshold level, pick either the *Increase* icon (to increase the threshold level), or the *Decrease* icon (to decrease the threshold level):

Your Action	Comments
Pick the Decrease *icon to decrease the threshold level.*	Each time you pick the *Decrease* icon, the threshold level decreases one unit. Pick and hold to decrease rapidly.
Pick the Increase *icon to increase the threshold level.*	Each time you pick the *Increase* icon, the threshold level increases one unit. Pick and hold to increase rapidly.

As previously discussed in this chapter and in Chapter 3, changing the threshold level can dramatically alter the threshold image. Set the threshold level in order to best highlight the feature(s) of interest you want to measure.

Halting Image Acquisition

When the VIM2 module is in the *setup* mode (the mode used for configuring the VIM2 module), images normally are repeatedly and automatically acquired by the VIM2 module. If you choose, you can halt the image acquisition at any time, so that an acquired image remains "frozen" on the monitor. You halt automatic image acquisitions using the *Image Acquisition* icon on the *Image* menu.

Why use the "halt image" capability?
During configuration, it may be necessary or desirable to halt the image acquisitions. For example, you may want to configure vision tools based on the workpiece image, but the workpiece is not available for the entire setup process.

How can you use the "halt image" capability?
If you can have a workpiece in position in the workstage for only a limited time, you can, when the workpiece is available, run the VIM2 module in the normal *Image Acquisition* mode. Then, before the workpiece is removed from the field of view, you can select the *Halt Image* mode.

Another opportunity for using the halt image capability occurs when you use one of the *Halt Image* arm modes during a trial run of the VIM2 module (Chapter 13 provides information on the arm modes). When you set the VIM2 module for either the *Halt on Reject* or the *Halt on Next Inspection* option, and a halt occurs, the VIM2 module stops inspections, and stores all four of the threshold images from the inspection. You can at that point re-enter setup mode; the halt image feature will automatically be enabled.

What happens with "halt image" selected?
With the *Halt Image* mode selected, the VIM2 module stops acquiring images; all four of the threshold images remain intact for the last image acquired. This allows you to configure or adjust line gages and windows based on these threshold images. However, you cannot move the brightness probe, or change the threshold levels of the four threshold images while *Halt Image* is selected.

Halting Image Acquisition *(continued)*

To select the *Halt Image* mode, begin by picking the *Image Acquisition* icon on the *Image* menu.

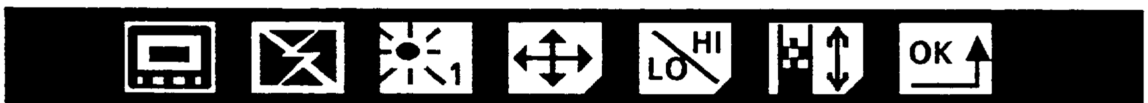
Your Action	Comment
Pick the Image Acquisition *icon on the* Image *menu.*	When the *Image Acquisition* icon is picked, the icon menu strip disappears from the screen. The VIM2 module begins to repeatedly acquire images, and updates the monitor image. Each image displayed is brightness compensated.

Image Menu

Image Acquisition Icon

Your Action	Comment
Depress the light pen tip to halt *image acquisition.*	When you depress the light pen tip at this point, the VIM2 module stops acquiring new images. The monitor image appears "frozen." The icon menu strip reappears, with the *Halt Image* icon in place of the *Image Acquisition* icon.

Image Menu



Halt Image Icon

Note: If you select the *halt on reject* mode (see Chapter 13 for *run mode* information), and you return to the *setup* mode after a reject decision occurs during run mode, the *Halt Image* mode will be selected automatically, so that the last acquired image is saved. This way you can review your line gage and/or window configurations and results as desired, based on the last acquired image. If and when you pick the *Halt Image* icon to resume the normal *Image Acquisition* mode, the last acquired image is *lost*, as the VIM2 module begins acquiring new images. Therefore make sure your work with the halted image is complete before resuming normal image acquisition.

To resume image acquisition:

Your Action	Comment
Pick the Halt Image *icon icon on the* Image *menu.*	When the *Halt Image* icon is picked, the icon menu strip disappears. The VIM2 module begins to repeatedly acquire images; images displayed are still brightness compensated.
Depress the light pen tip to begin *image acquisition.*	When you depress the light pen tip at this point, the VIM2 module stops acquiring new images. The monitor image is "live" as usual during setup (unless the *Strobe* option is enabled) without brightness compensation. The icon menu strip reappears, with the *Image Acquisition* icon in place of the *Halt Image* icon.

Brightness Registration

The VIM2 module has a capability called *brightness registration* which allows you to automatically adjust the threshold levels of threshold images according to a new nominal brightness level.

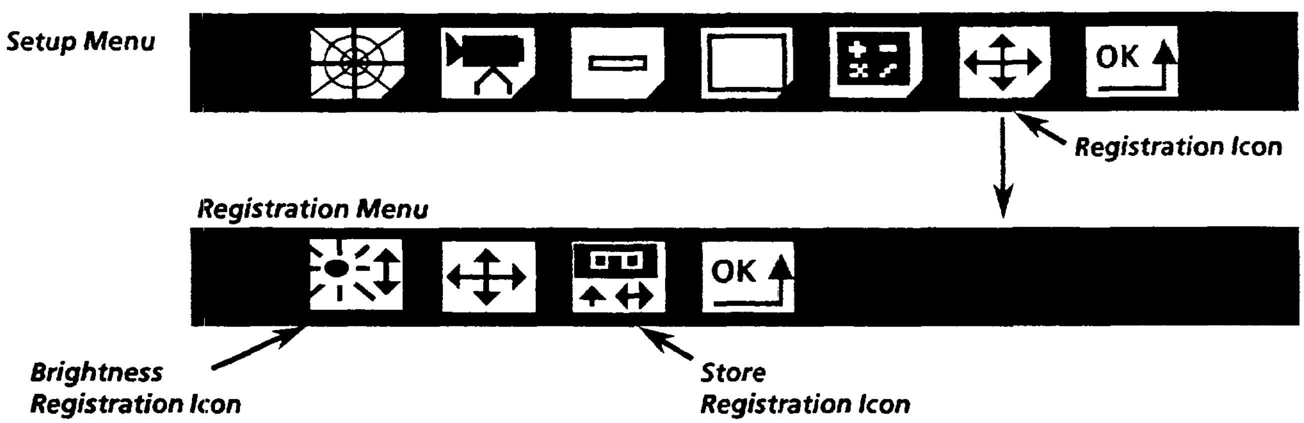
Note: In order to make use of this capability, threshold images must be set to *Float* status (i.e., brightness compensation *enabled*). Any threshold image which is set for *Anchor* status (brightness compensation *disabled*) is not affected by the use of brightness registration.

Use brightness registration when the nominal or usual brightness level for your inspection changes, and you want to readjust the threshold images accordingly.

Note: Brightness registration can be used with either a "live" image or a halted image. If the *Halt Image* icon is selected, registration is based on probe levels and thresholds computed during the prior image acquisition.

To use brightness registration, return to the *Setup* menu, and pick the *Registration* icon to access the *Registration* menu.

Your Action	Comment
Pick the Registration *icon on the* Setup *menu to access the* Registration *menu.*	The *Registration* icon is the next to last icon on the *Setup* menu.



After you have accessed the *Registration* menu, effecting brightness registration requires just two steps – picking the *Brightness Registration* icon, then picking the *Store Registration* icon.

Your Action	Comment
Pick the Brightness Registration *icon on the* Registration *menu.*	Picking the brightness registration icon (the icon color is *inverted* when picked) causes brightness registration to take effect immediately.

Brightness Registration
(continued)

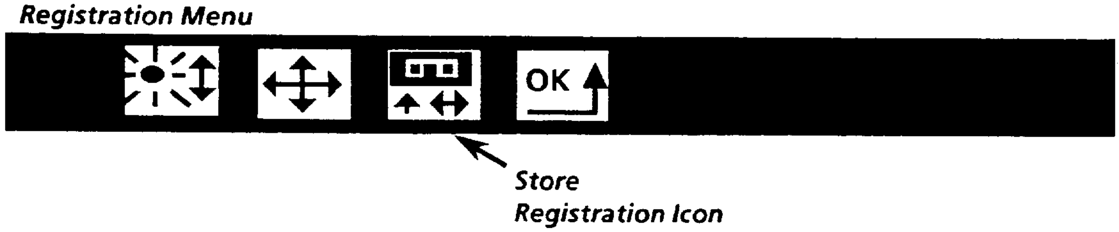
If the brightness probe detects a brightness level which is different from the learned nominal level, the newly detected level becomes the nominal level. Each *threshold image* which is configured for *float* (brightness compensation *enabled*) is then adjusted in the same amount and direction as the change in the brightness level; you can verify this by checking the threshold levels for the threshold images.

Note: If you do not want to use the new nominal brightness level and threshold levels, pick the *Brightness Registration* icon again, or simply pick the *OK* icon to exit the *Registration* menu. The brightness registration will be cancelled.

Otherwise, if you *do* want to use the new nominal brightness level and threshold levels, *you must store them as follows*:

Your Action	Comment
Pick the Store registration *icon to save the new parameters.*	Picking the *Store Registration* icon stores the new nominal brightness level and thresholds as part of the current working (RAM) configuration.

Registration Menu



Note: If brightness registration is active, shift registration will use a compensated image when initiated (see *Shift Registration* in Chapter 11).

Note: Brightness and shift registration will be stored at the same time when you pick the *Store Registration* icon.

(This page intentionally left blank)

Chapter 9 Using Line Gages

Chapter Objectives

The objective of this chapter is to show how to use the *Line Gage* menu icons to set up a line gage as part of a VIM2 module configuration for an inspection application. This chapter begins with questions and answers about gages.

Some Questions and Answers about Gages

This section introduces the topic of *line gages* by posing some questions about line gages, and answering them.

What is a line gage?
A line gage is an inspection tool which checks features in a selected binarized (threshold) image by analyzing linear groups of pixels – either black or white (see Chapter 3).

What does a line gage detect?
Line gages can be set to operate in any of a number of ways. A line gage can detect position of edges of "blobs" (see Chapter 3 for "blob" description), detect position of center of blobs, detect widths of blobs, count number of pixels crossed (black or white), or count number of edges crossed.

How many line gages are there?
There are 24 line gages in all – line gages X, Y, and numbers 1-22. This chapter discusses *only line gages 1-22*. Line gages X and Y are special cases – they are used for workpiece position shift compensation (discussed in Chapter 11).

How do you set up a line gage?
Using the *Line Gage* menu *icons*, you first select the line gage number (1-22), and enable the gage. You then configure the selected line gage, which is shown on the monitor: you select which one of four *threshold images* to inspect; you select attributes such as *vertical* or *horizontal* orientation, line gage *function*, pixel *color*, etc.; you *place* the line gage over the feature(s) of interest on the threshold image; and you then set an *acceptance range* for the line gage inspection results – during inspection this range provides the basis for an accept or reject decision.

How can you place a line gage?
Using the *Line Gage* menu icons you can:

- Select horizontal or vertical orientation (line gages can have only one of two orientations, vertical or horizontal).
- Lengthen or shorten the line gage (a line gage can cross the entire monitor screen if need be, or be as short as a single pixel).
- Move the line gage up or down, and/or side to side, until positioned as required over the feature(s) of interest.

Some Questions and Answers about Gages *(continued)*

What does a line gage inspect?
A line gage inspects *only the pixels it crosses in the selected threshold image*, so positioning is very important. A line gage analyzes the pixels it crosses to detect *blobs*, *edges*, or *pixels* (see Chapter 3 for an overview of line gage operation).

When do you set up a line gage?
Set up a line gage after you have:

- Set up the workstage, and have a workpiece in the nominal or ideal position (see Chapter 4), so that the workpiece features to be inspected are shown clearly in the field of view (and on the monitor).
- Set the threshold image(s) to enhance the particular feature(s) you are going to inspect (see Chapter 8).

Accessing the Line Gage Menu

To begin to set up a line gage, first access the *Line Gage* menu as described below.

Your Action	Comments
Pick the Setup *icon on the* Main *menu to access the* Setup *menu.*	If the *Main* menu is displayed, access the *Setup* menu by picking the *Setup* icon (the "open padlock"). If the *Setup* menu is already displayed (as shown below), skip this step.

Setup Menu — Line Gage Icon

Your Action	Comments
Pick the Line Gage *icon.*	Picking the *Line Gage* icon on the *Setup* menu displays the *Line Gage* menu:

Line Gage Menu

Enable / Disable Icon

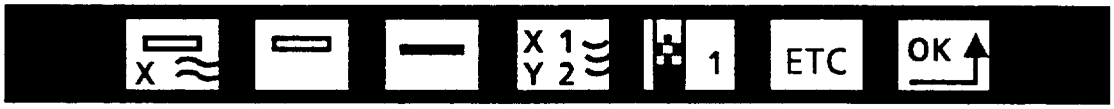
When you pick this icon, the diagonal slash disappears.

Your Action	Comments
Pick the second (enable/disable)*icon.*	Picking the *enable /disable* icon enables the line gage, and allows you to pick other icons on the *Line Gage* menu.
Pick the ETC *icon.*	Picking the *ETC* icon leads to more *Line Gage* icons:

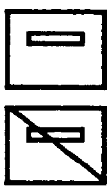
Line GageMenu – second part

The Line Gage Icons

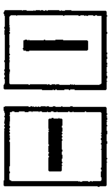
Here are brief descriptions of each of the *Line Gage* icons, beginning with the first set:



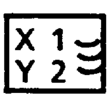
Gage Select – Pick this icon to select the gage from a choice of X, Y, or numbers 1-22. X, Y, then numbers 1-22, then X, Y, etc., are displayed in sequence as you repeatedly pick the icon, or hold the light pen tip down against the icon. The selected line gage is displayed on the monitor.



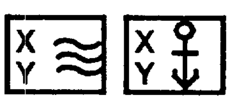
Gage Enable – Pick this icon to enable or disable the selected gage. This icon, when picked, toggles between *enabled* or *disabled* (*disabled* is indicated by diagonal slash across the icon). You must enable a selected gage in order to use the remaining *Line Gage* icons to configure a gage. Also, a gage must be enabled in order to be part of your inspection configuration.



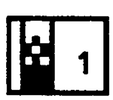
Gage Orientation – Pick this icon to select line gage orientation – either *horizontal* or *vertical*. For gages 1-22, this icon, when picked, toggles between vertical and horizontal.



Anchor / Float – This icon appears as shown at left when either the X or Y gage is selected.



For gages 1-22, this icon enables or disables position shift compensation, and when picked toggles between enabled (indicated by wavy lines) or disabled (indicated by anchor).

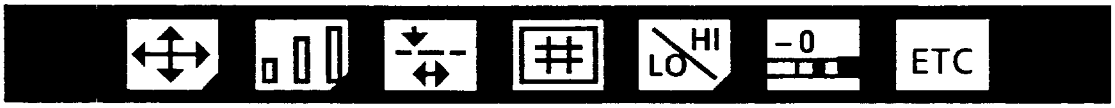


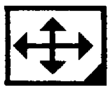
Threshold Image – Pick this icon to select the threshold image which the gage will inspect (there are 4 threshold image choices). As you repeatedly pick this icon, numbers 1-4 are displayed in sequence, indicating the selected threshold image. The selected threshold image is displayed on the monitor.

The other two icons displayed on the menu are the familiar *ETC* icon and *OK* icon. Recall that picking the *OK* icon returns you to the previous menu (the *Setup* menu in this case). Picking the *ETC* icon displays additional *Line Gage* menu icons (see next page).

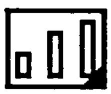
The Line Gage Icons
(continued)

If you pick the *ETC* icon on the *Line Gage* menu, a second set of *Line Gage* icons appears. These additional *Line Gage* icons are described below:

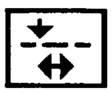




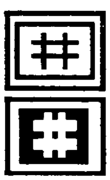
Gage Move – Pick this icon to access the *Gage Move* menu for the selected line gage. Use the *Gage Move* menu to strategically position the line gage on the image shown on the monitor screen.



Gage Size – Pick this icon to access the *Gage Size* menu for the selected line gage. Use the *Gage Size* menu to lengthen or shorten the line gage.



Gage Function – Pick this icon to assign the functions for the selected line gage. This icon, when picked, displays in sequence the different icons symbolizing the available line gage functions. Each icon represents two functions; a line gage always performs two inspection functions simultaneously.



Color – Pick this icon to select the color – either black or white – of the blobs or pixels to be detected by the line gage inspection. This icon, when picked, toggles between black (indicated by the black box with white "#") and white (white box with black "#").

Acceptance Range – Pick this icon to access the *Acceptance Range* menu for the selected line gage. Use the *Acceptance Range* menu to set the upper and lower limits for the inspection results, and to read the current line gage result and/or line gage statistics.

Gage Filter – Pick this icon to select the type of pixel filtering for the selected line gage. Picking this icon toggles through the three options – none ("0"), 1-, or 2-pixel filtering.

If you pick the *ETC* icon on the second *Line Gage* menu, the first set of *Line Gage* icons returns to the screen.

Exiting the Line Gage Menu

Remember, to exit the *Line Gage* menu, you must return to the first set of line gage icons, which includes the *OK* icon. Then, pick the *OK* icon to exit the *Line Gage* menu and return to the *Setup* menu.

Configuring Line Gages

This section explores the features and options you will encounter when configuring a line gage, and leads you through example procedures for setting up a line gage.

Note: For example purposes, these procedures refer to the target pattern included with this manual. Refer to the "Quick Start" section of Chapter 6 for more information.

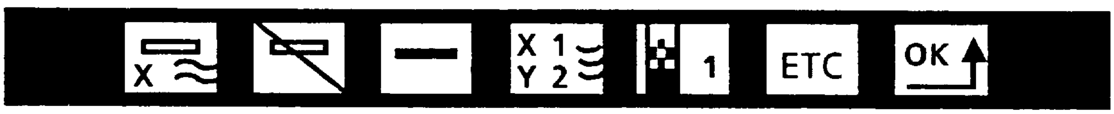
Access the Line Gage Menu

Your first step is to access the *Line Gage* menu.

Your Action	Comments
Pick the Setup *icon on the* Main *menu to access the* Setup *menu*	If the *Main* menu is displayed, pick the *Setup* (the "open padlock") icon to access the *Setup* menu (if the *Setup* menu is already displayed, skip this step):

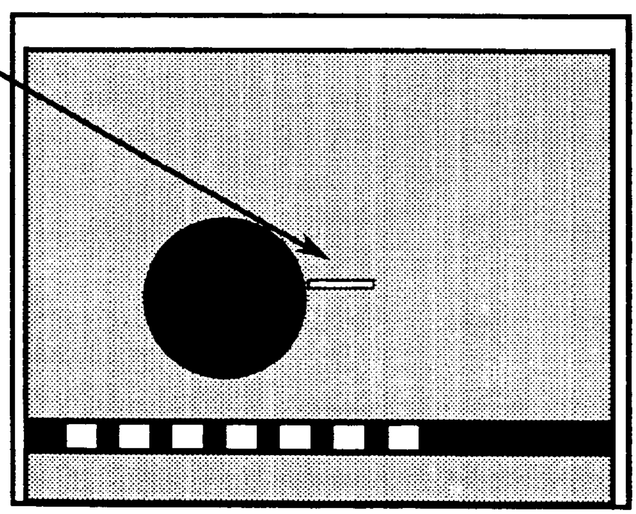
Setup Menu **Line Gage Icon**

Your Action	Comments
Pick the Line Gage *icon to access the* Line Gage *menu.*	When you pick the *Line Gage* icon, the *Line Gage* menu appears:



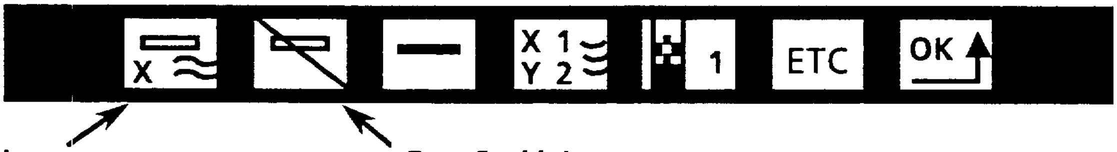
When you select the *Line Gage* menu, a line gage is displayed on the monitor. The first time you use the *Line Gage* menu, the line gage is displayed in the default position in the center of the screen, with horizontal orientation (see illustration below). The default threshold image is "1."

Line gage displayed on threshold image 1 (example image).



Select and Enable Line Gage

Select the line gage you wish to configure using the *Gage Select* icon (**Note:** Two special cases of line gage, the "X" and "Y" float gages, are discussed in Chapter 11; this chapter refers only to line gages 1-22).



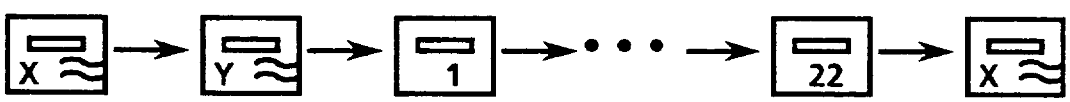
Gage Select Icon — *Gage Enable* Icon

When you pick the *Gage Select* icon, the number increments through all the line gages – X, Y, 1, 2 . . . 22, X, Y, 1, etc. The selected line gage is displayed on the monitor; when first displayed, the line gage is shown in the "home" position in the center of the monitor, with horizontal orientation.

Gage Enable – In order to configure the selected line gage, you must *enable* the line gage. Pick the *Gage Enable* icon so the diagonal slash is removed, indicating gage is enabled.

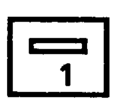
Note: A line gage must be *enabled* in order to be part of your inspection operation. You can disable a line gage after you have configured it – when *disabled,* a line gage does not function as part of the configuration (**Note:** A line gage cannot be *disabled* if it is included in a math tool configuration – see Chapter 12).

Your Action	***Comments***
Select the line gage number.	Pick the *Gage Select* icon (the first icon on the left) to change the gage number. To cycle through the line gages rapidly, hold the light pen tip down against the icon.

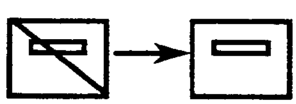


Example: Select and Enable Line Gage 1

Select line gage number 1.	Pick the *Gage Select* icon until the number "1" is displayed.

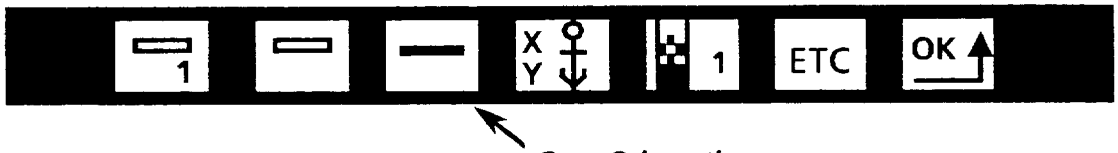


Enable selected line gage.	Pick the *Gage Enable* icon (the second icon) so the diagonal slash is removed, indicating the line gage is enabled.



Select Gage Orientation

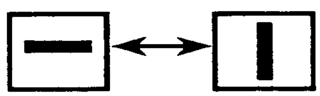
Each line gage is oriented in either of two selectable directions – *horizontal* or *vertical*; you select the gage orientation by picking the *Gage Orientation* icon.



Line gages are *horizontal* when first selected. As you pick the *Gage Orientation* icon, the icon toggles. The orientation of the line gage on the screen matches that of the icon.

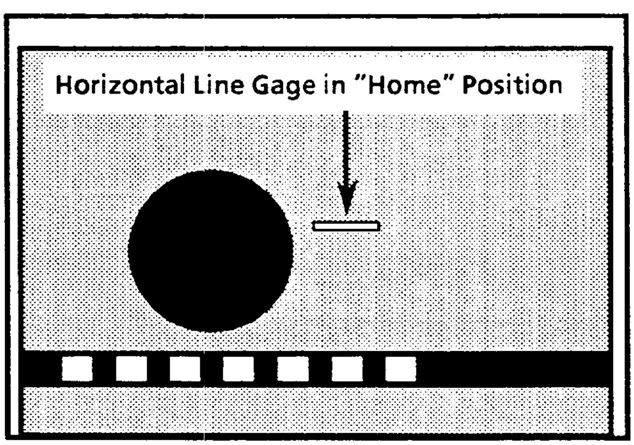
Select the orientation based on the application. If you wish to measure vertical dimensions of the workpiece feature(s), for example, choose *vertical* for the line gage orientation.

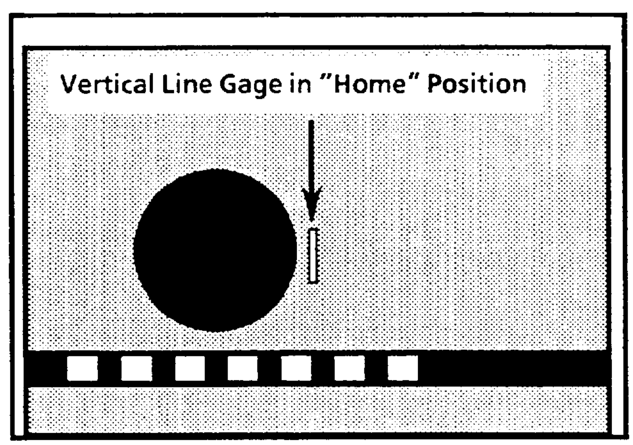
Your Action	*Comments*
Change the Gage Orientation.	Pick the *Gage Orientation* icon to toggle the line gage between *vertical* and *horizontal.*

Horizontal  Vertical

Example – Select Horizontal Orientation

Select horizontal. Pick the *Gage Orientation* icon so the *horizontal* line gage is displayed on the screen.

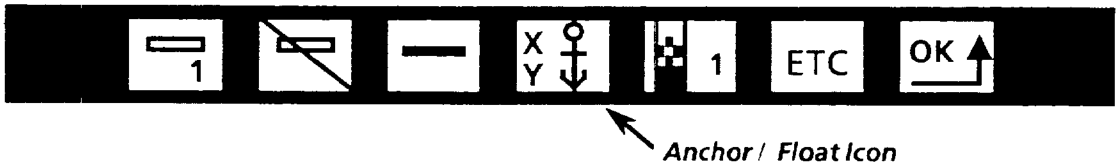




Note: The VIM2 module stores the horizontal and vertical selections as two different gages in terms of line gage placement. For example, if you select *vertical*, then change the position of the gage, then select *horizontal*, the gage orientation changes to *horizontal*, and the line gage returns to its "home" position in the center of the screen. If you then select *vertical* again, the line gage returns to the last position it held as a *vertical* gage.

Select Anchor or Float Status

Anchor or *Float* status refers to VIM2 module's X/Y shift compensation capability. Use the *Anchor / Float* icon to enable or disable the X/Yshift compensation feature.

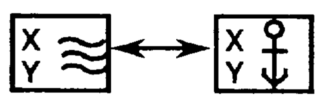


Anchor / Float Icon

X / Y Shift compensation – The VIM2 module can be configured to adjust for shifts of the workpiece in the image from inspection to inspection, using the "X" and "Y" float gages (the VIM2 module's X/Y shift compensation is discussed in detail in Chapter 11).

When configuring a line gage, you select *Float* if you expect the workpiece to shift, and you want to *enable* the line gage to use the shift compensation capability. With *Anchor* selected, position shift compensation is *disabled.*

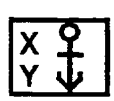
Your Action	***Comments***
Change the Anchor / Float status.	Pick the *Anchor / Float* icon to toggle the line gage status between *anchor* (indicated by the anchor in the icon) and *float* (indicated by the three wavy lines).* As you toggle the icon, the appearance of the line gage does not change.

Float  Anchor

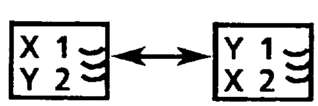
***Note:** In order to select the *float* status for a line gage, either or both of the "X" and "Y" float gages must first be enabled (if the *Anchor / Float* icon does not toggle, this is because both the "X" and "Y" float gages are disabled).

Example – Select Anchor

Select Anchor Status — Pick the *Anchor / Float* icon so that the *Anchor* is displayed.

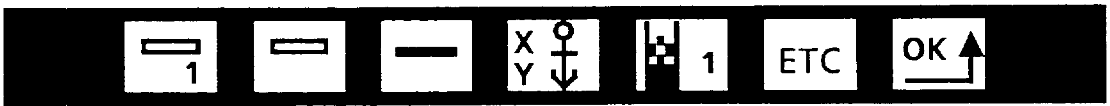


***Note:** The *Anchor / Float* icon appears as shown above for line gages 1-22 only. When either "X" or "Y" float gages are selected, the icon has a different function, and toggles between these two icons (see Chapter 11 regarding float gage information):



Select Threshold Image

Select the threshold image you wish to use with the selected line gage. There are four different threshold images (numbered 1-4) available. The number in the threshold image icon indicates the selected threshold image. The selected threshold image is displayed on the screen.

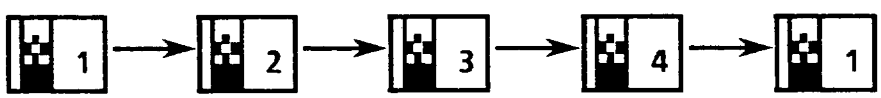


Threshold Image Icon

In selecting the threshold image, pick the one which best highlights the feature you wish to measure with the selected line gage. That is, the feature you want to measure should stand out in contrast from the area surrounding that feature.

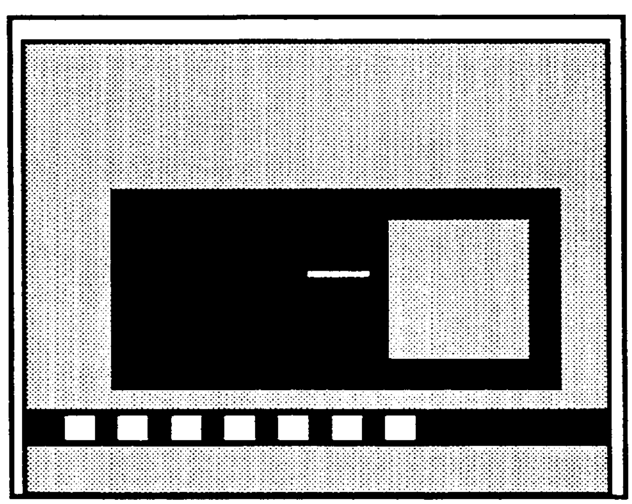
Note: Prior to setting up the line gages, first set the threshold images you wish to use for your inspections, at least preliminarily (see Chapter 7). For example purposes, we refer in this chapter to the use of the two threshold images set in the "Quick Start" section of Chapter 6.

Your Action	*Comments*
Pick the threshold image icon to change threshold images.	Picking the threshold image icon cycles through images 1- 4. The corresponding images are shown on the monitor.



Select threshold image 2.	**Example - Select Threshold Image 2** Pick the icon until the number "2" appears in the icon. With the icon showing "2," threshold image 2 appears on the screen (see diagram below).

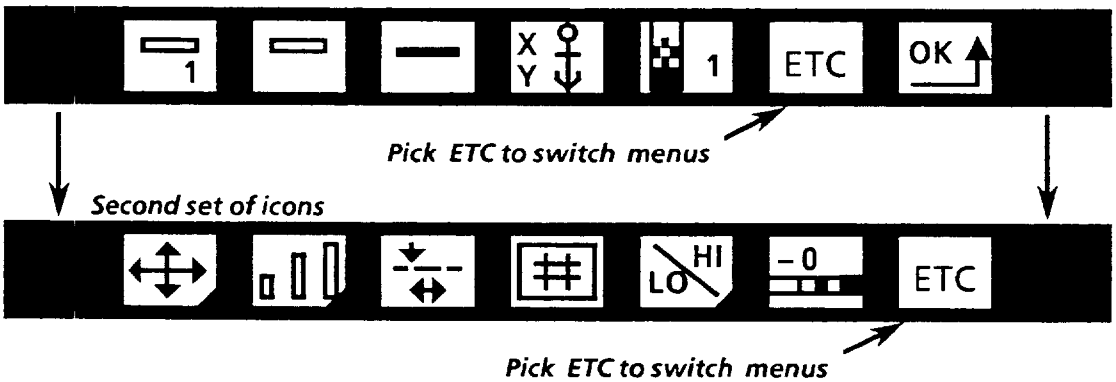
Line Gage Displayed on Threshold Image 2 (example image)



Additional Line Gage Icons

At this point, go on to the second set of line gage icons to continue. To do this, use the *ETC* icon.

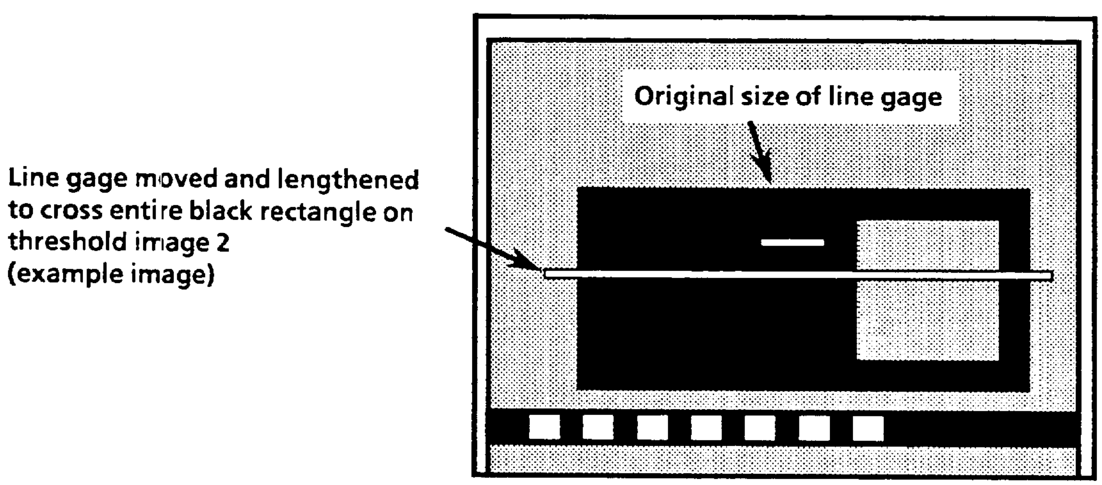
Your Action	***Comments***
Pick the ETC *icon to access the second set of* Line Gage *menu icons.*	When you pick the *ETC* icon, the second part of the line gage menu is displayed (**Note:** Pick *ETC* on the second menu to redisplay the first).



Placing the Line Gage

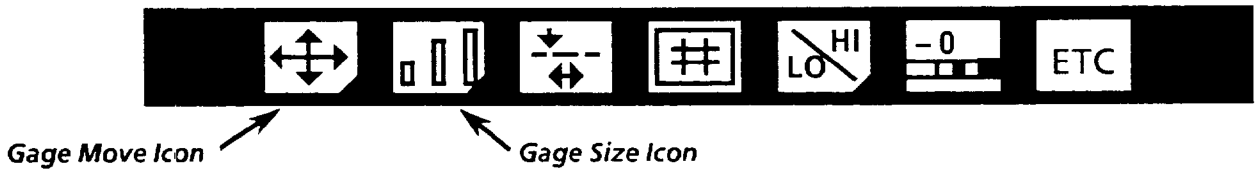
Placing the line gage includes both *moving the gage*, and also *adjusting the gage size (length)* as necessary. This section discusses using the *Gage Move* and *Gage Size* menus for the purpose of placing a line gage in position for an inspection.

Generally speaking, you should place a gage so that the gage crosses the feature to be measured with some overlap on each side of the feature. To do this, you probably will need to both move the line gage (using the *Gage Move* menu), and adjust the line gage length (using the *Gage Size* menu).



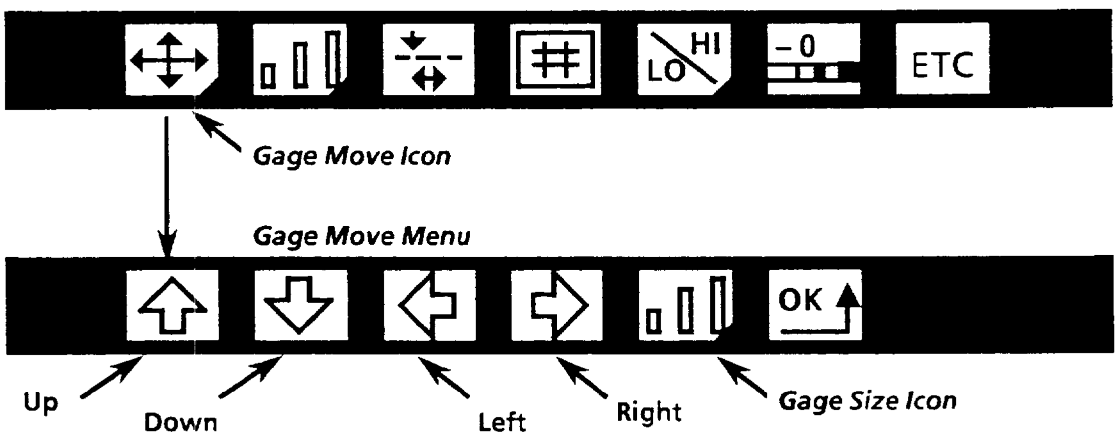
Placing the Line Gage
(continued)

The first two icons on the second *Line Gage* menu are the *Gage Move* icon (used to access the *Gage Move* menu), and the *Gage Size* icon (used to access the *Gage Size* menu).



First let's look at the *Gage Move* menu.

Your Action	*Comments*
Pick the Gage Move *icon to access the* Gage Move *menu.*	The *Gage Move* icon is the first icon on the second *Line Gage* menu (indicated by the four directional arrows).



The *Gage Move* menu includes four directional arrow icons. Notice the *Gage Move* menu also includes the *Gage Size* icon for immediate access to the *Gage Size* menu (the *Gage Size* Menu has a *Gage Move* icon as well). The VIM2 line gage menus allow you to conveniently toggle back and forth between the *Gage Size* and *Gage Move* menus while placing the line gage.

To *move* the line gage into the desired position, pick the appropriate arrow icon(s), in one of two ways:

- *Pick* the arrow icon to move the line gage one pixel in the direction indicated by the icon.
- *Pick and hold* the light pen down against the arrow icon to cause *continuous movement* in the indicated direction (*if you pick an arrow icon and it flashes*, this indicates that the line gage is at a boundary – such as the screen edge –and cannot be moved further in that direction).

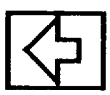
Placing the Line Gage (continued)

Note: Depending on the *brightness compensation* mode you have selected, placement of the line gage may be restricted (see Chapter 7 for brightness compensation information). Briefly, if you have selected *same frame* compensation (mode "1"), you cannot place a line gage in the top 48 pixel rows (or about the top 20%) of the video image.

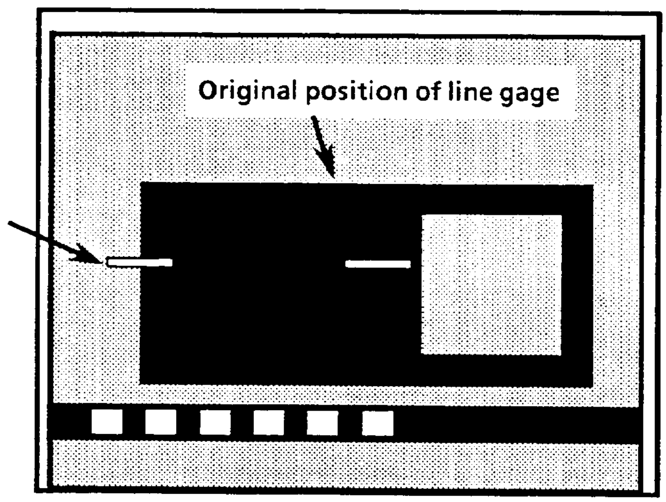
Example Line Gage Move
This demonstration places the line gage at the left edge of large black rectangle in threshold image 2.

Hint: When placing a line gage, *move* the gage first before adjusting the length. And, because the line gage length is adjusted from the right (or bottom) end, first move the gage so the left (top) end of the gage is in the desired position. Then adjust the length (using the *Gage Size* menu) so the right (bottom) end is in the desired position.

Your Action	*Comments*
Place the line gage across the left edge of the black rectangle *of the target.*	Pick the *left* arrow icon as necessary to move the line gage so the line gage crosses the left side of the black rectangle.



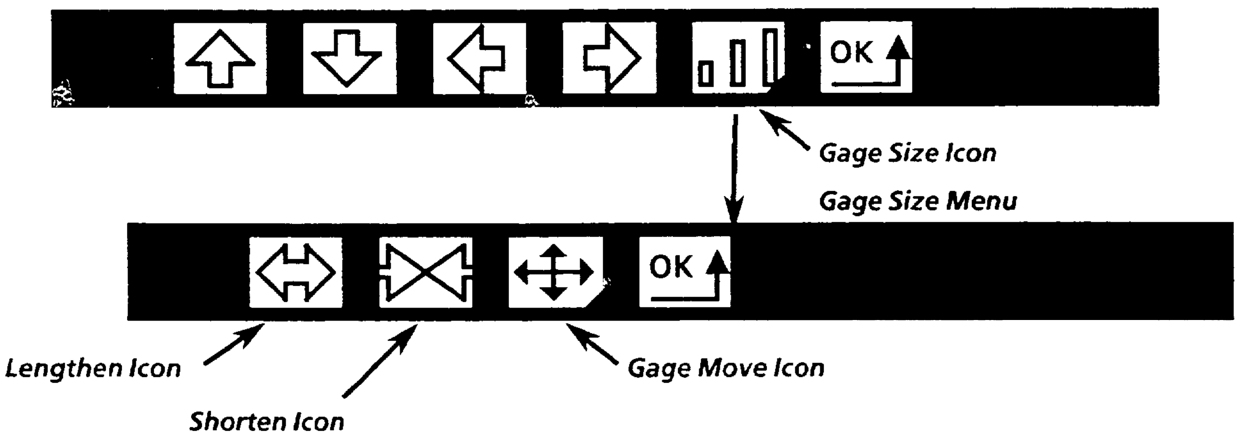
Line gage moved left to left edge of black rectangle on threshold image 2 (example image)



Placing the Line Gage *(continued)*

Adjusting Gage Size – Use the *Gage Size* menu to shorten or lengthen the line gage.

Your Action	Comments
Pick the Gage Size *icon on the* Gage Move *menu.*	If the *Gage Move* menu is displayed, access the *Gage Size* menu by picking the *Gage Size* icon:



Note: Remember you can also access the *Gage Size* from the *Line Gage* menu by picking the *Gage Size* icon on that menu. Also, notice the *Gage Move* icon, which can be used for immediate access to the *Gage Move* menu.

There are two ways of adjusting the line gage length using the *Gage Size* icons:

- *Pick* the appropriate *Gage Size* icon to lengthen or shorten the line gage one pixel at a time.
- *Pick and hold* the light pen down against the appropriate icon to cause *continuous lengthening* or *shortening.*
- If you pick *Gage Size* icon and it flashes, this indicates that the line gage is at its limit and cannot be further adjusted in the direction indicated by the selected icon.

Note: When you change the length of a line gage, the left (top) end of the line gage remains fixed; only the position of the right (bottom) end of the line gage changes.

Gage Size Limits

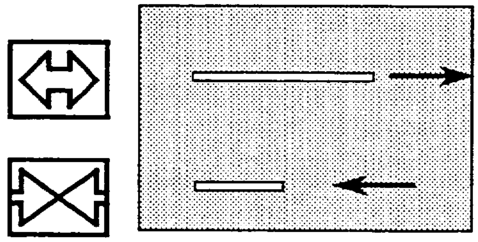
You can lengthen a line gage until it meets either the bottom edge (for a *vertical* gage) or right edge (for a *horizontal* gage) of the screen.

You can shorten a line gage to a *single pixel* in length.

Placing the Line Gage *(continued)*

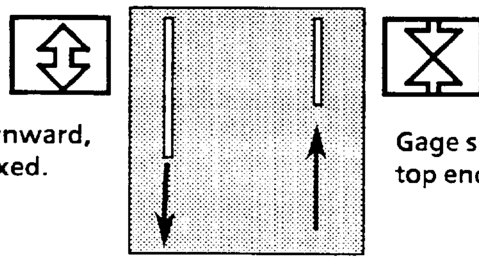
For a *horizontal* gage, pick the *lengthen* icon to lengthen to the right, or the *shorten* icon to shorten from the right:

Horizontal Gage



For a *vertical* gage, pick the *lengthen* icon to lengthen downward, or pick the *shorten* icon to shorten upward:

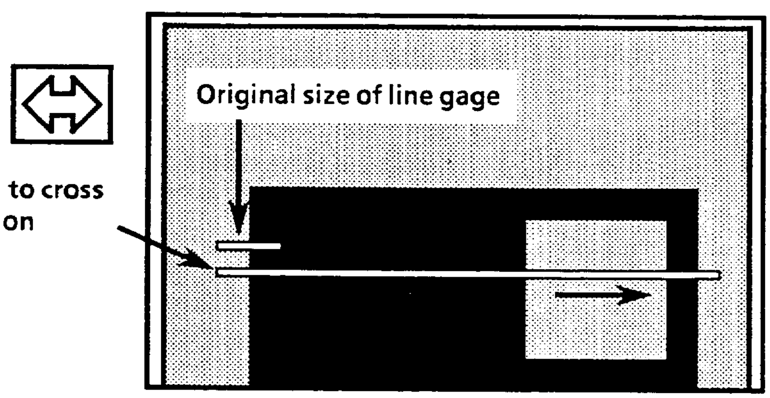
Vertical Gage



Example Size Adjustment

Lengthen the line gage so it crosses the entire width of black rectangle on the target pattern in threshold image 2.

Your Action	*Comments*
Lengthen the line gage so it crosses the right edge of the black rectangle.	Pick and hold the *Lengthen* icon as necessary to lengthen the line gage until it crosses the right edge of the black rectangle on the target pattern in threshold image 2:



Make sure the line gage crosses over the white square.	After placing the line gage, check to see if the line gage also crosses over the white square. If not, pick the *Gage Move* icon to access the *Gage Move* menu. Use the *Up* or *Down* icons to move the line gage so it crosses the white square.

Note: After placing the line gage, return to the *Line Gage* Menu by picking the *OK* icon.

Setting the Line Gage Function

Each of the line gages can perform several types of inspections. Setting the line gage function involves the use of these two icons:

Gage Function Icon – Pick this icon to assign the function for the selected line gage.

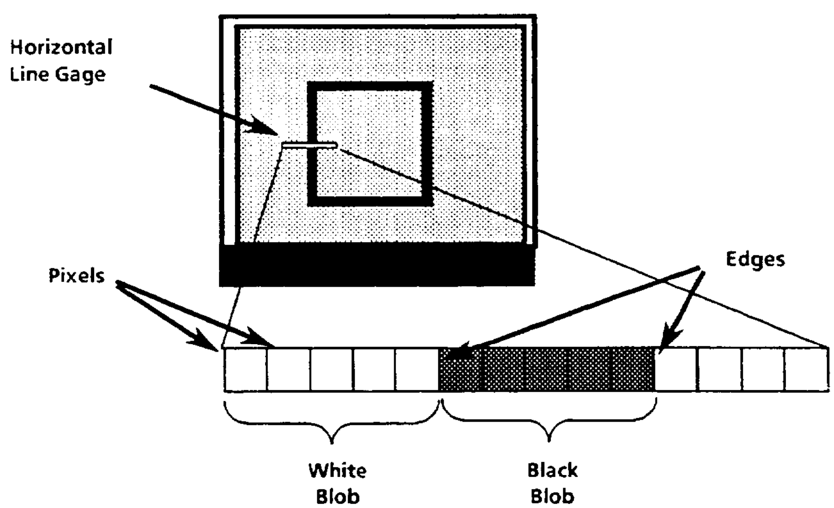
Color Icon – Pick this icon to select the blob or pixel color – either black or white – for the line gage function.

Gage Function

To set up the line gage function, (1) pick the *Gage Function* icon to select the function, then (2) pick the *Color* icon to set the *color (black* or *white)* for the selected function.

Recall from Chapter 3 that a line gage can be set up to detect three basic types of image characteristics (see figure below):

- **Blob** – a group of consecutive pixels of the same color
- **Edge** – wherever a black and a white blob meet (that is, wherever a black pixel is next to a white pixel) *
- **Pixel counts** – either black or white pixels. Recall that each threshold image consists of black and white pixels.

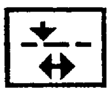
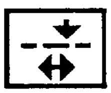
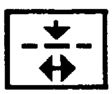
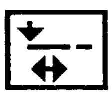
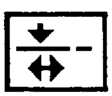
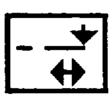
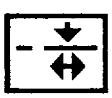
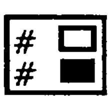
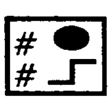


***Note on edge location:** The location given for the *left (top) edge* of a blob is the location of the first pixel of the blob – that is, where the line gage first detects a transition in color, indicating the beginning of a blob (the *vertical* line gage scans from top to bottom; the *horizontal* line gage scans from left to right). The location given for the *right (bottom) edge* of a blob is the location of the last pixel of that blob, before a change in pixel color is detected.

Setting the Line Gage Function *(continued)*

You select the function by picking the *Gage Function* icon. The *Gage Function* icon, when picked repeatedly, cycles through the icons symbolizing the available functions. Each icon represents two functions (referred to as *upper* and *lower* functions); each *enabled* line gage always performs the two selected functions (upper and lower) simultaneously.

Your Action	*Comments*
Pick the Gage Function *icon 9 times to cycle through all the available functions*	Picking the *Gage Function* icon changes the icon. The nine different icons and their associated functions are described throughout this section (**Note**: The last two icons below are not available for the X and Y float gages):

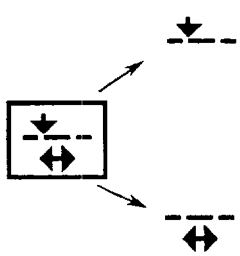
Icon	Functions (Upper and Lower)
	(Upper) Locate left (top)* edge of largest blob (Lower) Width of largest blob
	(Upper) Locate right (bottom)* edge of largest blob (Lower) Width of largest blob
	(Upper) Locate center of largest blob (Lower) Width of largest blob
	(Upper) Locate left (top) edge of leftmost (top)* blob (Lower) Width of leftmost (top)* blob
	(Upper) Locate center of leftmost (top)* blob (Lower) Width of leftmost (top)* blob
	(Upper) Locate right (bottom)* edge of rightmost (bottom)* blob (Lower) Width of rightmost (bottom)* blob
	(Upper) Locate center of rightmost (bottom) blob (Lower) Width of rightmost (bottom)* blob
	(Upper) Count white pixels (Lower) Count black pixels
	(Upper) Count blobs (Lower) Count edges

*Parentheses indicate function as used with a *vertical* line gage, as opposed to a *horizontal* line gage.

Setting the Line Gage Function *(continued)*

This section describes the different *Gage Function* icons and their associated functions, providing examples of most of the line gage functions. The remaining functions are similar to those illustrated by example, with only small differences in operation (these differences are explained).

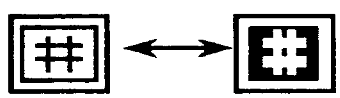
Your Action	*Comments*
Select the first *icon (shown below).*	Pick the *Gage Function* icon as necessary until the *first* icon is displayed (**Note:** This icon is the default setting).



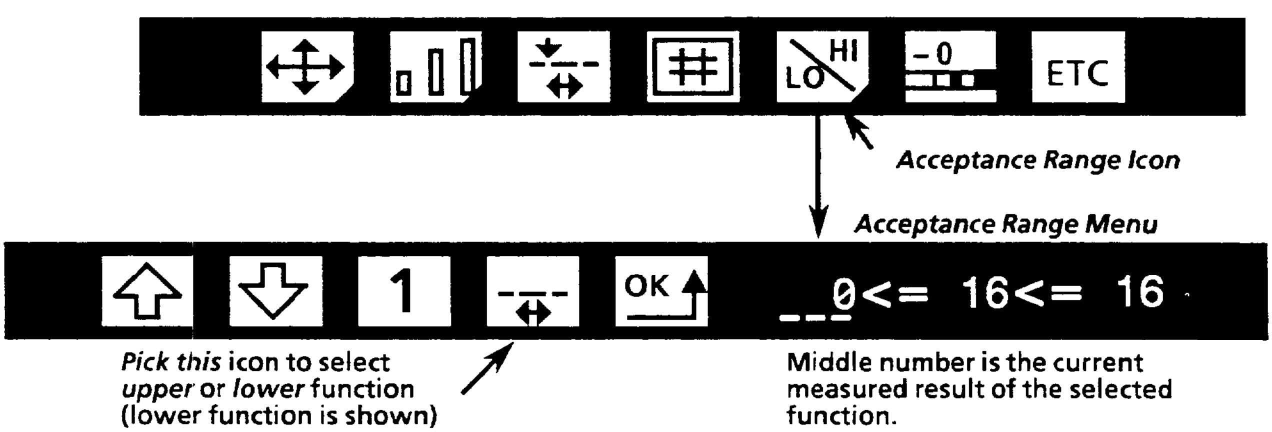
Upper Function – Gives pixel location of the *left edge (*or, for *vertical* gages, the *top edge)* of the *largest blob* (black or white, as selected by *Color* icon) crossed by the line gage. The location given for the edge is actually the pixel column number for *horizontal* line gages, or pixel row number for *vertical* (see example on next page).

Lower Function – Gives the *pixel count* of the width of the *largest blob* (black or white, as selected by color icon) crossed by the line gage (see example on next page).

Select the appropriate Color. — Pick the *Color* icon to select the color (black or white).



Note on displaying line gage measurement results: When setting up a line gage, access the *Acceptance Range* menu to display the measurement results of the line gage functions. To do this, after *placing* the gage, selecting the line gage *function,* and selecting *color*, pick the *Acceptance Range* ("Hi/Lo") icon:

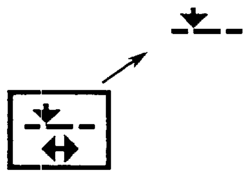


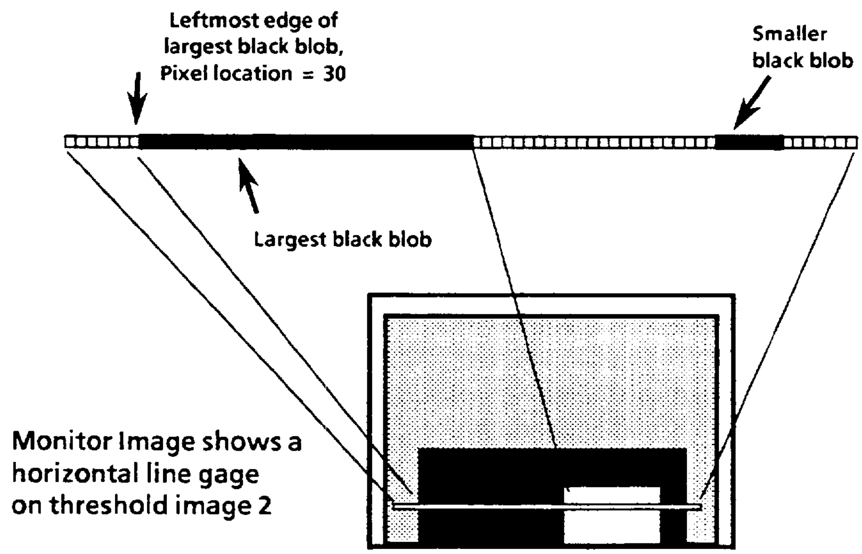
Note: For the upper function of the first seven gage function icons, if the line gage cannot locate a blob of the selected color (no pixels of the selected color are crossed by the line gage), a result value of 255 is listed, and the line gage fails.

Setting the Line Gage Function (continued)

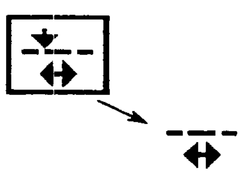
Here are examples of the upper and lower functions of the first *Gage Function* icon:

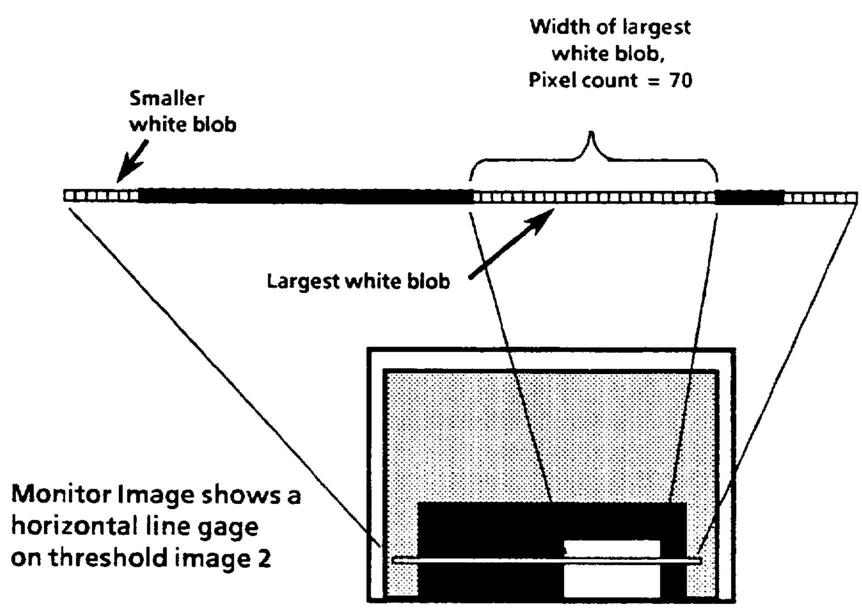
Example: Upper Function of first icon – Assuming the *Color* icon is set to *black*, the *horizontal* line gage shown below gives the location of the *left edge* of the *largest* black blob, which corresponds in this case to the left edge of the black rectangle (**Note**: the example pixel location = 30; pick the *Acceptance Range* icon to see the actual measured value).





Example: Lower Function of first icon – Assuming the *Color* icon is set to *white*, the *horizontal* the gage shown below gives the width (in pixel units) of the largest white blob, which corresponds in this case to the width of the white square (**Note**: the example pixel count = 70; pick the *Acceptance Range* icon to see the actual measured value).

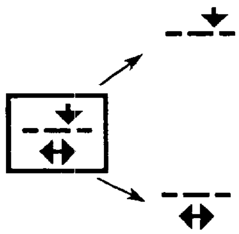




Setting the Line Gage Function
(continued)

Select the second *icon (shown below).*

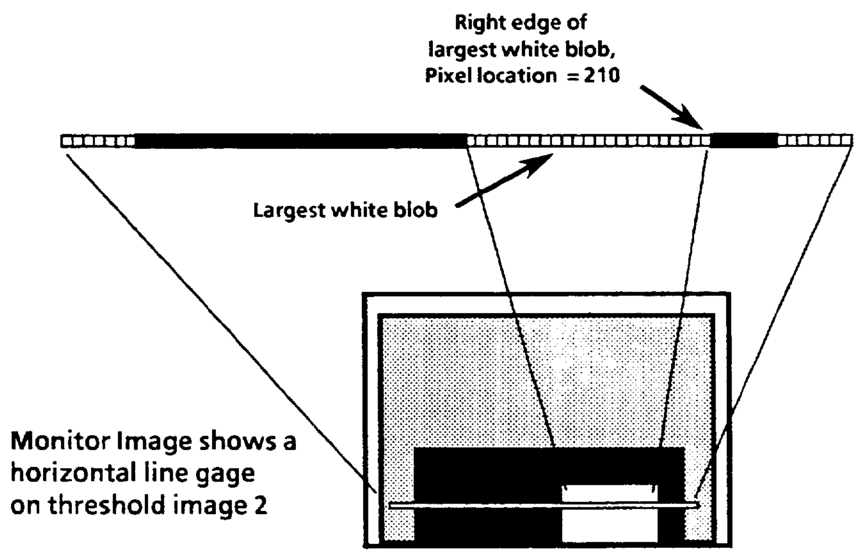
Pick the *Gage Function icon* to display the *second* icon.



Upper Function – Locates and gives pixel location of the *right (bottom) edge* of the *largest blob* (black or white, as selected by *Color* icon) crossed by the line gage.

Lower Function – (same as lower function of first icon).

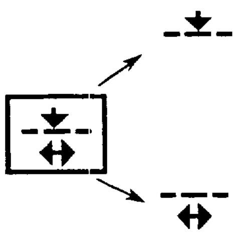
Example: Upper Function – Assuming the *Color* icon is set to *white*, the *horizontal* line gage shown below gives the pixel location of the right edge of the largest white blob, which corresponds in this case to the location of the right edge of the white square (**Note**: the example pixel location = 210; pick the *Acceptance Range* icon to see the actual measured value).



Setting the Line Gage Function (continued)

Select the third *icon (shown below).*

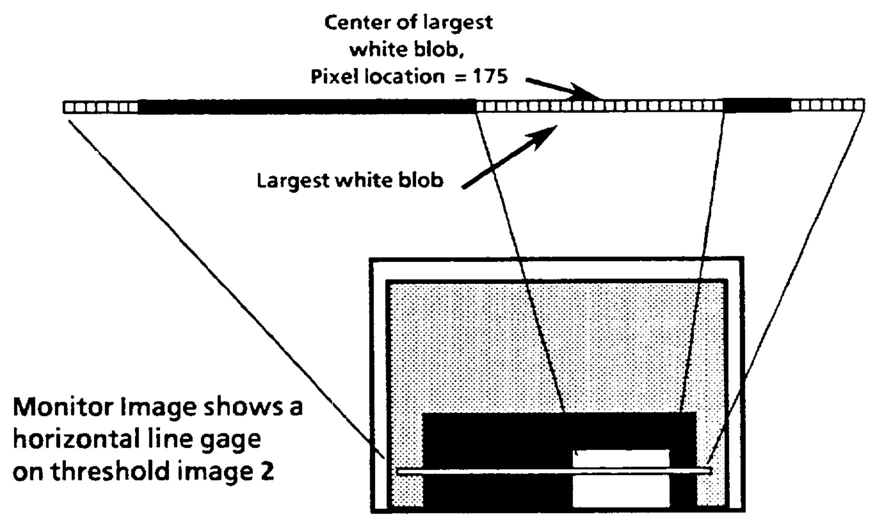
Pick the *Gage Function* icon to display the *third* icon.



Upper Function – Gives location of the *center pixel* of the *largest blob* (black or white, as selected by *Color* icon) crossed by the line gage.

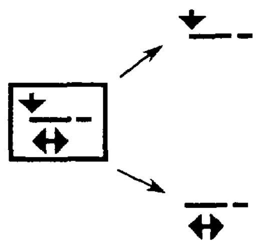
Lower Function – (same as lower function of first icon).

Example: Upper Function – Assuming the *Color* icon is set to *white*, the *horizontal* line gage shown below gives the location of the *center* pixel of the largest white blob, which corresponds to the horizontal center of the white square (**Note** : the example pixel location = 175; pick the *Acceptance Range* icon to see the actual measured value).



Setting the Line Gage Function (continued)

Select the fourth *icon (shown below).*

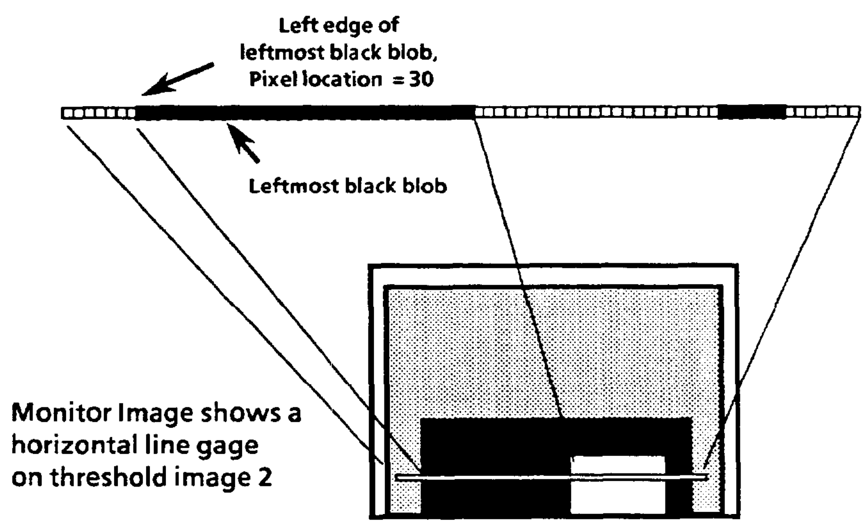


Pick the *Gage Function* icon to display the *fourth* icon.

Upper Function – Gives pixel location of the *left (top) edge* of the *leftmost (top) blob* (black or white, as selected by *Color* icon) crossed by the line gage.

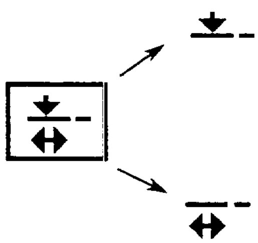
Lower Function – Gives the *pixel count* of the width of the *leftmost (top) blob* (black or white, as selected by color icon) crossed by the line gage. This is similar to the lower function of the *first icon* (see *first icon* example); the only difference is this function gives the width of the *leftmost (top)* blob instead of the *largest* blob.

Example: Upper Function – Assuming the *Color* icon is set to *black*, the *horizontal* line gage gives the location of the left edge of the leftmost black blob, which corresponds to the left edge of the black rectangle (Note : the example pixel location = 30; pick the *Acceptance Range* icon to see the actual measured value).



Setting the Line Gage Function (continued)

Select the fifth *icon (shown below).*

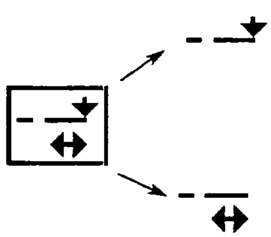


Pick the *Gage Function* icon to display the *fifth* icon.

Upper Function – Gives pixel location of the *center* of the *leftmost (top) blob* (black or white, as selected by *Color* icon). This is similar to the *third icon* upper function (see *third icon* example); the only difference is this function locates the center of the *leftmost (top)* blob instead of the center of the *largest* blob.

Lower Function – (same as lower function of fourth icon).

Select the sixth *icon (shown below).*

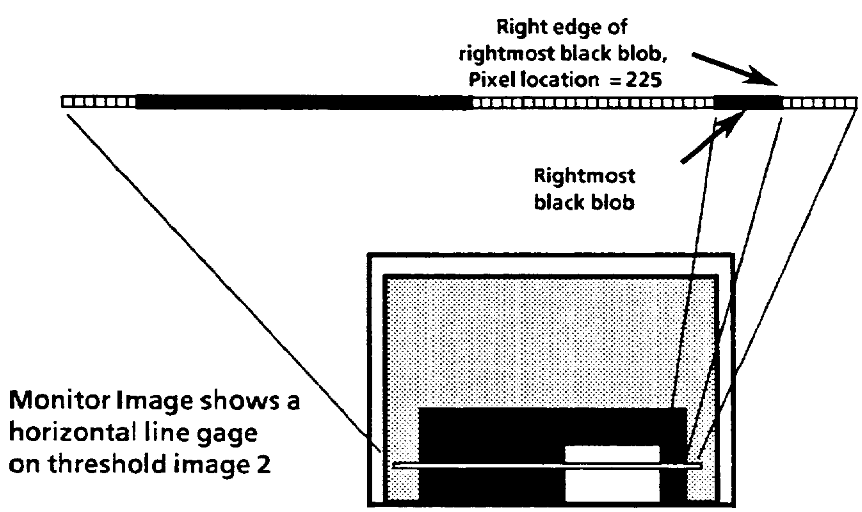


Pick the *Gage Function* icon to display the *sixth* icon.

Upper Function – Gives pixel location of the *right (bottom) edge* of the *rightmost (bottom) blob* (black or white, as selected by *Color* icon) crossed by the line gage.

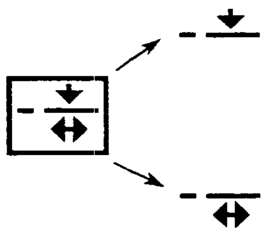
Lower Function – Gives the *pixel count* of the width of the *rightmost (bottom) blob* (black or white, as selected by color icon). This is similar to the *first icon* lower function (see *first icon* example); the only difference is this function gives the width of the *rightmost (bottom) blob* instead of the *largest* blob.

Example: Upper Function of sixth icon – Assuming the *Color* icon is set to *black*, a *horizontal* line gage gives the location of the right edge of the right-most black blob, corresponding to the right edge of the black rectangle (**Note** : the example pixel location = 225; pick the *Acceptance Range* icon to see the actual measured value).



Setting the Line Gage Function (continued)

Select the seventh *icon (shown below).*

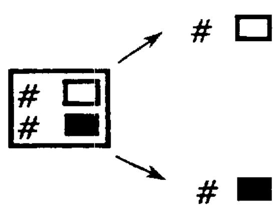


Pick the *Gage Function* icon to display the *seventh* icon.

Upper Function – Gives pixel location of the *center* of the *rightmost (bottom) blob* (black or white, as selected by *Color* icon). This is similar to the *third icon* upper function (see *third icon* example); the only difference is this function locates the *center* of the *rightmost (bottom) blob* instead of the *center* of the *largest* blob.

Lower Function – (same as lower function of *sixth* icon).

Select the eighth *icon (shown below).*

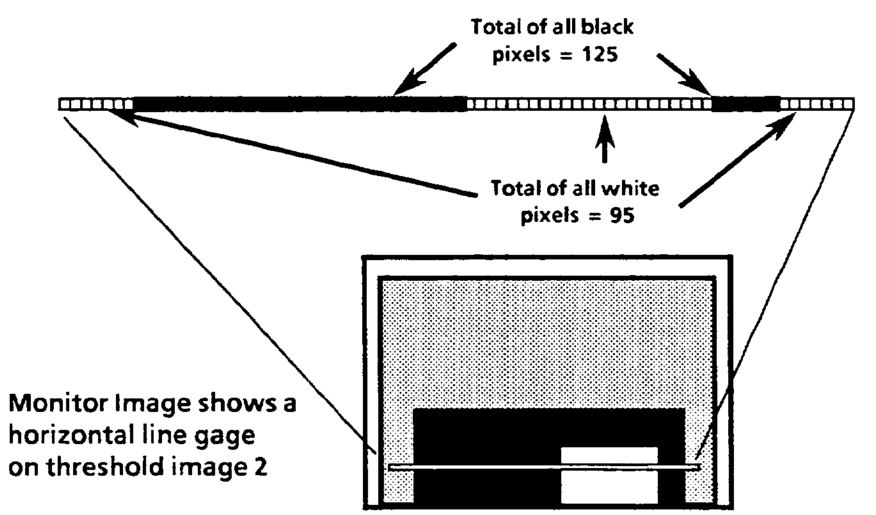


Pick the *Gage Function* icon to display the *eighth* icon.

Upper Function – Gives *total* of all *white pixels* crossed by the line gage (status of *Color* icon has no effect on this function).

Lower Function – Gives *total* of all *black pixels* crossed by the line gage (status of *Color* icon has no effect on this function).

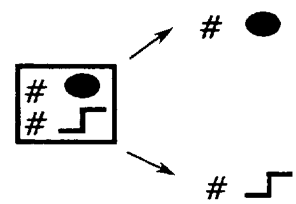
Example of Upper and Lower Functions – A *horizontal* line gage gives the number of *black pixels* crossed, and the number of *white pixels* crossed (**Note**: the example *black* pixel count = 125, and the example *white* pixel count = 95; pick the *Acceptance Range* icon to see the actual measured values).



Setting the Line Gage Function (continued)

Select the ninth *icon (shown below).*

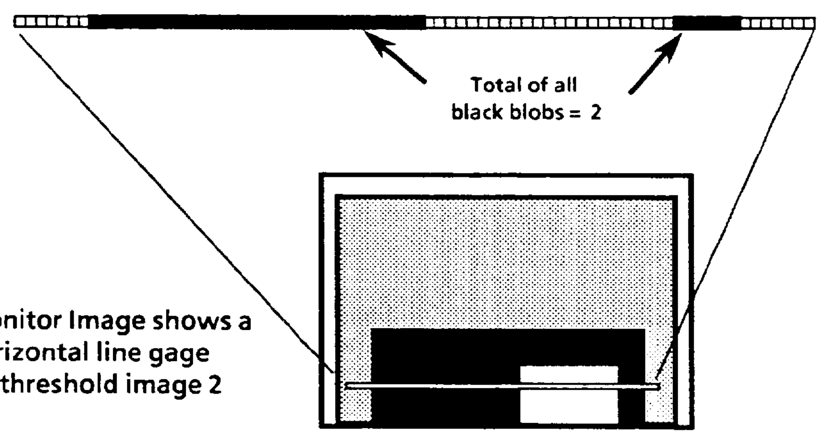
Pick the *Gage Function* icon to display the *ninth* icon.



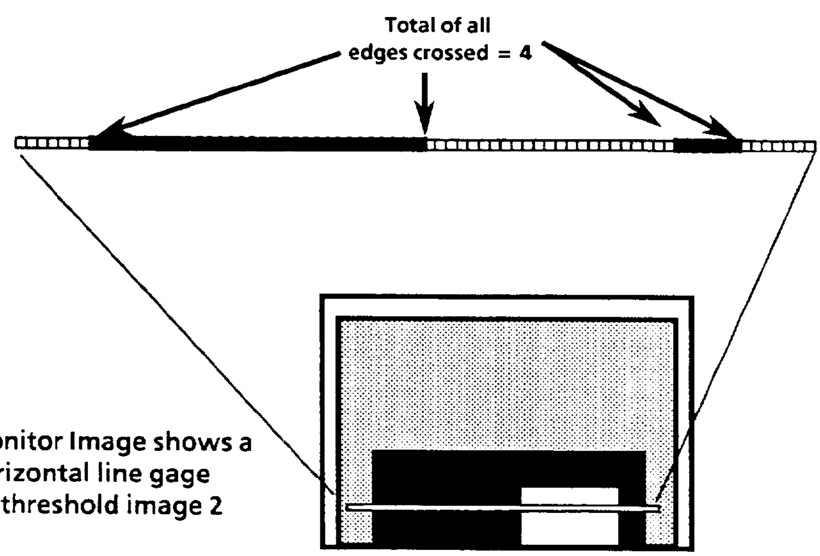
Upper Function – Gives *total* of all *blobs* (black or white, as selected by *Color* icon) crossed by the line gage.

Lower Function – Gives *total* of all *edges* crossed by the line gage (status of *Color* icon has no effect on this function).

Example: Upper Function – Assuming the *Color* icon is set to *black*, a *horizontal* line gage gives the number of *black blobs* crossed (2 in the diagram). **Note:** If the *Color* icon were set to *white*, the number of *white blobs* crossed would be 3 in the diagram below).



Example: Lower Function – A *horizontal* line gage gives the number of *edges* crossed (4 in the diagram below). **Note:** The line gage ends are not counted as edges by the line gage – only transitions from white to black or black to white are counted as edges).



Setting the Acceptance Range

Once you have placed a line gage and set the functions, the next step is to set the *acceptance range* of the line gage.

What is the acceptance range?
Before you use a line gage to measure the feature(s) of interest on a workpiece, you must set the range of acceptable measurement values for each function of that line gage. That is, you set an *upper* limit and a *lower* limit for the line gage results, for both the *upper* and *lower* function.

When the VIM2 performs an inspection, each line gage returns two results based on its two functions; if both line gage results come within the upper and lower limits (or *acceptance range*) for those functions, the line gage inspection passes. If *either* result is not within range, the line gage fails. If a line gage fails, the inspection fails; as a result, the VIM2 module indicates a *Reject* decision through the swingarm, the *Decision* LED, and the discrete bit, as well as through the results data blocks.

How do you set the acceptance range for a line gage?
In order to set the acceptance range for a line gage function, you must know (1) the measured result of the line gage function for a nominal or ideal part, and (2) the amount of variation that is acceptable for that measurement (if any). What you must do is:

1. Stage and light the ideal (or nominal) part with the part in the ideal (or nominal) position for inspection.

2. Set up the line gage (select and enable line gage, select threshold image, place the line gage, and select function).

3. Pick the *Acceptance Range* icon and observe the result for the ideal part for each of the two line gage functions.

4. Determine the amount of acceptable variation for the measured value for each of the two line gage functions, and set the upper and lower limits for each function.

   Hint: If you are interested in applying the result of only one of the two functions, set the upper and lower limits for the *unused* function to extreme values, so that the unused function *does not fail* during inspection.

   Do the same any time you need to use a gage function result and you do not want the result to cause the inspection to fail during operation. For example, you may want to use the line gage result in a math tool, and not have the line gage result cause an inspection failure.

Setting the Acceptance Range *(continued)*

5. Test the inspection configuration by using the VIM2 module as configured, and inspecting typical parts. Refine the line gage function limits as necessary.

Can you use the VIM2 module's Learn mode and statistics to help set the acceptance ranges?
Yes, a good way to set an appropriate acceptance range would be to use the VIM2 module's *learn* mode. You could, for example, set the acceptance ranges as follows:

1. When configuring the VIM2 module for your inspection, set the acceptance ranges for the line gage functions to *extreme limits,* since the VIM2 module does not include results of failed inspections in the statistics.

2. Set the VIM2 module to run in *learn* mode (see Chapter 13 for more information on the *learn* mode) before exiting to run mode.

3. While the VIM2 module is running in *learn* mode mode, inspect a sampling of known good parts.

4. Go back to setup mode, and observe and record the statistics for the line gage functions (see "Accessing Statistics" this chapter) for known good parts. **Note:** The statistics will include minimum result, maximum result, and average result for the inspected sample for each line gage function.

5. Reset the statistics to 0 (see *Resetting the Statistics*, Chapter 13).

6. Inspect a sampling of known bad parts, with the VIM2 module set to *learn* mode.

7. Go back to setup mode, and observe and record the statistics for the line gage functions for known bad parts.

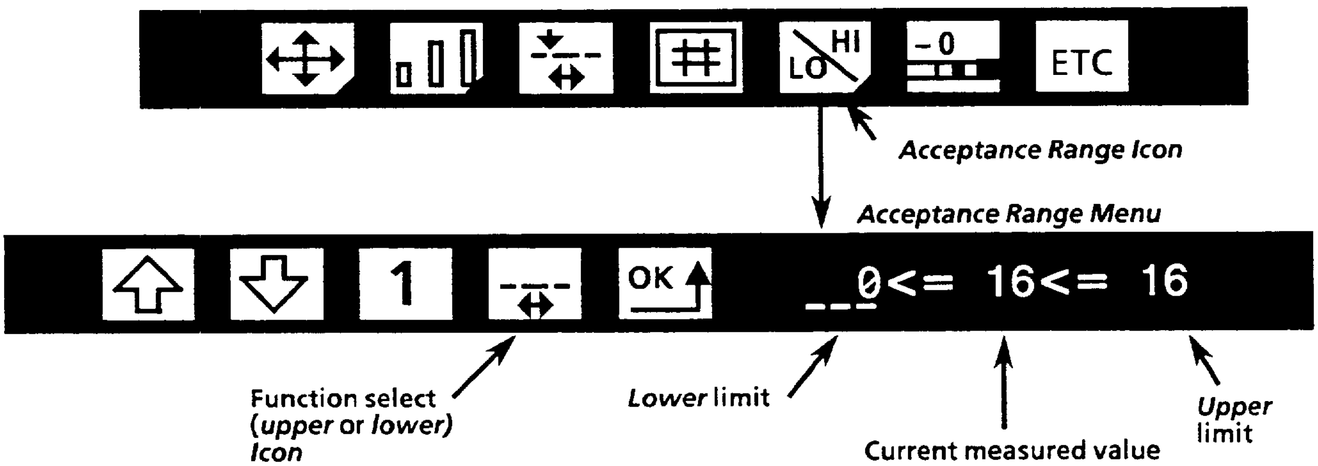
8. Adjust the acceptance range limits according to the statistical feedback. That is, set the acceptance ranges so that results for good parts will be within range, and results for bad parts will be out of range.

The next section tells how to use the *Acceptance Range* menu to set the upper and lower limits, or *acceptance range*, of the line gage functions.

Setting the Acceptance Range *(continued)*

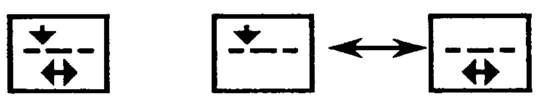
To set the *acceptance range* for a line gage function, access the *Acceptance Range* menu.

Your Action	*Comment*
Pick the Acceptance Range *icon on the* Line Gage *menu.*	



Your Action	*Comment*
Pick the Function Select *icon to select the* upper *or* lower *function.*	Select either the *upper* or the *lower* function of the line gage. The icon, when picked, toggles between the *upper* and *lower* function of the function icon selected on the *Line Gage* menu:

Selected *Gage Function* icon on *Line Gage* menu



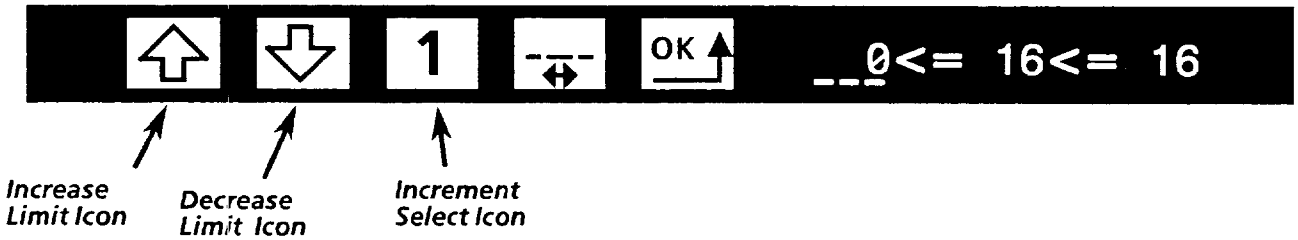
Function Select icon toggles between *upper* and *lower* function

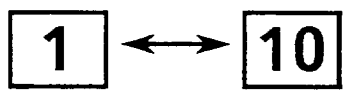
Your Action	*Comment*
Observe the function result.	On the right side of the *Acceptance Range* menu are three numbers, separated by "< =" signs. The *middle* number is the current measured *result* of the selected line gage function. For example, if you selected the upper function, the result of the line gage's upper function is displayed.
*Select the limit (*upper *or* lower*) you want to change.*	The *lower limit* is the *left* number of the three numbers displayed on the right side of the menu, the *upper limit* is the number on the *right* (**Note:** Default values are displayed for the limits initially). To select either limit, pick that limit with the light pen. The selected limit is *underlined.*

Selected limit is underlined

Setting the Acceptance Range *(continued)*

Acceptance Range Menu

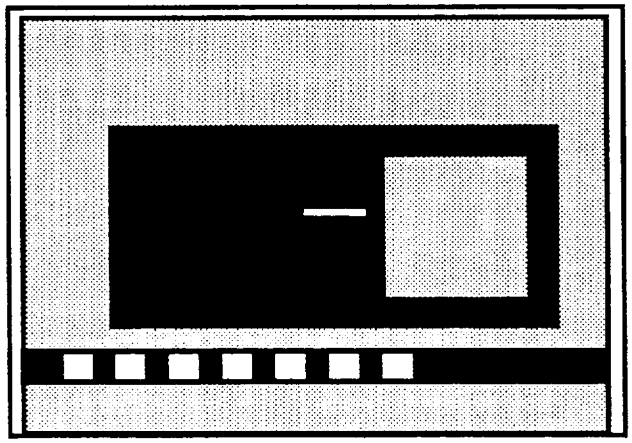


Your Action	Comments
Pick the Increment Select *icon to select the increment*	Picking the *Increment Select* icon toggles the icon between 1 and 10. 
Increase or decrease limit.	To *increase* the selected limit by the amount shown on the *Increment Select* icon, pick the *Increase Limit* icon. You can increase the limit to the maximum allowed by the line gage (this depends on the function, size of gage, and position of gage). To *decrease* the selected limit by the amount shown on the *Increment Select* icon, pick the *Decrease Limit* icon. You can decrease the limit to a value of 0. **Note:** The VIM2 does not allow you to set the upper limit to a value lower than the lower limit, or to set the lower limit to a value higher than the upper limit.
When finished, pick the OK *icon to display the* Line Gage *menu.*	After you have set both limits as desired, return to the *Line Gage* menu.

Setting the Acceptance Range (continued)

Example: Setting Acceptance Range

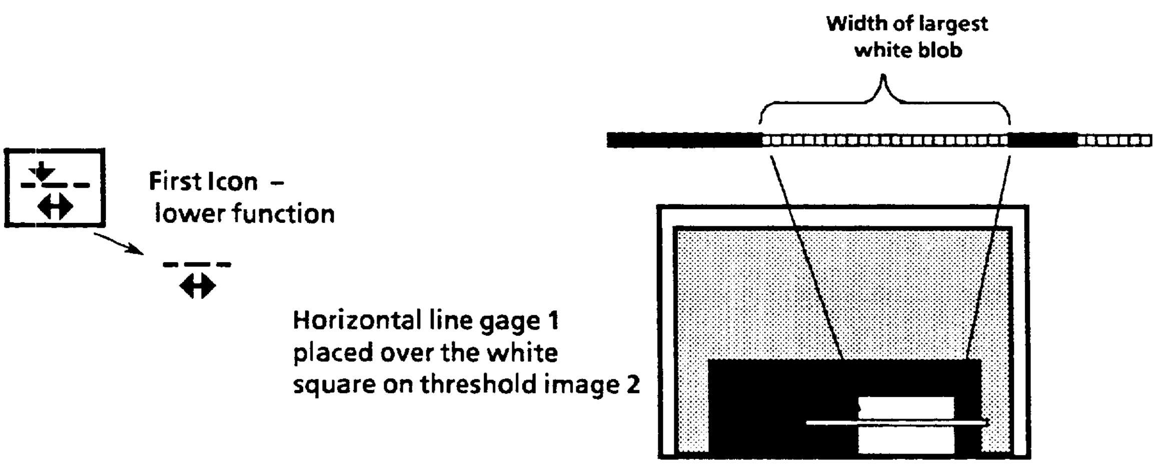
Let's suppose that the target pattern represents a part. Suppose you want to inspect the width of the light colored square on the part (shown as a white square within a black rectangle in threshold image 2, as illustrated below).



Line gage 1 displayed on threshold image 2 (example image)

Your Action	*Comments*
Set up line gage 1 using the Line Gage *menu icons.*	To gage the size of the white square, set up line gage 1 (using the *Line Gage* menu icons) as follows:

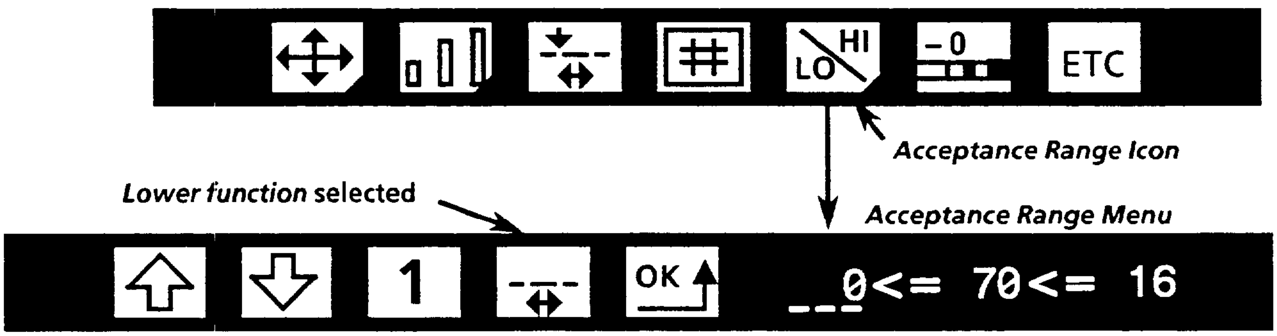
- Select *horizontal* orientation.
- Select *threshold image* 2.
- *Place* the line gage to cross and overlap the white square.
- Set *Gage Function* icon to the *first icon* (in order to use the *lower* function – *width of largest blob*).
- Set *Color* icon to white.



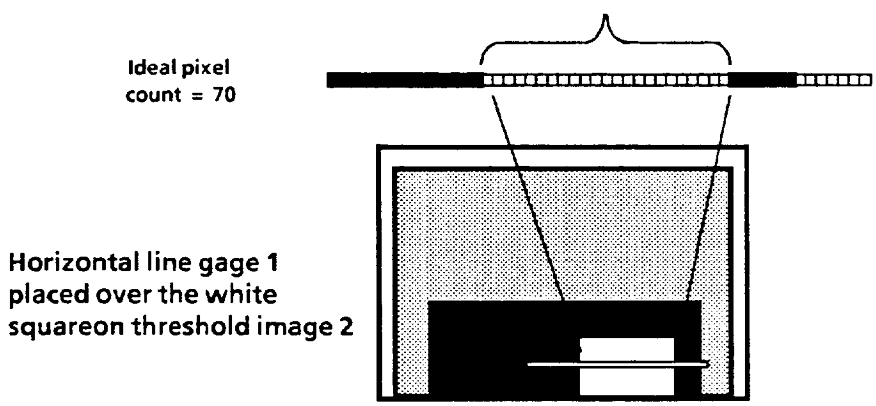
Setting the Acceptance Range (continued)

Example: Setting Acceptance Range *(continued)*

Your Action	*Comments*
Access the Acceptance Range *menu.*	With line gage 1 in place and configured as described on the previous page, pick the *Acceptance Range* icon.

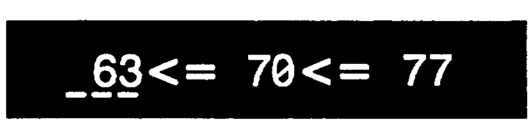


Select the lower *function*	With the *Acceptance Range* menu displayed, pick the *Function Select* icon so the *lower* function icon is displayed; that way, the current measured result (the *middle* number of the three numbers) for the *lower* function is given.
Observe the measured result and determine the limits.	Suppose the lower function result is 70 – that is, the width of the square, as measured by the line gage, is 70 pixels across.



Suppose the value of "70" is for the ideal part, and that you can accept any part within ±10%, or 70 pixels ±7 pixels. You then set the *upper* limit to 77, and the *lower* limit to 63.

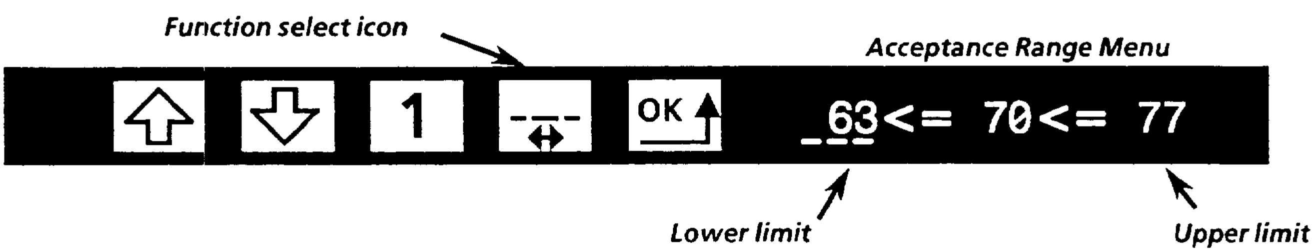
Select and set the upper *limit to 77.*	Pick the *upper* limit. Then pick the *Increase Limit* or *Decrease Limit* icons to change the limit to 77.
Select and set the lower *limit to 63.*	Pick the *lower* limit. Then pick the *Increase Limit* or *Decrease Limit* icons as necessary to change the limit to 63.



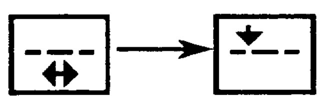
Setting the Acceptance Range (continued)

Example: Setting Acceptance Range *(continued)*
Remember that a gage performs two functions. You have set the acceptance range limits for one function (the *lower function*); you must now set the range for the *upper function* (which locates the left edge of the largest blob).

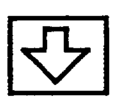
Suppose that, in this case, you do not need to use the *upper function* result. Then, you should set the limits of the unused to extreme values, so that the line gage does not fail due to the upper function result.



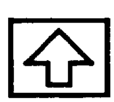
Your Action	*Comments*
Select the upper *function.*	Pick the *Function Select* icon on the *Acceptance Range* menu so the *upper* function is indicated. This allows you to set the upper function limits.



Select and set the lower *limit.*	Pick the lower limit. With the lower limit selected, pick and hold the *Decrease Limit* icon until the icon *flashes*; this indicates the lowest possible setting has been reached.



Select and set the upper *limit.*	Pick the upper limit. With the upper limit selected, then pick and hold the *Increase Limit* icon until the icon *flashes*; this indicates the highest possible setting has been reached.



With the upper function acceptance range set to as wide a range as possible, the line gage upper function will not fail unless the line gage fails to detect even a a single white pixel anywhere on the line gage; in that case, however, the lower gage function would have failed anyway.

Setting the Acceptance Range (continued)

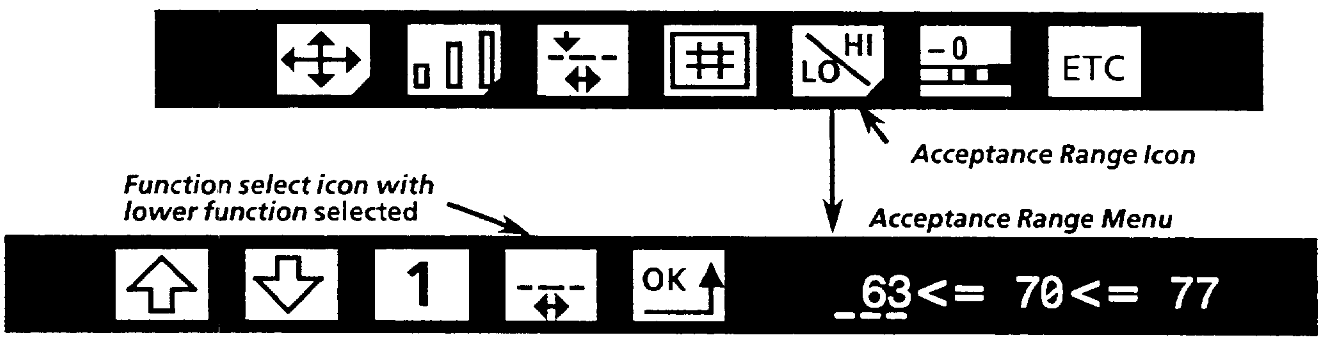
Using the Learn Mode and other Run Mode Functions
As mentioned previously in this chapter, the VIM2 module features a variation of its *run* mode called the *learn* mode. In *learn mode,* the VIM2 module accumulates *statistics for each enabled inspection tool, and the brightness probe,* based on a number of trial inspections. Use of the *learn* mode can be helpful in determining the optimal *acceptance range* for inspection tools and probe.

The VIM2 module also features a number of run mode *arm* options, which can be helpful in refining your configuration. See Chapter 13 for more information regarding use of the *learn* mode and other *run* mode options.

Accessing Statistics

When you run the VIM2 module in *learn mode,* you accumulate statistics for all enabled visions tools and probes. To observe the collected statistics for a line gage function, first access the *Acceptance Range* menu on the *Line Gage* menu, as described below:

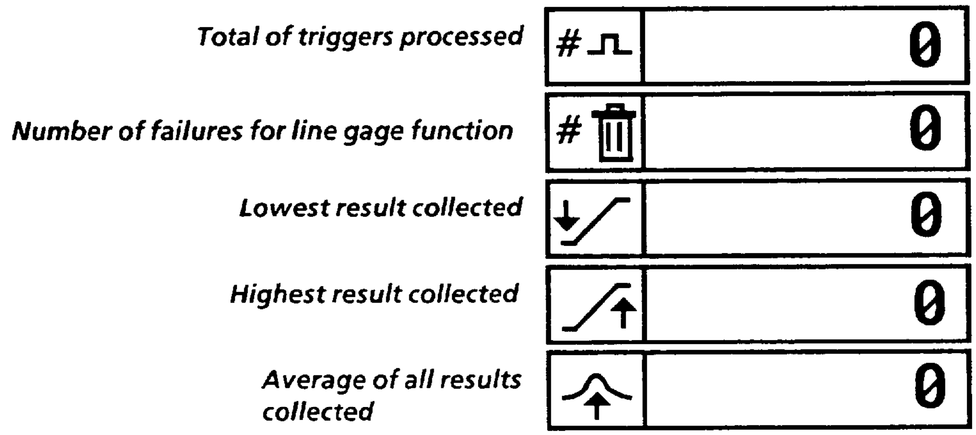
Your Action	*Comments*
Pick the Acceptance Range *icon on the* Line Gage *menu.*	



Select the upper/lower *function*	With the *Acceptance Range* menu displayed, pick the *Function Select* icon so the *desired* function icon *(upper or lower)* is displayed.
Pick the current measured result.	On the right side of the *Acceptance Range* menu are three numbers, separated by "< =" signs. The *middle* number is the *current measured result* of the selected line gage function. When you pick this number, the line gage disappears, and the statistics are displayed (see diagram on the next page).

Accessing Statistics
(continued)

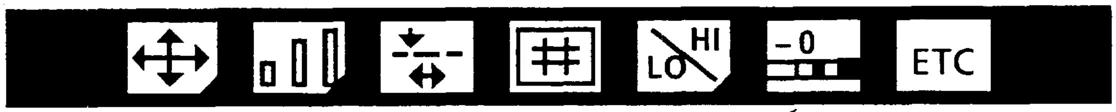
Note: The results of *failed* inspections for the line gage functions are not included in the *lowest*, *highest*, and *average* values. These statistics are based on passing inspections only. Therefore, for preliminary trial runs, you may want to set the acceptance range limits to extreme limits initially, for the purpose of collecting statistics for all inspections.



Your Action	Comments
Pick the current measured result again to remove the statistics display.	Picking the *current measured result* again removes the statistics display and redisplays the line gage.

Using the Line Gage Filter

The VIM2 line gages have a "filter" capability which you can activate using the *Gage Filter* icon on the *Line Gage* menu.



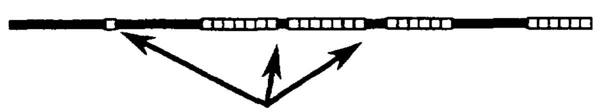
Gage Filter Icon

The line gage filter can help reduce or eliminate the effect of "noise" in the threshold images used by the line gages. Noise may appear in the image as "graininess" near the edges of features in the image (this can happen especially under "low-contrast" conditions, where there is not a great difference in the background brightness and the brightness of the feature of interest). Line gage filtering can be useful if noise appears in the image, or if the measured results of the line gage seem to be inconsistent, unsteady, or unexpected.

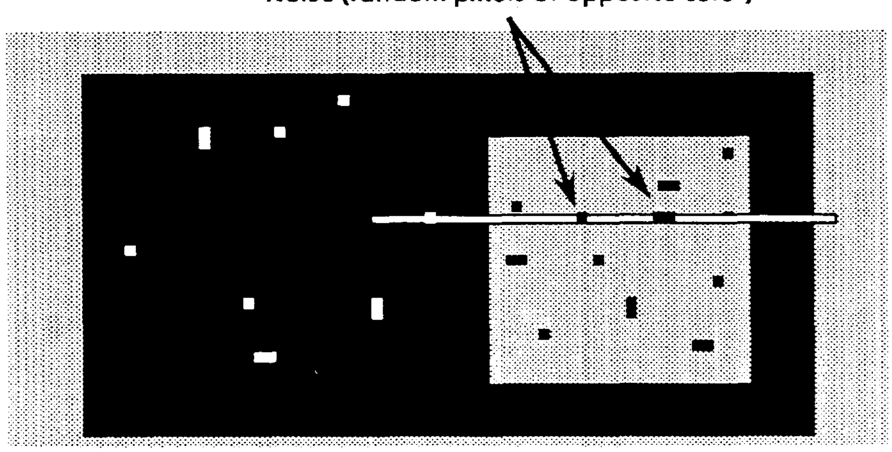
What is "noise?"

In some applications, there may be difficulty in establishing or maintaining an ideal, "clean" threshold image for use during inspection. That is, "noise" occurs in the image – random black pixels appear in white areas in the image, or random white pixels appear in black areas. These random pixels do not correspond to feature characteristics, and create variations which can be interpreted by the VIM2 as blobs, or breaks between blobs, where none should exist.

Line Gage detects pixel "noise"



Noise (random pixels of opposite color)



Line Gage shown on noisy threshold image

When noise in the image is crossed by a line gage, the line gage, without filtering, recognizes the noise as additional blobs or edges, which affects size or location measurement results, or affects the pixel counts.

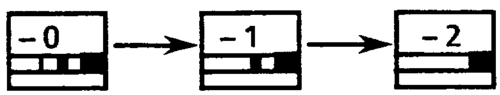
Using the Line Gage Filter (continued)

Note: Before using line gage filtering, eliminate noise in the image, as much as possible, by making adjustments in the level of threshold in the threshold image (see Chapter 8), or by making adjustments in the staging and lighting (see Chapter 4).

What does the line gage filtering do?
The line gage filtering allows the line gage to ignore or "filter" the noise. You can set the filtering to operate in one of two ways – *1-pixel* filtering, or *2-pixel* filtering.

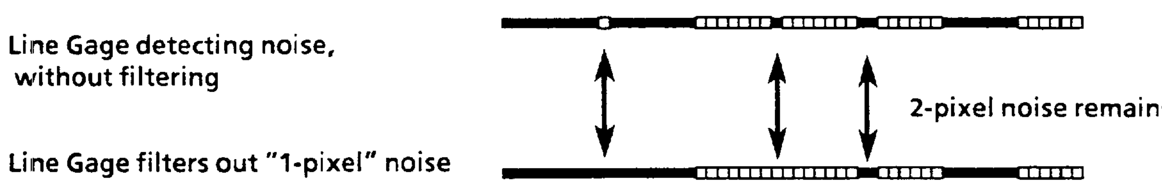
Note: There is no change in appearance of the line gage or the threshold image when you use line gage filtering.

You select the type of filtering by picking the *Gage Filter* icon. When picked, the icon cycles through three settings – *Filtering Off, 1 – pixel filtering, and 2 – pixel filtering*:

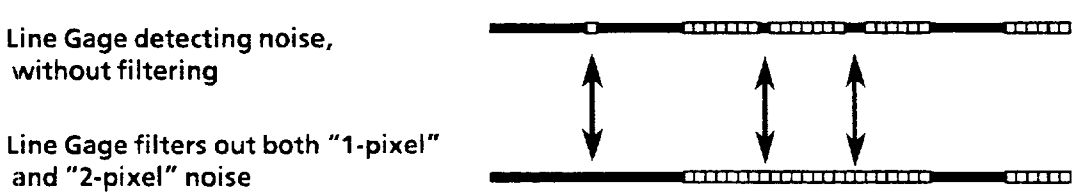


Filtering off *1-pixel Filtering* *2-pixel Filtering*

1 – pixel filtering:
The line gage ignores the occurrence of *one* white pixel within an otherwise all-black blob (or, one black pixel within an otherwise all-white blob).



2 – pixel filtering:
The line gage ignores the occurrence of *one or two* consecutive white pixels within a black blob (or, *one or two* black pixels within an otherwise all white blob).



Using the Line Gage Filter (continued)

How do you determine whether to use line gage filtering, and how much filtering to use?

It is a good practice to try using both 1- and 2-pixel filtering with any line gage, and use the filtering which yields the best results. As a general rule, after trial and observation, use the least amount of filtering which yields good results; *filtering can increase VIM2 module processing time per inspection.*

Chapter 10 Using Windows

Chapter Objectives

The objective of this chapter is to show how to use the *Window* menu icons to set up a window as part of a VIM2 module configuration for an inspection application. This chapter begins with questions and answers about windows.

Some Questions and Answers about Windows

This section introduces the topic of *windows* by posing some questions about windows, and answering them.

What is a window?
A window is a two-dimensional inspection tool which checks features in a selected area of a binarized (threshold) image.

What does a window do?
Unlike line gages, which can be set to operate in any of a number of ways, a window performs only one function – it counts the number of pixels (either black or white) in the area of the image over which a window is placed.

How many windows are available for use?
There are 4 windows (1-4), with 8 different window shape options available, including *rectangular*, *triangular*, *circular*, *donut*, and *train-thru-the lens mask* windows.

How do you set up a window?
Using the *Window* menu *icons*, you first select the window number (1-4), and enable the window. You then configure the selected window, which is shown on the monitor: you select window *shape*; select pixel *color*; *place* the window over the feature(s) of interest on the threshold image (**Note:** Each threshold image, 1-4, is always assigned to the window of the respective number); and then set an *acceptance range* for the window inspection results – this range provides the basis for an accept or reject decision.

How can you place a window?
Using the *Window* menu icons you can:

- Increase or decrease the window *height*;
- Increase or decrease the window *width*;
- Move the window up or down, and/or side to side, until positioned as required over the feature(s) of interest.

When do you set up a window?
Set up a window after you have:

- Set up the workstage, and have a workpiece in the nominal or ideal position (see Chapter 4), so that the workpiece features to be inspected are shown clearly in the field of view (and on the monitor).
- Set the threshold image(s) to enhance the particular feature(s) you are going to inspect (see Chapter 8).

Accessing the Window Menu

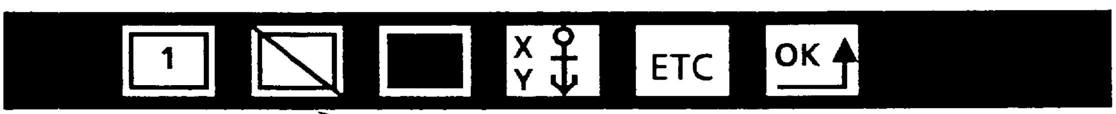
To begin to set up a window, first access the *window* menu as described below.

Your Action	*Comments*
Pick the Setup *icon on the* Main *menu to access the* Setup *menu.*	If the *Main* menu is displayed, access the *Setup* menu by picking the *Setup* icon (the "open padlock"). If the *Setup* menu is already displayed (as shown below), skip this step.

Setup Menu

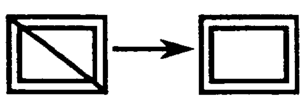
Window Icon

Pick the Window *icon.*	Picking the *Window* icon on the *Setup* menu displays the *Window* menu (Note: The *Window* menu has two parts; the first part is displayed when you pick the *Window* icon):



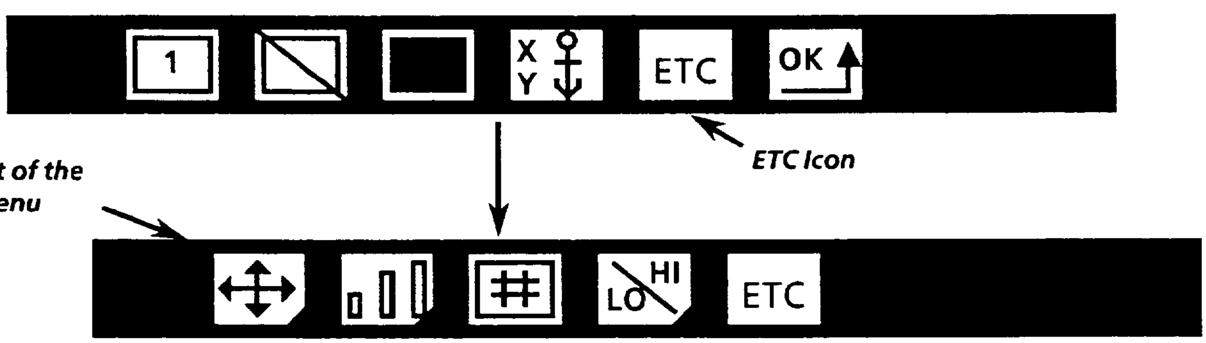
Enable / Disable Icon

Pick the second (enable/disable) *icon.*	Picking the *enable / disable* icon enables the window, and allows you to pick other icons on the *Window* menu.



When you pick this icon, the diagonal slash disappears.

Pick the ETC *icon.*	Picking the *ETC* icon displays the second part of the *Window* menu:

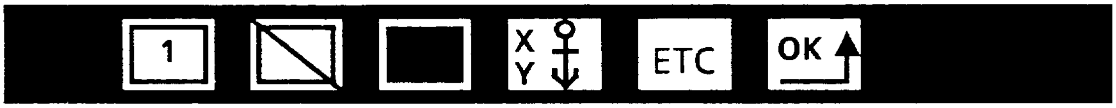


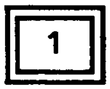
ETC Icon

Second part of the Window menu

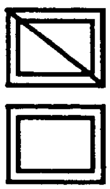
The Window Icons

Here are brief descriptions of each of the *Window* icons, beginning with the first part of the *Window* menu:



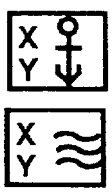


Window Select – Pick this icon to select the window from a choice of numbers 1-4. Numbers 1-4 are displayed in sequence as you repeatedly pick the icon, or hold the light pen tip down against the icon. The selected window is displayed on the monitor.

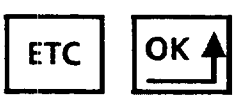


Window Enable – Pick this icon to enable or disable the selected window. This icon, when picked, toggles between *enabled* or *disabled* (*disabled* is indicated by diagonal slash across the icon). You must enable a selected window in order to use the remaining *Window* icons to configure a window. Also, a window must be enabled in order to be part of your inspection configuration.

Window Shape– Pick this icon to select window shape; there are 8 different shapes available. This icon, when picked, toggles through the 8 shape choices – rectangular, triangular (4 different triangle orientations), circular, donut, and "train-through-the-lens mask."



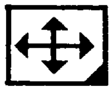
Anchor / Float – Pick this icon to enable or disable position shift compensation. This icon, when picked, toggles between enabled (indicated by wavy lines) or disabled (indicated by anchor).



ETC and OK –The other two icons displayed on the menu are the familiar *ETC* icon and *OK* icon. Recall that picking the *OK* icon returns you to the previous menu (the *Setup* menu in this case). Picking the *ETC* icon displays additional *Window* menu icons (see next page).

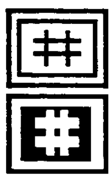
The Window Icons (continued)

If you pick the *ETC* icon on the *first part* of the *Window* menu, the second part of the *Window* menu appears:



Window Move – Pick this icon to access the *Window Move* menu for the selected window. Use the *Window Move* menu to strategically position the window on the image shown on the monitor screen.

Window Size – Pick this icon to access the *Window Size* menu for the selected window. Use the *Window Size* menu to increase or decrease the size of the window.



Color – Pick this icon to select the color – either black or white – of the pixels to be counted by the window inspection. This icon, when picked, toggles between *white* (white box with black "#") and *black* (indicated by the black box with white "#").

Acceptance Range – Pick this icon to access the *Acceptance Range* menu for the selected window. Use the *Acceptance Range* menu to set the upper and lower limits for the inspection results, to read the current window result, and to read window statistics.

ETC – If you pick the *ETC* icon on the second part of the *Window* menu, the first part of the *Window* menu returns to the screen.

Exiting the Window Menu

Remember, to exit the *Window* menu, you must return to the first part of the *Window* menu, which includes the *OK* icon. Then, pick the *OK* icon to exit the *Window* menu and return to the *Setup* menu.

Configuring Windows

This section explores the features and options you will encounter when configuring a window, and leads you through procedures for setting up a window for inspection. **Note:** For example purposes, these procedures refer to the target pattern included with this manual. See the "Quick Start" section of Chapter 6 to set up the VIM2 module, peripheral equipment, and target pattern.

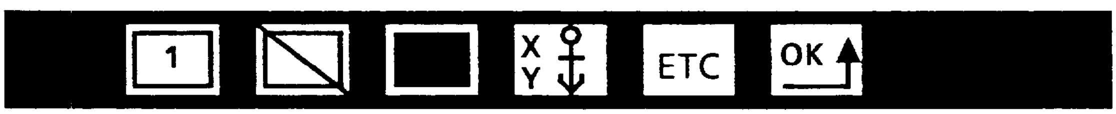
Access the Window Menu

Your first step is to access the *Window* menu.

Your Action	Comments
Pick the Setup *icon on the* Main *menu to access the* Setup *menu*	If the *Main* menu is displayed, pick the *Setup* (the "open padlock") icon to access the *Setup* menu (if the *Setup* menu is already displayed, skip this step):

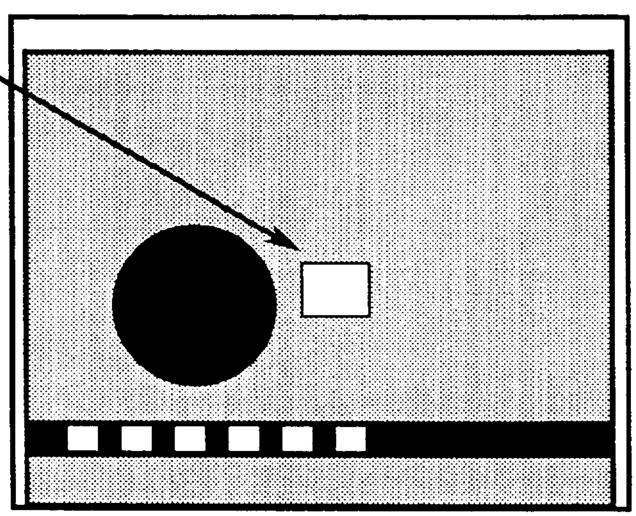
Window Icon

Your Action	Comments
Pick the Window *icon to access the* Window *menu.*	When you pick the *Window* icon, the *Window* menu appears:



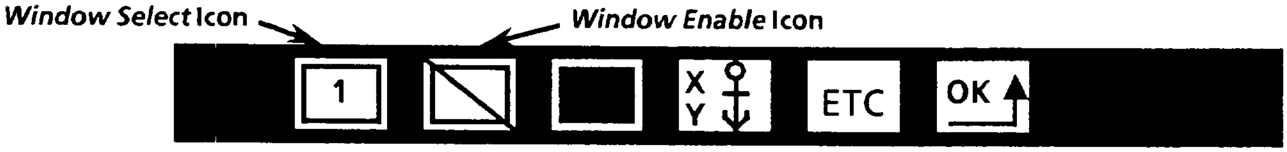
When you select the *Window* menu, a window is displayed on the monitor. The first time you use the *Window* menu, the window is displayed in the default position in the center of the screen, with rectangle shape (see illustration below). The threshold image is number "1" (the threshold image number is always the same as the window number).

Window displayed on threshold image 1 (example image).



Select and Enable Window

Select a window to configure using the *Window Select* icon.



When you pick the *Window Select* icon, the number increments through all the windows – 1, 2, 3, 4, 1, 2, etc. The selected window is displayed on the monitor.

Note: In selecting a window, be aware that you are also selecting the *threshold image* you will use. This is because each window is linked to the threshold image of the same number; that is, window 1 is always placed on threshold image 1, window 2 uses threshold image 2, etc. You cannot otherwise select a threshold image for a window.

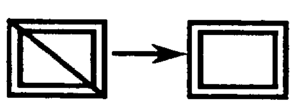
Window Enable – In order to configure a window, you must *enable* the window. Pick the *Window Enable* icon so the diagonal slash is removed, indicating the window is enabled.

Note: A window must be *enabled* in order to be part of your inspection operation. You can disable a window after you have configured it – when *disabled,* a window does not function as part of the configuration (**Note:** A window cannot be *disabled* if it is included in a math tool configuration – see Chapter 12).

Your Action	***Comments***
Pick the Window Select *icon to change the window number.*	Picking the *Window Select* icon (the first icon on the left) changes the window number. To cycle through the windows rapidly, hold the light pen tip down against the icon.

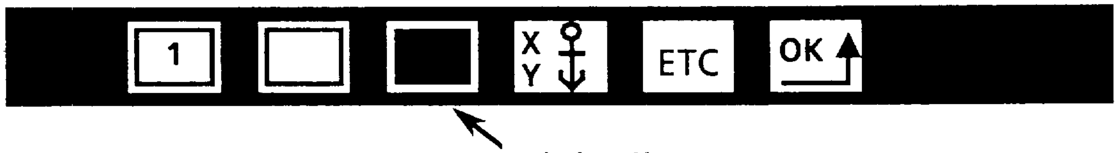
Note: As you cycle through the windows 1-4, notice you are cycling through the threshold images 1-4 at the same time.

Pick the Window Select *icon so that "2" is displayed.*	**Example: Select and Enable Window 2** With "2" selected, window number 2 is displayed over threshold image number 2.
Pick the Window Enable *icon to enable the window.*	When the diagonal slash disappears, this indicates the window is enabled.



Select Window Shape

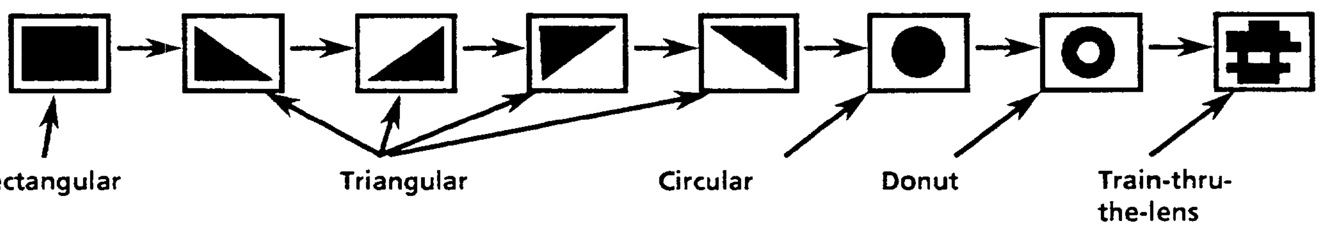
Each window can be assigned a specific shape; you select the window shape by picking the *Window shape* icon.



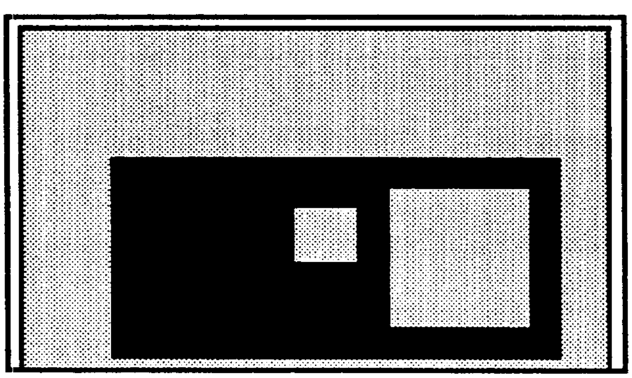
Windows are *rectangular* when first selected. As you pick the *Window Shape* icon, the icon cycles through eight shape selections, and the shape of the displayed window changes to match the icon, except for the *train-thru-the-lens mask* (**Note:** The *train-thru-the-lens mask* window is a special case of window shape, which allows you to design the shape of the window. This is discussed later in this chapter).

Select the *window shape* according to the application requirements.

Your Action	*Comments*
Pick the Window Shape *icon.*	Picking the *Window Shape* icon repeatedly toggles through the eight window shapes.



	Example – Select Rectangular Shape
Select rectangular.	Pick the *Window Shape* icon so the *rectangular* window is displayed on the screen.



Rectangular window 2 displayed on threshold image 2.

Select Window Shape (continued)

Note: The VIM2 module stores the window shape selections as different windows in terms of window placement. For example, if you select *rectangular*, then change the position of the window, then select *circular*, the window shape changes to *circular*, and the window returns to its home (default) position in the center of the screen.

Select Anchor or Float Status

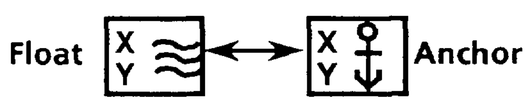
Anchor or *Float* status refers to VIM2 module's X/Y shift compensation capability. Use the *Anchor / Float* icon to enable or disable the shift compensation feature.

Anchor / Float Icon

X/Y shift compensation – the VIM2 module can be configured to adjust for shifts of the workpiece in the image from inspection to inspection, using the "X" and "Y" float windows (the VIM2 module's X/Y shift compensation is discussed in detail in Chapter 11).

When configuring a window, select *Float* to *enable* the window to use the shift compensation capability. With *Anchor* selected, shift compensation is *disabled.*

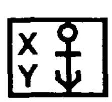
Your Action	*Comments*
Pick the Anchor / Float icon.	Picking the *Anchor / Float* icon toggles the window status between *anchor* (indicated by the anchor in the icon) and *float* (indicated by the three wavy lines).* As you toggle the icon, the appearance of the window does not change.



Note: In order to select the *float* status for a window, either or both of the "X" and "Y" float windows must first be enabled (if the *Anchor / Float* icon does not toggle, this is because both the "X" and "Y" float windows are disabled).

Example – Select Anchor

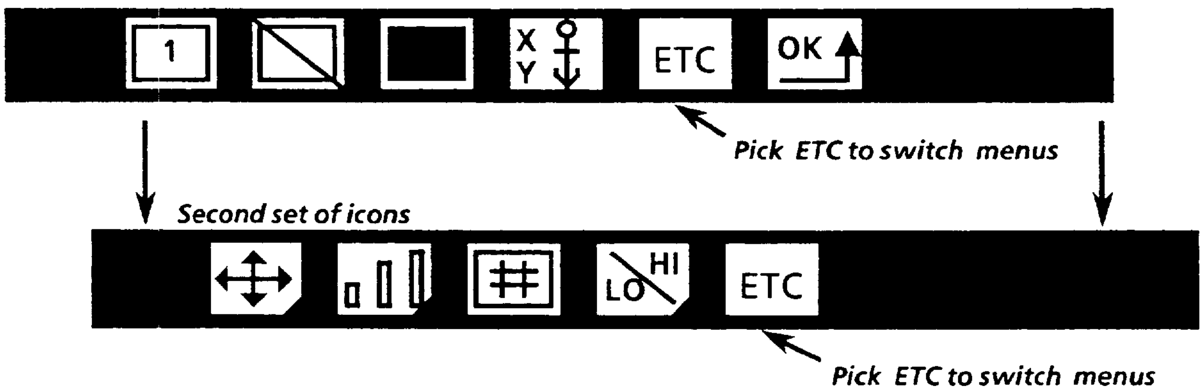
Select Anchor Status — Pick the *Anchor / Float* icon so that the *Anchor* is displayed.



Additional Window Icons

At this point, go on to the *second* part of the *Window* menu to continue. To do this, use the *ETC* icon.

Your Action	Comments
Pick the ETC *icon to access the second set of* Window *menu icons.*	When you pick the *ETC* icon, the second part of the *Window* menu is displayed (**Note:** Pick *ETC* on the second menu to redisplay the first).

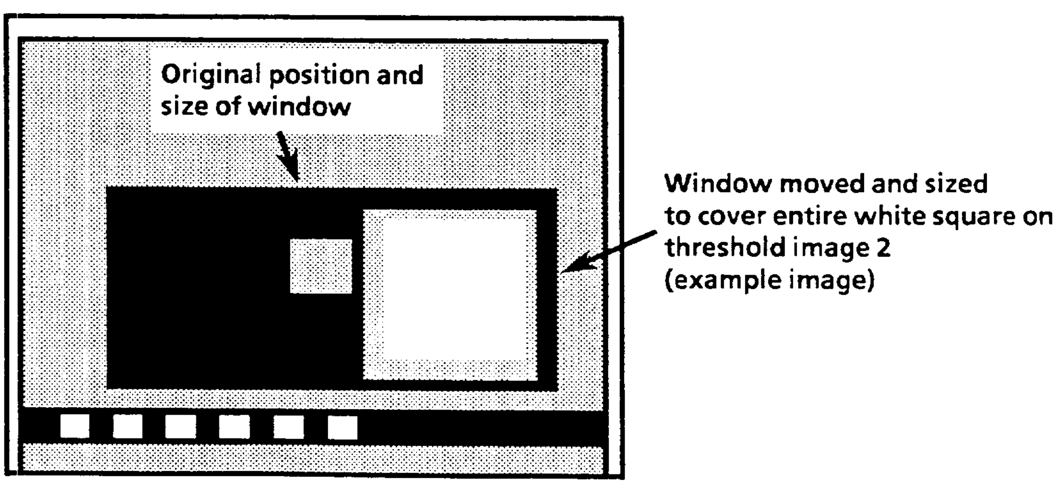


Moving the window

In order to use a window for inspection, you must *place* that window over the feature of interest you want to inspect.

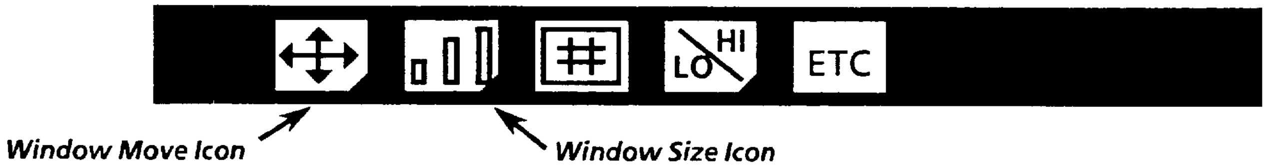
Generally speaking, you should place a window so that the window covers the feature to be measured with some overlap. To place a window, you probably will need to both move the window (using the *Window Move* menu), and adjust the window size (using the *Window Size* menu).

This section discusses using the *Window Move* menu to place a window in position for an inspection. Using the *Window Size* menu is discussed in the next section.



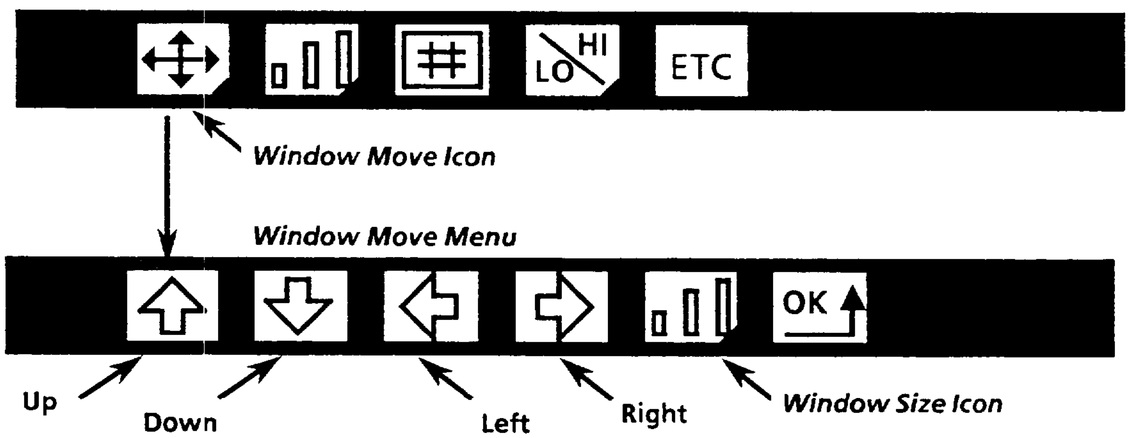
Moving the Window
(continued)

The first two icons on the second *Window* menu are the *Window Move* icon (used to access the *Window Move* menu), and the *Window Size* icon (to access the *Window Size* menu).



First let's look at the *Window Move* menu.

Your Action	*Comments*
Pick the Window Move *icon to access the* Window Move *menu.*	The *Window Move* icon is the first icon on the second *Window* menu (indicated by the four directional arrows).



The *Window Move* menu includes four directional arrow icons. Notice the *Window Move* menu also includes the *Window Size* icon for immediate access to the *Window Size* menu (the *Window Size* Menu has a *Window Move* icon as well). The VIM2 window menus allow you to conveniently toggle back and forth between the *Window Size* and *Window Move* menus while placing the window.

To *move* the window into the desired position, pick the appropriate arrow icon(s), in one of two ways:

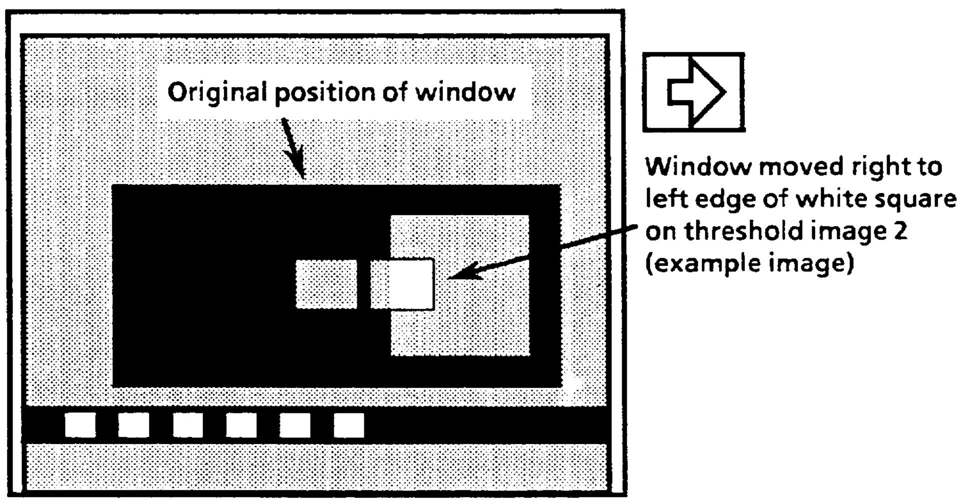
- *Pick* the arrow icon to move the window one pixel in the direction indicated by the icon.
- *Pick and hold* the light pen down against the arrow icon to cause *continuous movement* in the indicated direction
- If you pick an arrow icon and it flashes, this indicates that the window is at a boundary – such as the screen edge –and cannot be moved further in that direction.

Moving the Window
(continued)

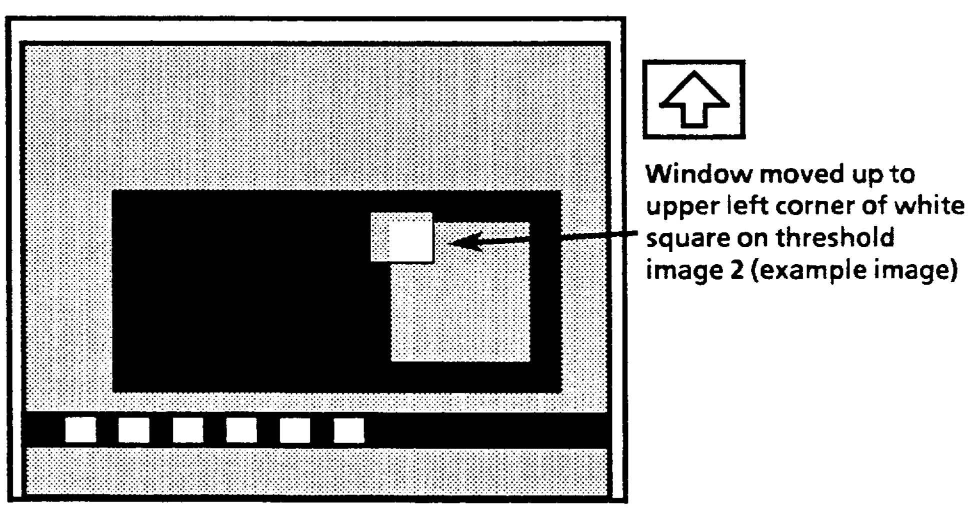
Note: Depending on the *brightness compensation* mode you have selected, placement of the window may be restricted (see Chapter 8 for brightness compensation information). Briefly, if you have selected mode 1 (*same frame* compensation), you cannot place a window in the top 48 pixel rows (or about the top 20%) of the video image.

Example Window Move
This demonstration places a *rectangular-shaped* window 2 at the upper left corner of white square in threshold image 2.

Your Action	*Comments*
Place the window across the left side of the white square *in the target.*	Pick the *right* arrow icon to move the window until the window crosses the left side of the white square.



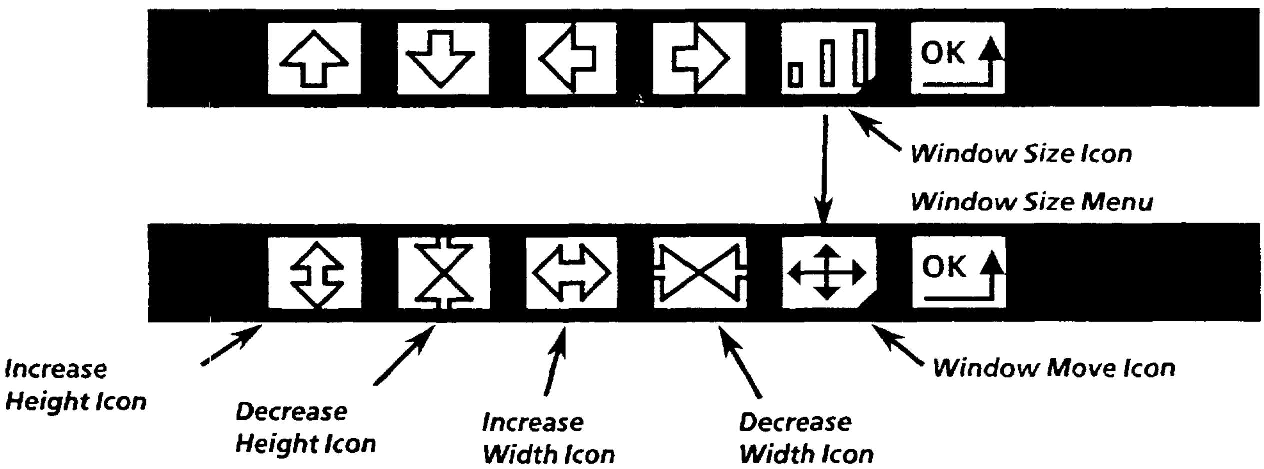
Place the window across the upper left corner of the white square *in the target.*	Pick the *up* arrow icon as necessary to move the window so the window crosses the top side of the white square.



Sizing the Window

Use the *Window Size* menu to adjust the window size to cover the feature of interest in the threshold image. You use the *Window Size* menu icons to increase or decrease the *height* and/or the *width* of a window. The method for changing the window size differs with each window shape.

Your Action	*Comments*
Pick the Window Size *icon on the* Window Move *menu.*	If the *Window Move* menu is displayed, access the *Window Size* menu by picking the *Window Size* icon:



Note: Remember you can also access the *Window Size* from the *Window* menu by picking the *Window Size* icon on that menu. Also, notice the *Window Move* icon, which can be used for access to the *Window Move* menu.

As illustrated above, there are four *Window Size* menu icons: *Increase Height, Decrease Height, Increase Width*, and *Decrease Width.* The actual *function* of these icons differs, depending on the selected *shape* of the window (these differences are explained throughout in this section).

There are two ways of using the *Window Size* icons for adjusting the window size:

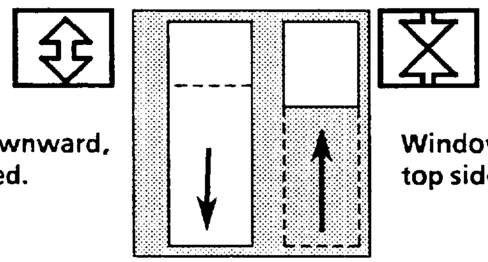
- *Pick* the appropriate *arrow* icon to decrease or enlarge the window size in a given direction, *one pixel at a time.*
- *Pick and hold* the light pen down against the appropriate icon to cause *continuous enlarging or decreasing* of size.
- If you pick an arrow icon and it flashes, this indicates that the window is at its limit and cannot be further adjusted in the direction indicated by the selected icon.

Sizing the Window
(continued)

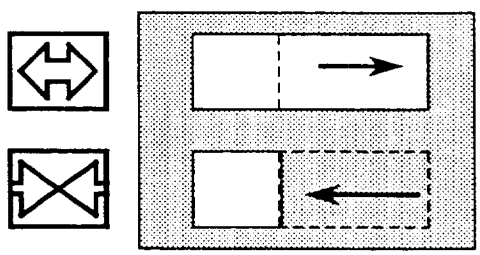
Rectangular Window Sizing

When you change the size of a *rectangular* window, the top left corner of the window remains fixed; only the position of the right or bottom side of the window changes.

To change the height of a rectangular window, pick the *increase height* icon to *increase height* downward, or pick the *decrease height* icon to *decrease height* upward:



To change the window width, pick the *increase width* icon to *increase width* to the right, or pick the *decrease width* icon to *decrease width* from the right:



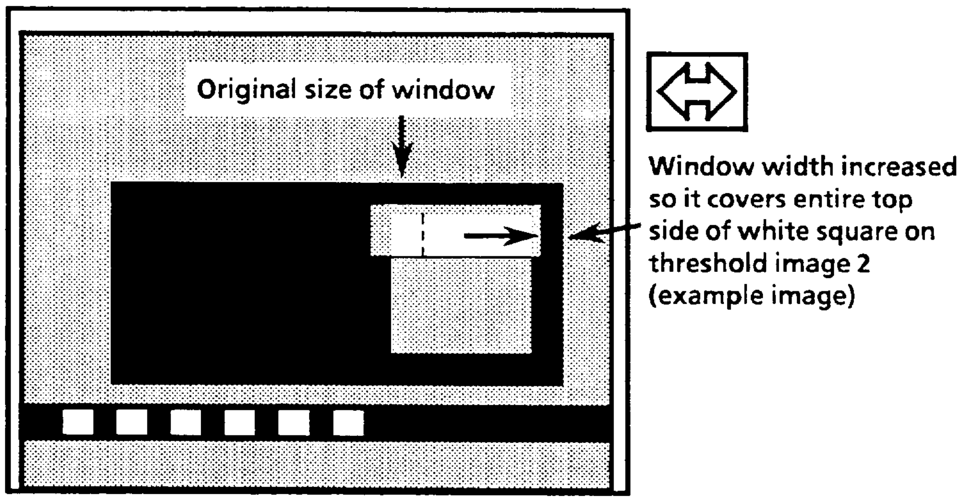
Rectangular Window Size Limits

- You can increase a rectangular window's size until it meets the bottom edge (for a *height increase*) or right edge (for a *width increase*) of the screen.
- You can decrease a rectangular window's size to a *single pixel* in height and/or width.

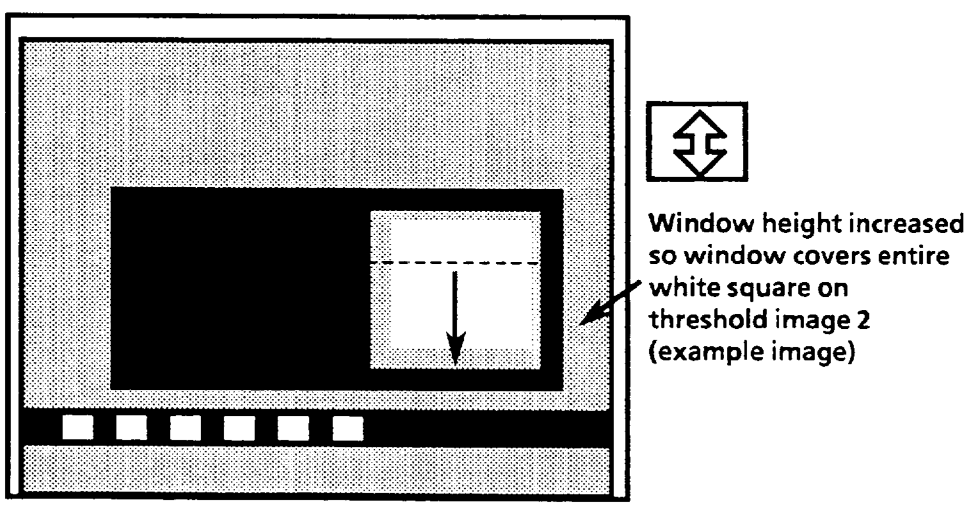
Sizing the Window (continued)

Example Size Adjustment of a Rectangular Window
Adjust both the width and height of the rectangular window so it covers the entire white square on the target pattern in threshold image 2 (**Note:** This procedure assumes you have moved the rectangular window 2 to the top left corner of the white square as shown on page 10-11).

Your Action	*Comments*
Increase the window width by picking and holding the Increase Width *icon.*	Increase the window width until it covers the top side of the white square on the target pattern in threshold image 2:



Increase the window height by picking and holding the Increase Height *icon.*	Increase the window height until the window covers the entire white square in threshold image 2:

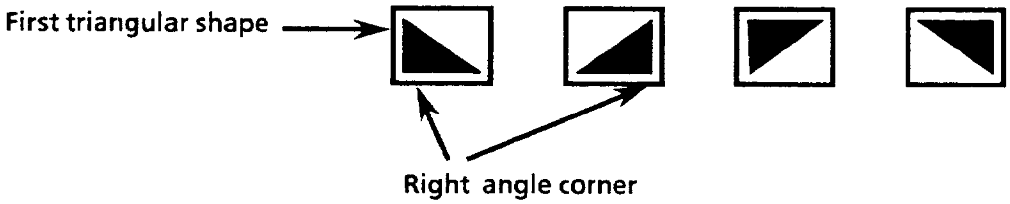


Pick the OK *icon.*	After placing the window, return to the *Window* menu by picking the *OK* icon.

Sizing the Window
(continued)

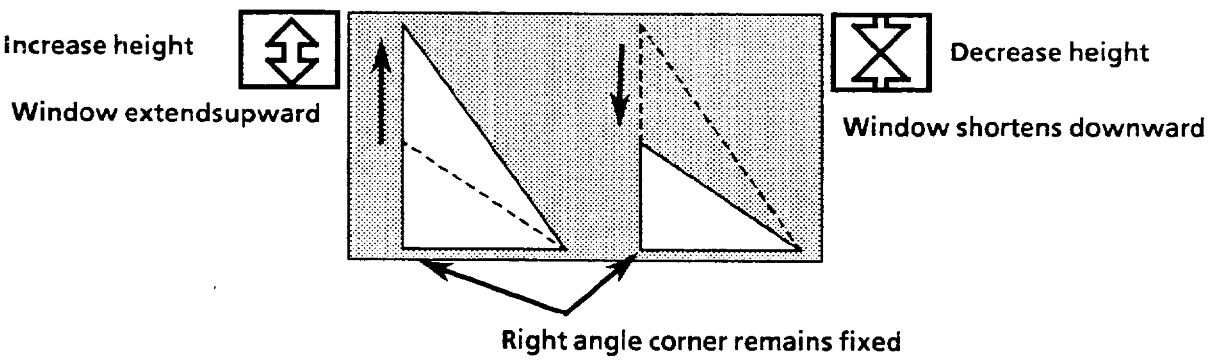
Triangular Window Sizing

There are four *triangular* windows shapes. Each *triangular* shape is a right triangle; the difference is in the orientation of the *right angle corner* of each. The four selectable triangular shapes are shown below in order of selection:

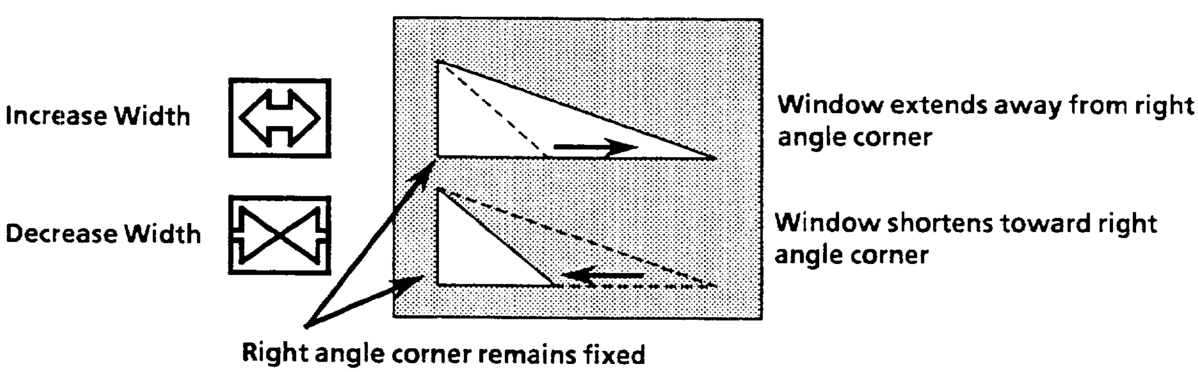


When you change the size of a *triangular* window, the position of the *right angle corner* of the window remains fixed; the lengths of each of the two sides adjacent to the *right angle corner* can be increased or decreased.

For example, to change the height of a first triangular shape window, pick the *increase height* icon to increase height *away from* the right angle corner, or pick the *decrease height* icon to decrease height *toward* the right angle corner.



To change the width, pick the *increase width* icon to increase width *away from* the right angle corner, or pick the *decrease width* icon to decrease width *toward* the right angle corner:



Sizing the Window *(continued)*

Triangular Window Sizing *(continued)*

The procedures for adjusting the size of the remaining three triangular shapes, you will find, are similar to the above procedure for the first triangular shape, so these procedures are not discussed further here.

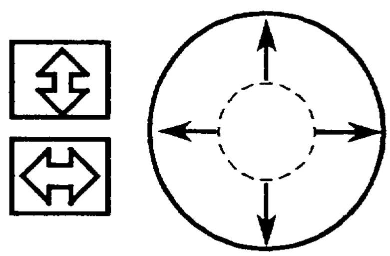
Triangular Window Size Limits

- You can increase a triangular window's size until it meets a top or bottom edge (for a *height increase*) or right or left edge (for a *width increase*) of the screen.
- You can decrease a triangular window's size to a *single pixel* in height and/or width.

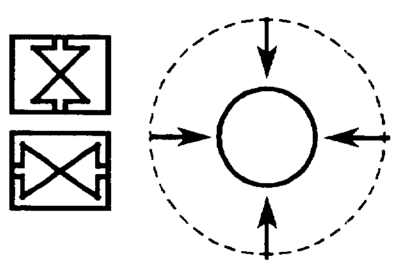
Circular Window Sizing

When you change the size of a *circular* window, the center of the window remains fixed; the radius and area of the circle increase or decrease.

To increase the area of a circular window, pick either the *increase height* icon or the *increase width* icon – both work the same way to increase the area of circular windows.



To decrease the area of a circular window, pick either the *decrease height* icon or the *decrease width* icon – both work the same way to decrease the area of circular windows.



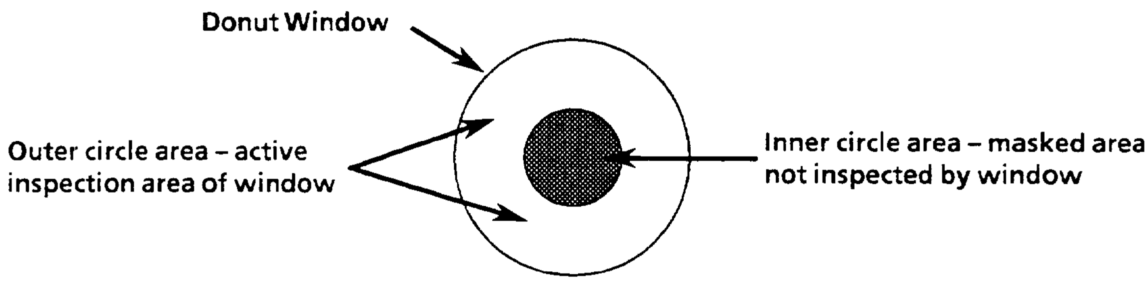
Circular Window Size Limits

- You can increase a circular window's size until it meets a top or bottom limit or right or left edge of the screen.
- You can decrease a *circular* window's size until it becomes a small *square (3 pixels × 3 pixels in size).*

Sizing the Window
(continued)

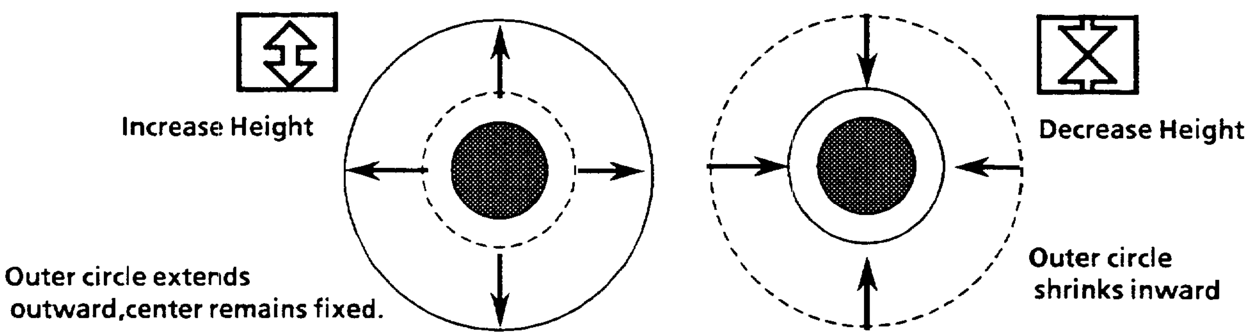
Donut Window Sizing

When sizing a *donut* window, you change the size of two parts of the window: 1) the outer white circle – the active inspection area of the window, and 2) the inner dark circle, which masks out the part of the outer circle area it covers – the window does not inspect this area (the "donut hole").

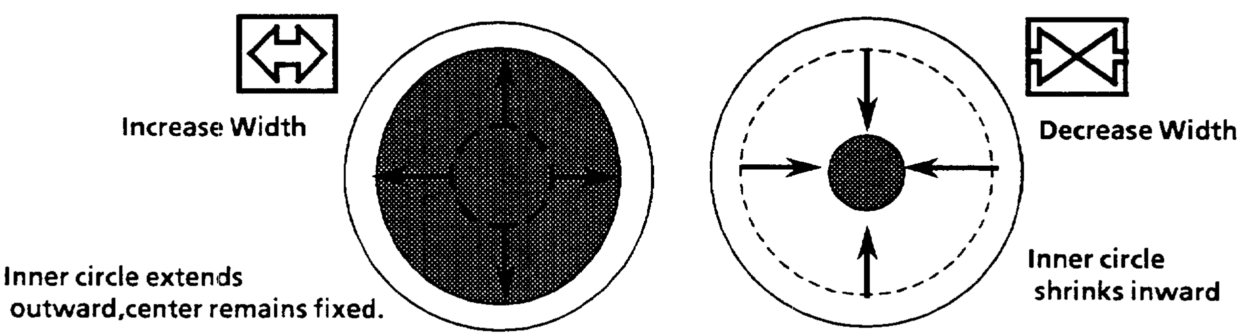


When you adjust the size of either the inner or the outer circle, the circle center remains fixed; the radius and area of the circles increase or decrease.

Outer circle – To adjust the area of the outer circle of the donut window, pick the *increase height* icon to increase the area, or pick the *decrease height* icon to decrease the area.



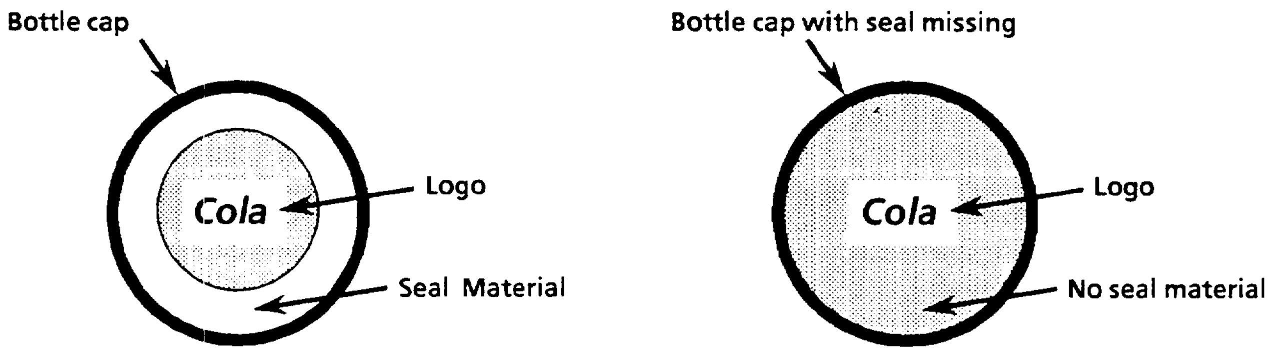
Inner circle – To adjust the area of the **inner circle** of the donut window, pick the *increase width* icon to increase the area, or pick the *decrease width* icon to decrease the area.



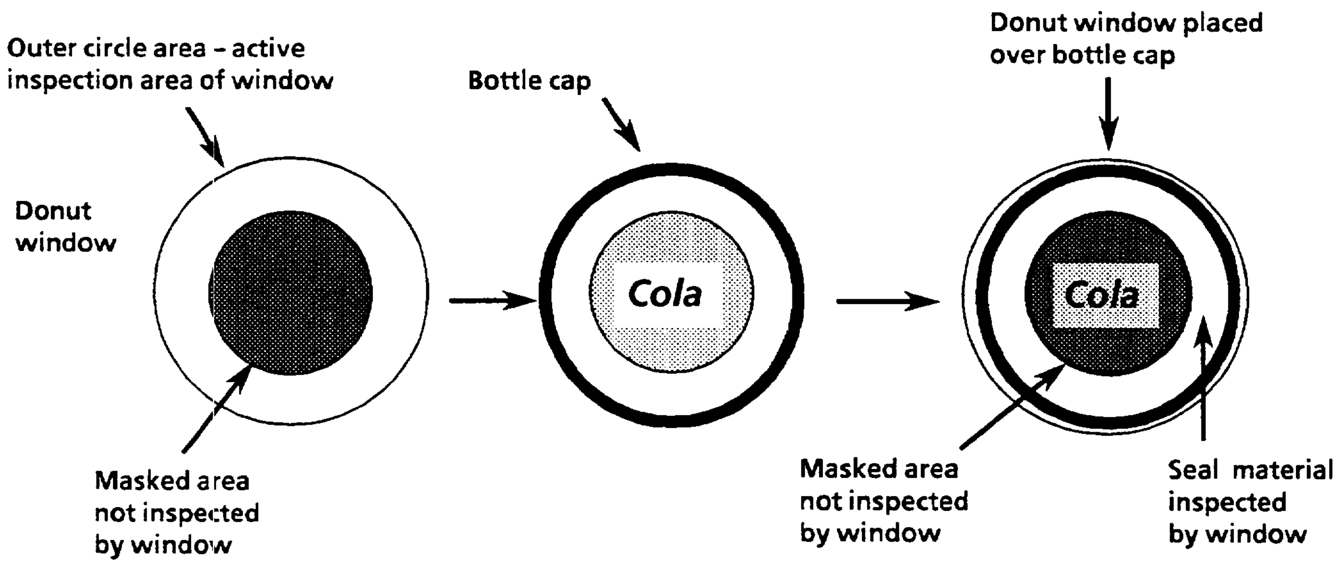
Sizing the Window (continued)

Example Application of the Donut Window

Suppose you are inspecting bottle caps before they are placed on the bottles. You want to check for the presense of a seal on the underside of the cap, and need to use a circular-shaped window to count pixels. However, on some bottle caps there is a logo placed on the bottle cap underside, but on others no logo is placed there. This inconsistency would cause a variation in pixel count from cap to cap when using a circular-shaped window, and could make setting the acceptance range for the window difficult, or impossible.



However, using a donut window, you could mask out the part of the bottle cap where the label appears, in order to count pixels only where the seal is placed. This would create much more consistency in pixel count from cap to cap – the presense or non-presense of a logo would have no effect on the pixel count – and allow you to set the window acceptance range to a workable range.



Sizing the Window (continued)

Donut Window Size Limits

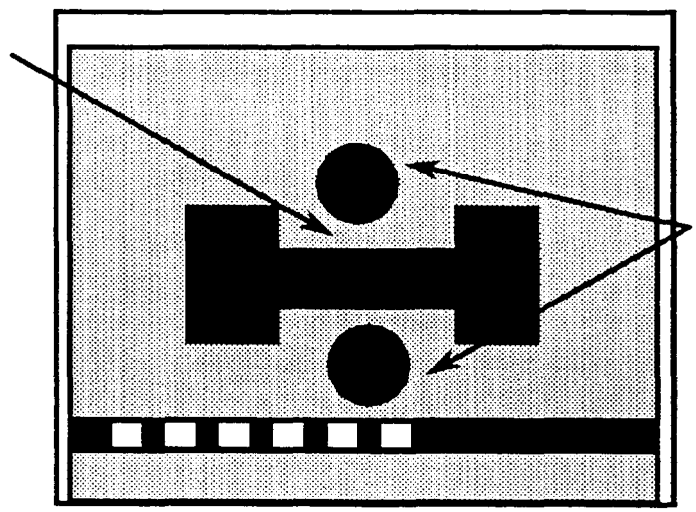
- You can increase the size of the *outer circle* until it meets a top or bottom limit or right or left edge of the screen.
- You can decrease the *outer circle's* size until its radius is within 4 pixels in length of the inner circle's radius – that is, the outer circle always has a radius that is at least 4 pixels longer than that of the inner circle.
- You can increase the *inner circle's* size until its radius is within 4 pixels in length of the outer circle's radius.
- You can decrease the *inner circle* until it becomes a small *square* (3 pixels × 3 pixels).

Train-thru-the-lens Mask

The last of the selectable window shapes is the *train-thru-the-lens mask* window (**Note:** This selection is available for windows 1 to 3 only). This selection is really not a shape, but an option which allows you to *custom design* the size and shape of a window. This feature can be especially useful when none of the preset window shapes is satisfactory for a specific feature of a workpiece you want to inspect.

For example, suppose your workpiece appears in the threshold image as shown below:

Rectangular part of the threshold image to be inspected (example image).



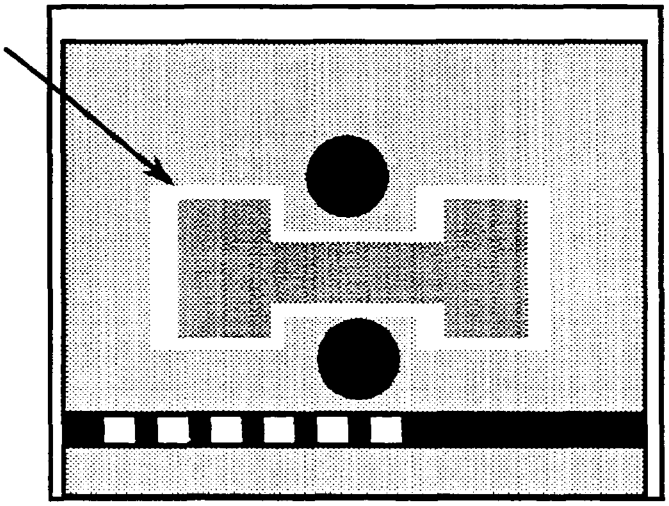
Circular parts of the threshold image to be ignored.

Now suppose you to want to set up a window to inspect only the central, somewhat rectangular feature of the workpiece, and to ignore the two circular features in the image. In this case, a *preset window shape* would not be the solution, because not one of the preset window shapes could be placed so that it would cover all of the rectangular features, yet not cover any of the circular features. In this case, a *train-thru-the-lens mask* window would be required.

Train-thru-the-lens Mask Window (continued)

In this case, the window you design using the *train-thru-the-lens mask* window might look like the one shown below:

Custom window designed using the train-thru-the-lens mask option.

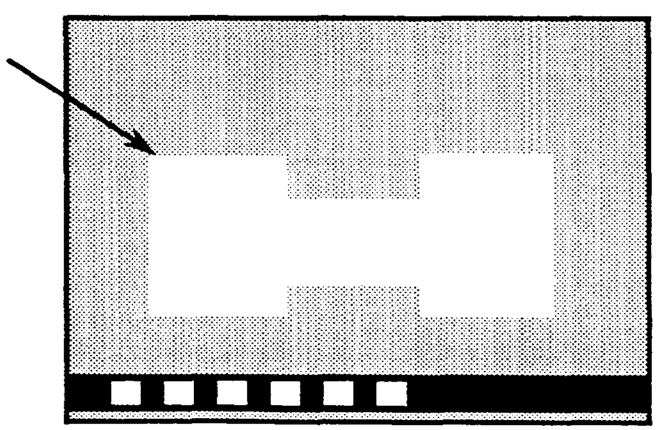


Only the area covered by the window (shown in white) would be inspected. In this case, only the somewhat *rectangular* feature, and not the *circular* features, would be inspected.

Example – Creating a *train-thru-the-lens mask* window
This section leads you through a sample procedure for creating a *train-thru-the-lens mask* window, based on the case of the example workpiece illustrated above.

Your Action	***Comment***
Select the window *(and* threshold image*) number.*	After staging and lighting the workpiece, and setting up a threshold image to highlight the feature(s) of interest for the inspection, select the appropriate window (and thus, the threshold image) which highlights the feature(s) of interest.
Determine the shape *and* size *required for the window.*	Upon observing the workpiece as it appears in the threshold image, determine the required window size and shape. In our example case, the required window would look like this:

Required window size, shape, and position.

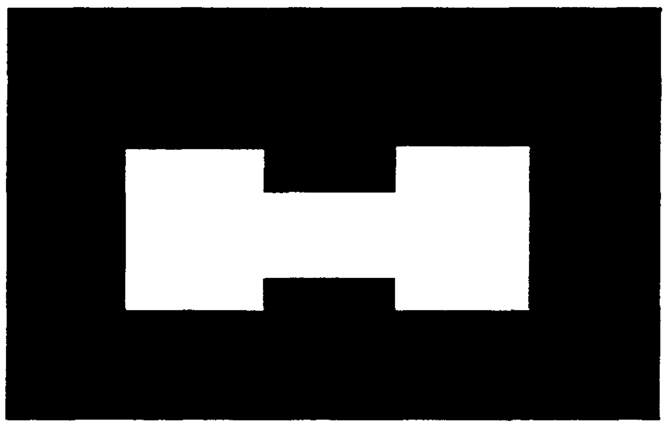


Train-thru-the-lens Mask Window (continued)

Example – Creating a *train-thru-the-lens mask* window

(continued)

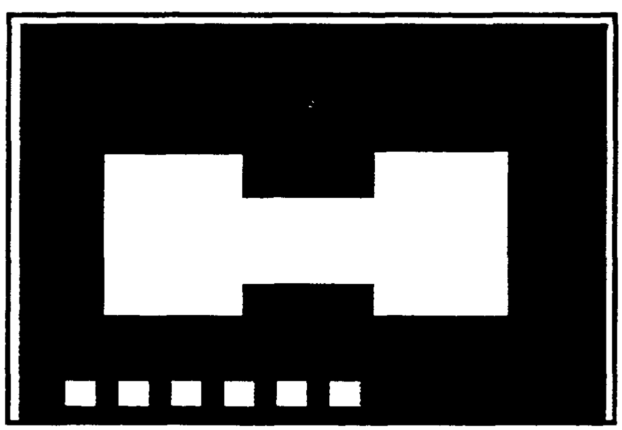
Your Action	*Comment*
Make a template *to match the desired window size, shape, and position.*	Design a template such that, when placed in the camera field of view, the area(s) to be inspected appear white in the selected threshold image, and the areas to be ignored appear black. In our case, the template might look like this:



White area of template represents the desired window size and shape.

Place the template in the camera field of view so that the threshold image indicates correct size and position.	With the template in the threshold image, adjust the distance between the camera and the template, and adjust the horizontal and vertical orientation of the template, so that the white areas of the template match the required inspection area of the window:

Template shown in threshold image matching required window size, shape, and position.



Pick the Window Shape *icon to select the* train-thru-the-lens mask *option.*	When you select the *train-thru-the-lens mask* icon, the VIM2 module takes a "snapshot" of the template; the white area shown in the threshold image is now the window.

Note: If the window does not come out right the first time, simply make any adjustments necessary, then cycle through the *Window Shape* icons and reselect the *train-thru-the-lens mask* icon.

Train-thru-the-lens Mask Window (continued)

The *train-thru-the-lens mask* window cannot be resized or moved using the *Window Move* or *Window Size* menu icons. The only way to alter the *train-thru-the-lens mask* window is to redo the set-up procedure previously described, after making the necessary changes to the template or to the positioning of the template.

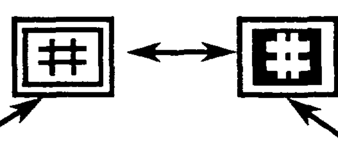
Selecting the Window Color

The windows can perform one function; they count pixels of a specified color – either black or white. To specify the color, after selecting and placing the window, pick the *Color* icon on the second part of the *Window* menu:

Color Icon – Pick this icon to select the pixel color – either black or white – for the window function.

Select the window *color* based on the color of the feature of interest you are inspecting – if you are inspecting a *black* feature on a white background, select *black;* if you are inspecting a *white* feature on a black background, select *white.*

Your Action	***Comments***
Pick the Color *icon to select the window color.*	The icon toggles between two options – *white* (white box with a black "#") or *black* (black box with a white "#").



White – With *white* selected, the window counts the number of white pixels it covers.

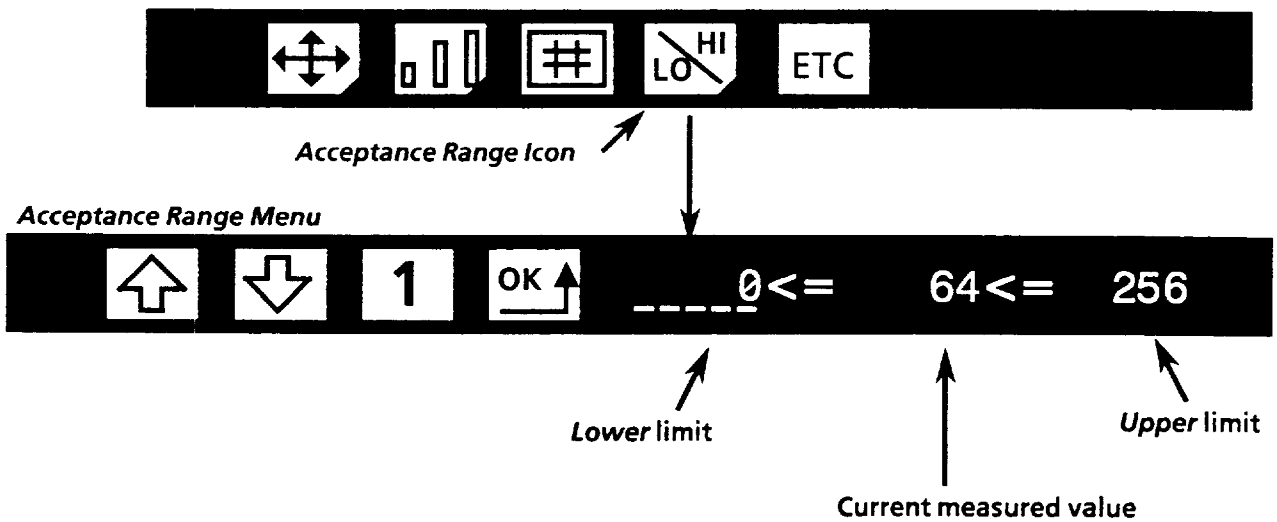
Black – With *black* selected, the window counts the number of black pixels it covers.

Setting the Acceptance Range

Once you have placed a window and selected the *shape* and *color*, next set the *acceptance range* of the window.

Note: For an overview of the acceptance range concepts and setup, see page 9-25, which discusses setting the acceptance range for *line gages*. This information is also generally applicable in setting the acceptance range for windows.

Your Action	*Comments*
Pick the Acceptance Range *icon on the* Window *menu.*	When you pick the *Acceptance Range* ("HI/LO") icon on the *Window* menu, the *Acceptance Range* menu appears:

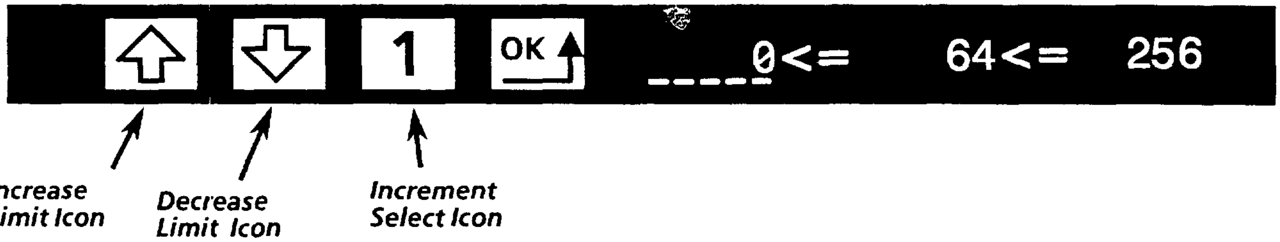


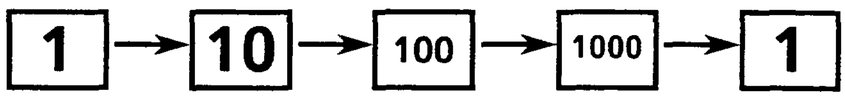
Observe the window result.	On the right side of the *Acceptance Range* menu are three numbers, separated by "< =" signs. The *middle* number is the current measured result of the window. For example, if you select the color *black*, then pick the *Acceptance Range* icon, the number of black pixels covered by the window is displayed.
*Select the limit (*upper or lower*) you want to change.*	The *lower limit* is the *left* number of the three numbers displayed on the right side of the menu, the *upper limit* is the number on the *right* (**Note:** Default values are displayed for the limits initially). To select either limit, pick that limit with the light pen. The selected limit is *underlined.*

Selected limit is underlined

Setting the Acceptance Range
(continued)

Acceptance Range Menu



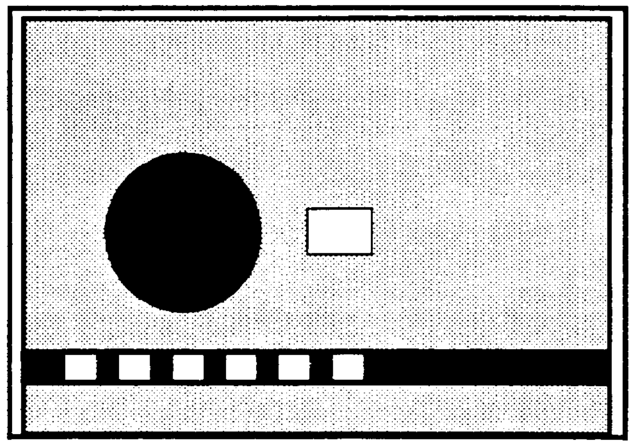
Your Action	Comments
Pick the Increment.Select *icon to select the desired increment.*	Picking the *Increment Select* icon causes the icon to cycle through the increment choices: 1, 10, 100, and 1000. 
Increase or decrease limit.	To *increase* the selected limit by the amount shown on the *Increment Select* icon, pick the *Increase Limit* icon. You can increase the limit to the maximum allowed by the window (this depends on the size ofthe window). To *decrease* the selected limit by the amount shown on the *Increment Select* icon, pick the *Decrease Limit* icon. You can decrease the limit to a value of 0. **Note:** The VIM2 does not allow you to set the upper limit to a value lower than the lower limit, or to set the lower limit to a value higher than the upper limit.
When finished, pick the OK *icon to display the* Window *menu.*	Picking the *OK* icon at this point returns the *Window* menu.

Setting the Acceptance Range (continued)

Example: Setting Acceptance Range

Let's suppose that the target pattern represents a part. Suppose you want to inspect the size of the dark circle on the part (shown as a lone black circle in threshold image 1, as illustrated below).

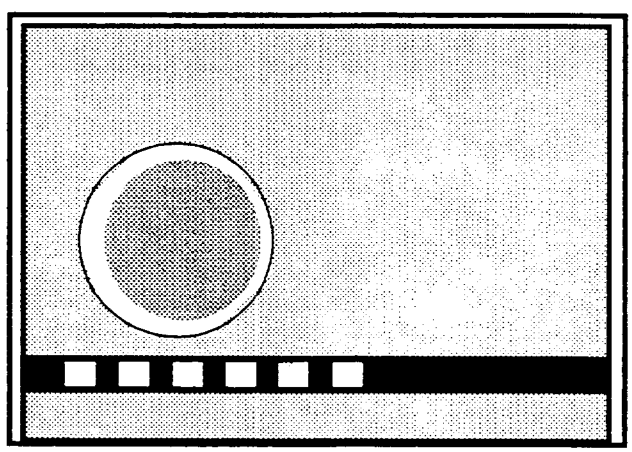
Window 1 displayed on threshold image 1 (example image)



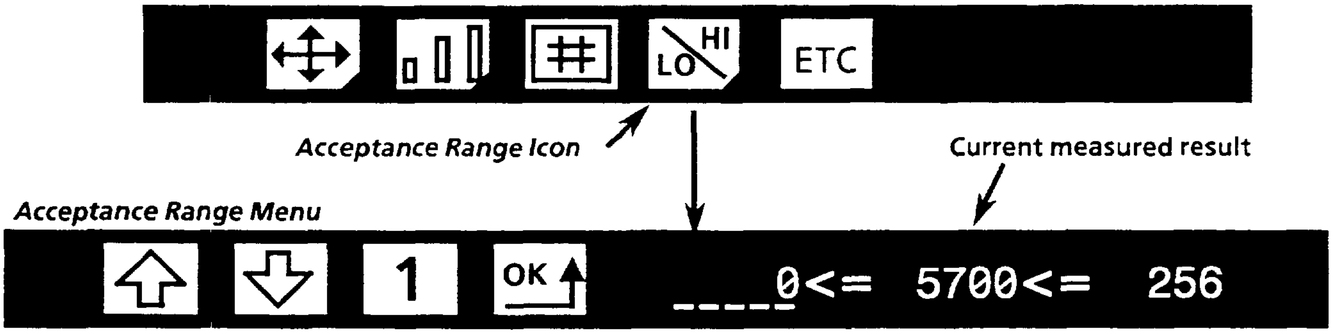
Your Action	*Comments*
Set up window 1 using the Window *menu icons.*	To measure the size of the black circle, set up window 1 (using the *Window* menu icons) as follows:

- Select *window 1* (and thus, threshold image 1).
- Select *circular* as the shape.
- *Place* the window to overlap the black circle.
- Set *Color* icon to black.

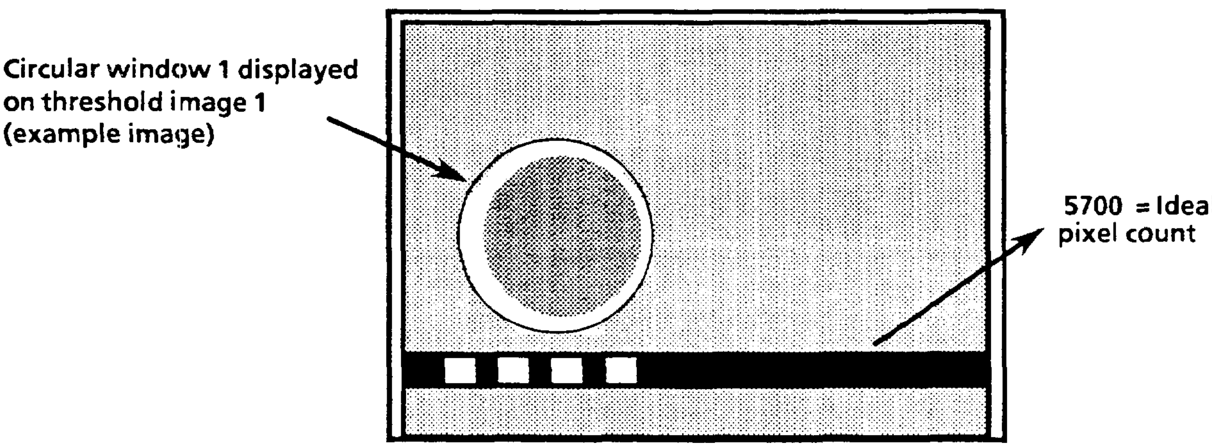
Circular window 1 displayed on threshold image 1 (example image)



Setting the Acceptance Range (continued)	**Example: Setting Acceptance Range** *(continued)*
Your Action	***Comments***
Access the Acceptance Range *menu.*	With window 1 in place and configured as described on the previous page, pick the *Acceptance Range* icon.



Observe the measured result and determine the limits.	Suppose the measured result is 5700 – that is, the area of the black circle, as measured by the window, is 5700 pixels.



	Suppose the value, 5700, is for the ideal part, and you can accept a part within ±5%, or 5700 (pixels), ±285. You then set the *upper* limit to 5985, and the *lower* limit to 5415.
Select and set the upper *limit to* 5985.	Pick the *upper* limit. Then pick the *Increase Limit* or *Decrease Limit* icons as necessary to change the limit.
Select and set the lower *limit to* 5415.	Pick the *lower* limit. Then pick the *Increase Limit* or *Decrease Limit* icons as necessary to change the limit.

Setting the Acceptance Range (continued)

Using the Learn Mode and other Run Mode Functions

The VIM2 module features a variation of its *Run* mode called the *learn* mode. In *learn mode,* the VIM2 accumulates *statistics for each enabled inspection tool,* based on a number of trial inspections. Use of the *learn* mode can be helpful in determining the optimal *acceptance range* for inspection tools and probe.

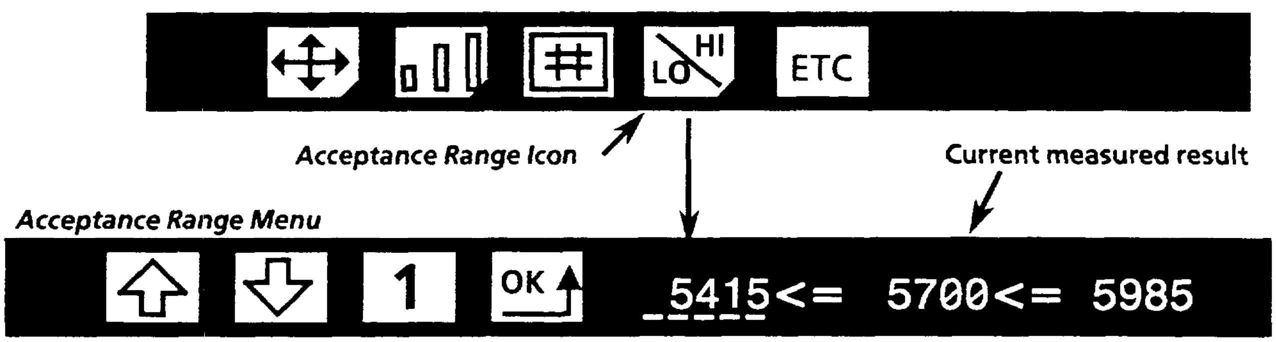
The VIM2 module also features a number of run mode *arm* options, which can be helpful in refining your configuration.

See Chapter 13 for more information regarding use of the *learn* mode and other *run* mode options.

Accessing Statistics

When you run the VIM2 module in *learn mode,* you accumulate statistics for all enabled visions tools and probes. To observe the collected statistics for a particular window, first access the *Acceptance Range* menu on the *Window* menu:

Your Action	*Comments*
Pick the Acceptance Range *icon to access the* Acceptance Range *menu.*	Picking the *Acceptance Range* icon on the *Window* menu displays the *Acceptance Range* menu.



Pick the current measured result.	On the right side of the *Acceptance Range* menu are three numbers, separated by "< =" signs. The *middle* number is the *current measured result* of the window. When you pick this number, the statistics are displayed (see diagram on the next page) and the window disappears from the screen.
Pick the current measured result again to remove the statistics display.	Picking the *current measured result* again redisplays the window and removes the statistics display.

Accessing Statistics
(continued)

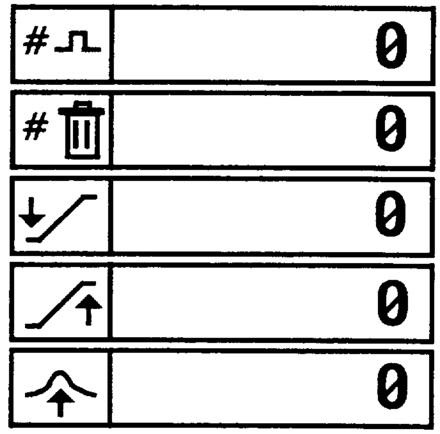
Total of triggers processed

Number of failures for window

Lowest result collected

Highest result collected

Average of all results collected



Note: The results of *failed* inspections for the window are not included in the *lowest*, *highest*, and *average* values. These statistics are based on passing inspections only. Therefore, for preliminary trial runs, you may want to set the acceptance range to extreme limits initially, for the purpose of collecting statistics for all inspections.

Chapter 11 Using the X / Y Float Gages

Chapter Objectives

The objective of this chapter is to explain the concepts of *shift compensation* and *shift registration*, and show how to set up *X/Y float gages* in order to use the VIM2 module's shift compensation and shift registration capabilities.

Note: Before reading this chapter, be sure you have read and understand at least Chapters 6, 9, and 10.

Some Questions and Answers

This section introduces the topic of *X/Y float gages* by posing some questions about float gages, and answering them.

What is a float gage?
A float gage is a special case of line gage (see Chapter 9 for more about line gages). Like other line gages, float gages detect position of edges or centers of "blobs" (see Chapter 3 for "blob" definition), or widths of blobs. The *float gages* (gages "X" and "Y"), however, have a special purpose – they are specifically used for setting up the VIM2 module's *X/Y shift compensation* capability.

What is "X/Y shift compensation?"
X/Y shift compensation is a function of the VIM2 module which adjusts the line gage and window positions to compensate for horizontal (X) and/or vertical (Y) shifts in workpiece position from inspection to inspection. Without this shift compensation, if the workpiece is not in precisely the same position from inspection to inspection, the measurement of line gages and windows is affected; this can cause mistaken *reject* or *accept* decisions.

How does X/Y shift compensation work?
During setup, line gages and/or windows are placed in specific positions based on the nominal (setup) workpiece position in the threshold image(s). During setup, float gages can be placed as well, also based on the nominal workpiece position. Then if, during an inspection, *the workpiece position varies from the setup position*, the float gages measure the amount and direction of variance. *X/Y shift compensation* is then applied – line gages and/or windows are shifted in exactly the same amount and direction as the workpiece. In some cases of line gage upper functions (i.e., any functions which locate a blob edge or blob center), the acceptance range limits are adjusted also. The result is that the line gage and window *accept / reject* decisions are not affected by the shift in workpiece position. In the cases of windows (which count pixels), and line gage functions which measure width, or count pixels, blobs, or edges, the measurement results are relatively unaffected as well.

Some Questions and Answers *(continued)*

Which vision tools are affected by shift compensation?
Any line gage or window you configure can be set to use shift compensation; you *enable* or *disable* shift compensation for each line gage or window using the *Anchor / Float* icon (see Chapters 9 and 10 for line gage and window information). You must, of course, also set up at least one float gage in order to use shift compensation.

What kinds of float gages are there?
There are 2 float gages – "X" and "Y." Use float gage X for *horizontal* shift compensation; use Y for *vertical* shift compensation. You can use *either* or *both* float gages in your configuration, depending on your application requirements.

When do you need to use float gages?
Use float gages whenever the position of the workpiece may vary from inspection to inspection.

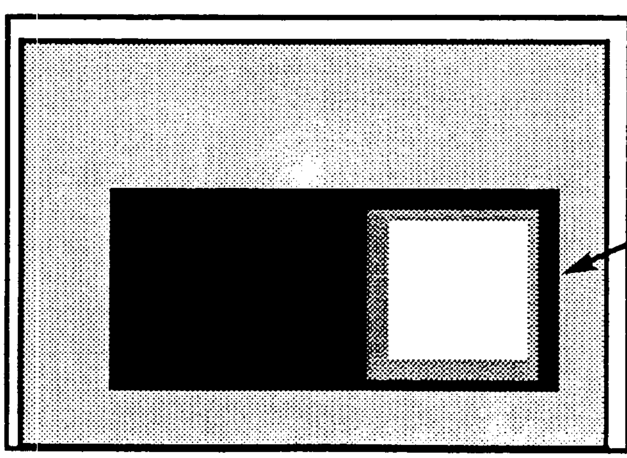
When do you set up a float gage?
As with any line gage, set up a float gage after you have:

- Set up the workstage, and have a workpiece in the nominal or ideal position (see Chapter 4), so that the workpiece features to be inspected are shown clearly in the field of view (and on the monitor).
- Set the threshold image(s) to enhance the particular feature(s) you are going to inspect (see Chapter 8).

Shift Compensation

This section discusses the concept of shift compensation, beginning with *horizontal* shift compensation (this section assumes the VIM2 module and target pattern are set up as described in Chapter 6).

To illustrate the *horizontal* shift compensation, suppose that the black rectangle in threshold image 2 is a workpiece. Suppose that you place window 2 so it covers the white square (see Chapter 10 for window information).

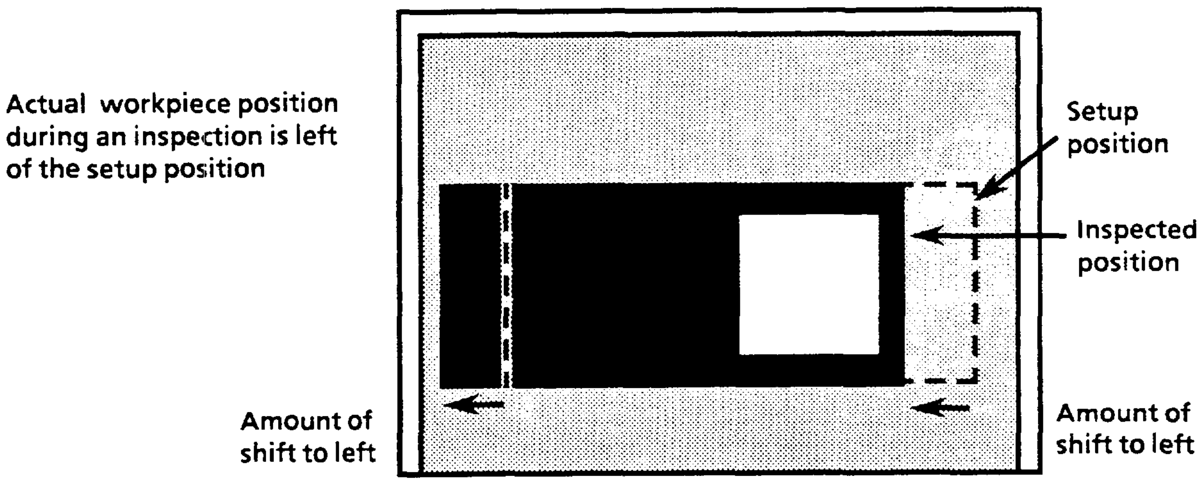


Window 2 placed to cover the white square on threshold image 2 (example image)

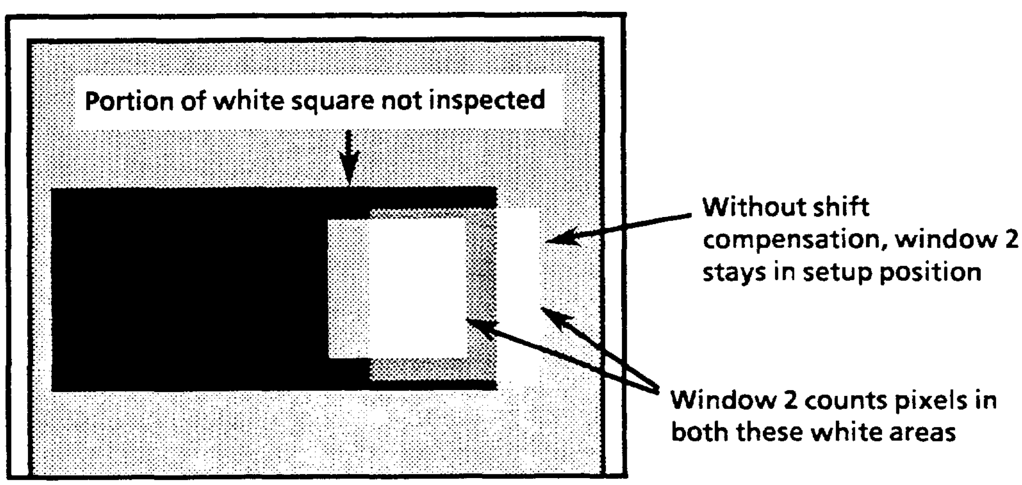
Shift Compensation
(continued)

Then you set up window 2 to count *white* pixels, in order to measure the size of the white square.

Now suppose that during your inspection process, the workpiece position varies *horizontally* from inspection to inspection – sometimes the workpiece is positioned a little right of the initial setup position, or a little left of it. Let's look at one case where, during an inspection, the workpiece is detected *left* of the *setup* position, as shown below:



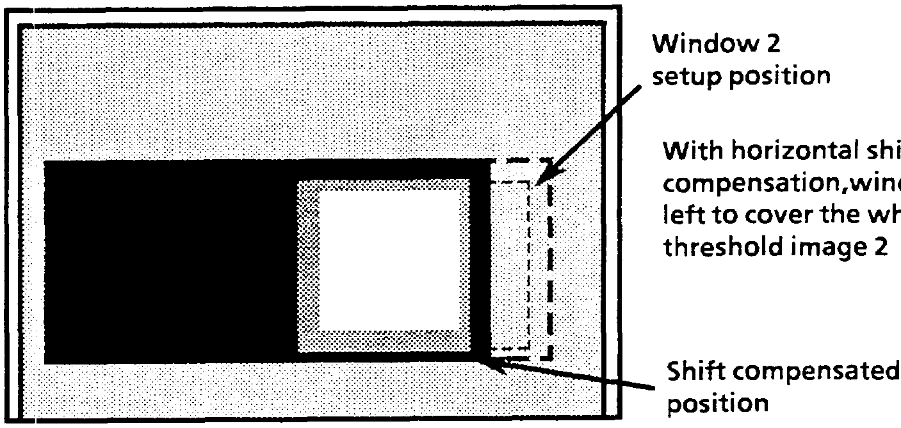
Without shift compensation (see diagram below), window 2 inspects the same area over which it was placed during setup, no matter what the actual position of the workpiece is.



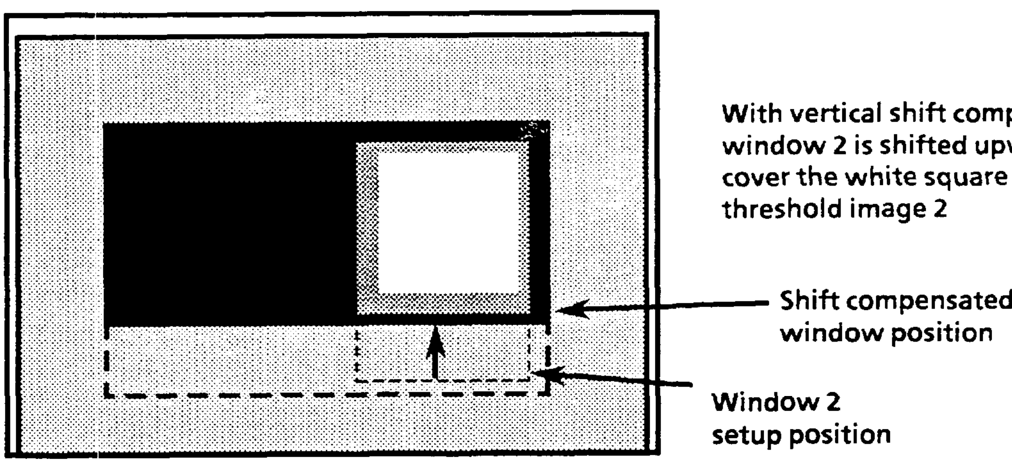
As shown above, the workpiece position during an inspection is *not the same* as the setup position. During the inspection, window 2 covers only part of the white square, and covers a white area outside the black rectangle; this results in an inaccurate measurement of the white square. Now suppose that you *do* use shift compensation. To do this, you set up horizontal float gage X, using it to measure the *horizontal* position shift.

Shift Compensation
(continued)

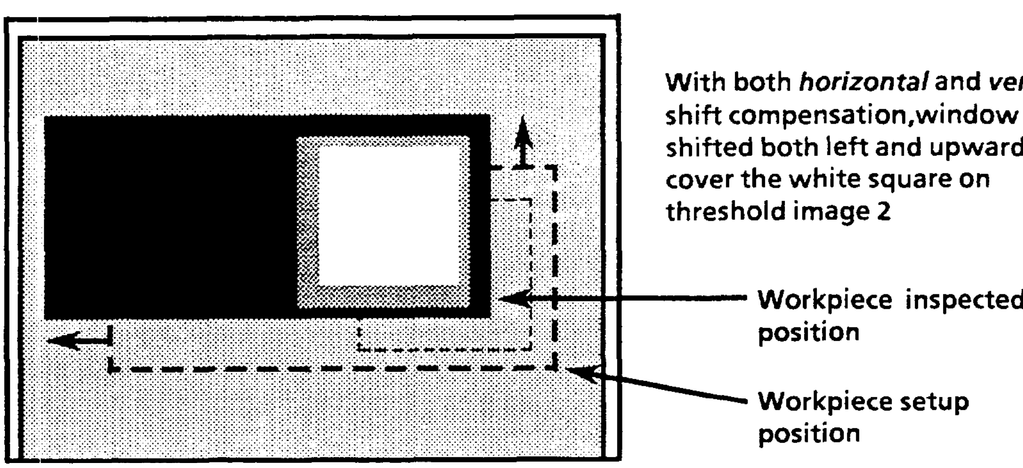
When using horizontal float gage X to measure horizontal position shift, if the workpiece position shifts relative to the setup position, the window is shifted also – in the same amount and direction as the workpiece:



The above example illustrates a case requiring *horizontal* shift compensation. *Vertical* shift compensation can also be applied, using float gage Y, for *vertical* position shifts:



Horizontal and *vertical* compensation can both be applied to compensate for a combination of *horizontal* and *vertical* shift:

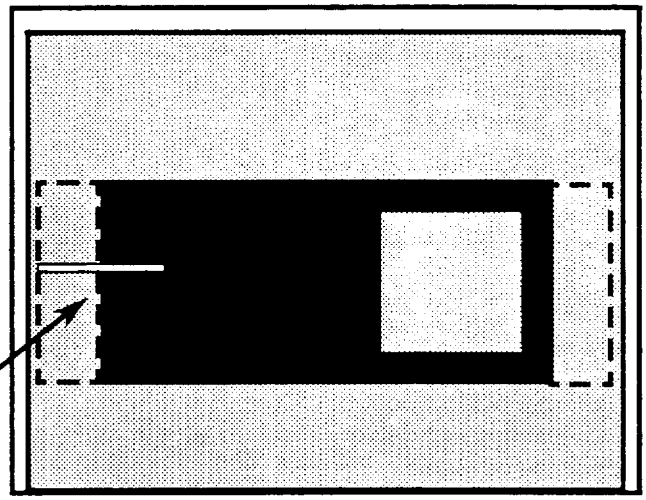


How to Use the Float Gages

Once you have determined that your application requires *horizontal* and/or *vertical* shift compensation, you will need to configure the appropriate float gage(s):

- Configure float gage X (which is always horizontal) to measure the *horizontal* position shift, along the gage's length, of a feature in the assigned threshold image.

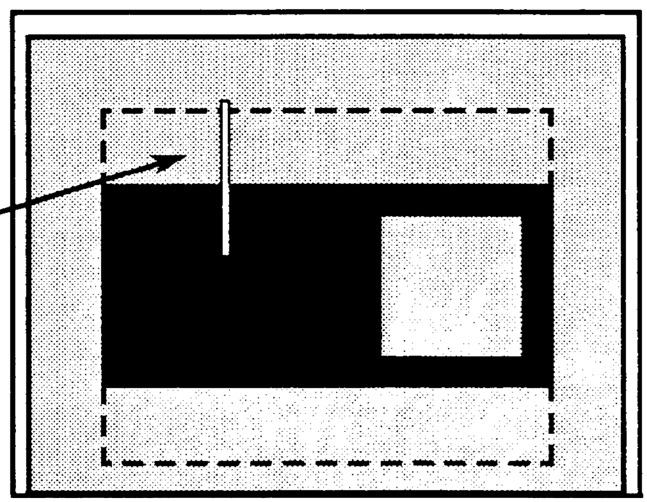
Float gage X displayed on threshold image 2 (example image)



Float gage X placed over left edge of black rectangle to gage amount of shift

- Configure float gage Y (which is always vertical) to measure the *vertical* position shift, along the gage's length, of a feature in the assigned threshold image.

Float gage Y displayed on threshold image 2 (example image)

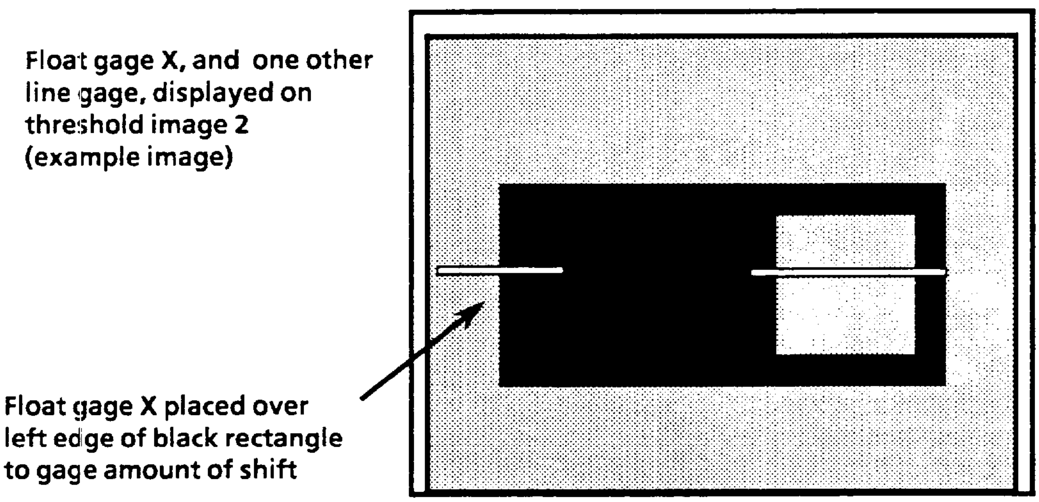


Float gage Y placed over top edge of black rectangle to gage amount of shift

- Use *both* float gages X and Y, if necessary, to measure both the *horizontal* shift and the *vertical* shift.

How a Float Gage Works

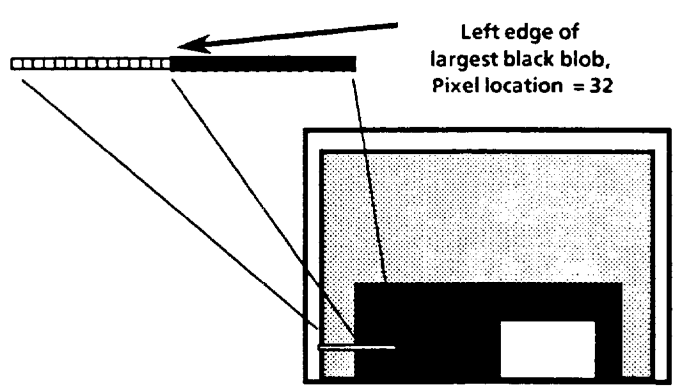
Suppose that the black rectangle in threshold image 2 (shown below) represents a workpiece. Suppose that during setup you have placed a horizontal line gage over the white square. You have configured the line gage with shift compensation *enabled.* And you have placed float gage X over the left edge of black rectangle in the threshold image, as shown.



While setting up float gage X in order to locate the edge of the black rectangle, you have:

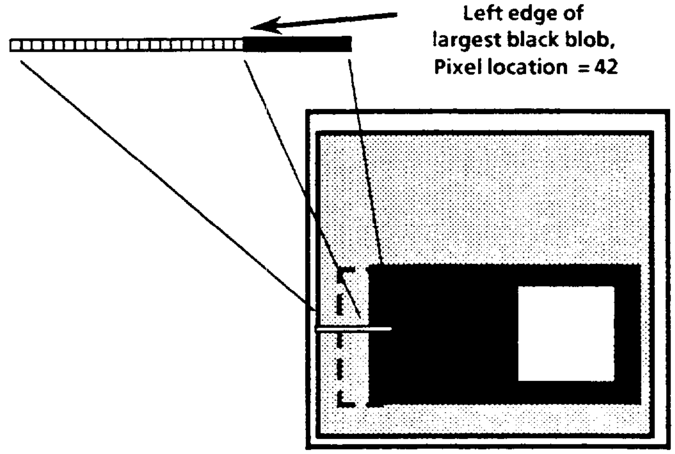
- Selected the *first Gage Function* icon, the upper function of which locates the *left edge of the largest blob.*
- Set the *Color* to *black*, to locate the left edge of the largest *black* blob.

During setup of float gage X, you pick the *Acceptance Range* ("Hi/Lo") icon to read the current measurement of the *upper* function; this turns out to be "32," which is the pixel location of the left edge of the black rectangle. When you exit the *Acceptance Range* menu, the "nominal" result for the float gage X upper function is set to 32.

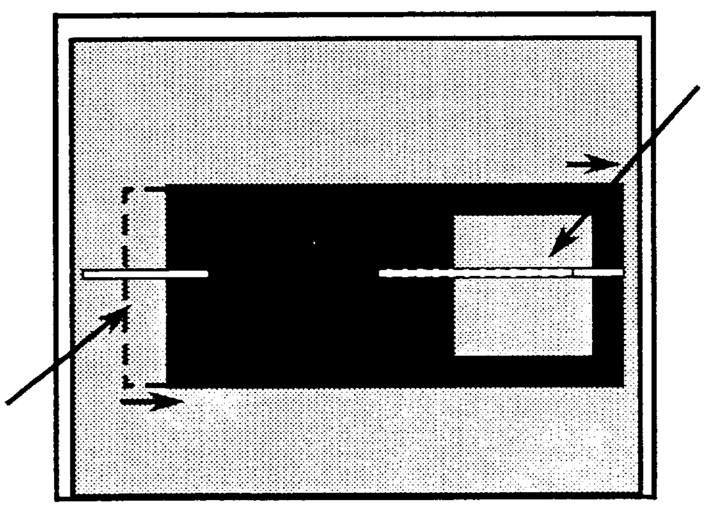


How a Float Gage Works (continued)

Suppose that later, with the VIM2 performing a particular inspection, the workpiece position is shifted – the location of the left edge is "42." That means that the workpiece has shifted 10 pixels to the right.



Any line gages or windows which you have configured with shift compensation *enabled*, will be shifted 10 pixels to the right also, for this particular inspection. The line gages and/or windows then proceed to measure as usual.

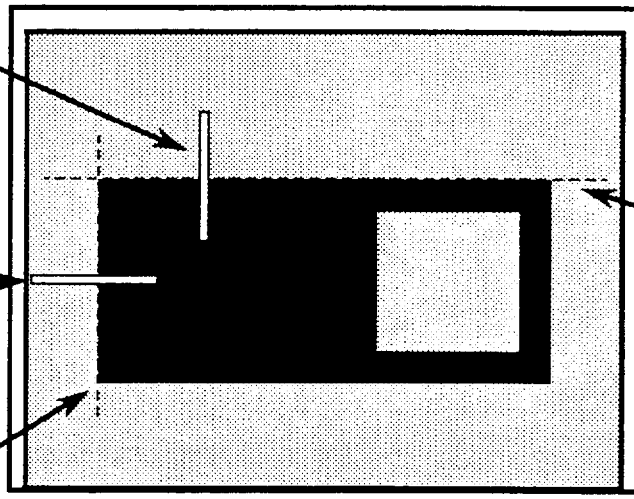


Acceptance Range Shift: For some line gage functions, the *acceptance range is adjusted* according to the amount and direction of shift detected. This is true for any line gage function which detects the *location* of a blob edge or center (i.e., the upper function for the first seven line gage function icons shown on page 9-16; see pages 9-15 to 9-24 for more line gage function information). For example, let's say the line gage shown above is used to detect the center of the white square, with the acceptance range lower limit set to 160, and the upper limit to 180. With a detected shift to the right of 10 pixels, the lower limit is changed, for this one inspection, from 160 to 170, and the upper limit from 180 to 190.

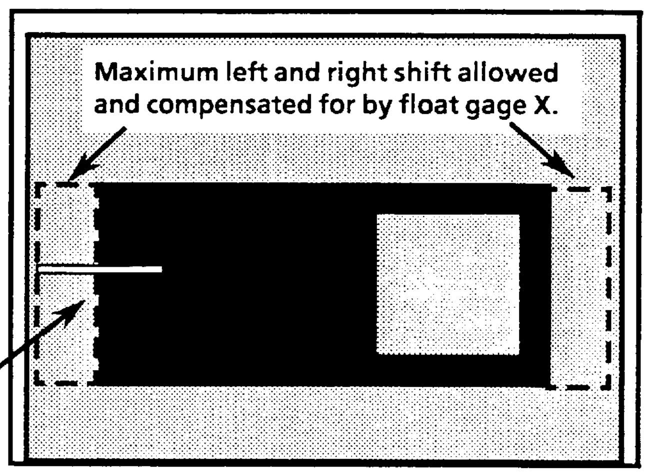
How to Set Up the Float Gages

In order to set up float gages effectively:

- *Select a threshold image* for each float gage which has at least one straight edge to locate (for float gage X, this would be a *vertical* edge, for float gage Y, *horizontal*).



- *Position the float gage*, and *select the gage function*, to locate the feature of your choice. The chosen feature must be in a position *far enough away* from the screen's boundaries to allow the gage to overlap over each side of the feature; in this way the gage can accomodate the workpiece shifting in two directions.



Note: In *setting the float gage function*, only the first seven of the function icons (see listing on page 9-16) are available for the float gage. Also, the shift compensation is based on the float gage's *upper function* only.

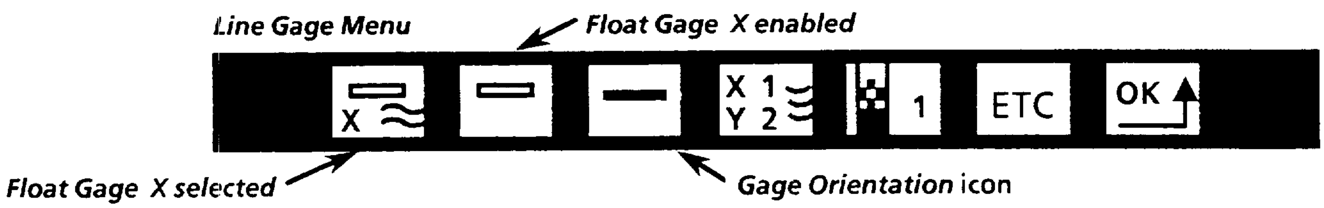
- *Set the acceptance range* for the float gage to limit the amount of workpiece shift so that no line gage or window is shifted partially or entirely off of the screen (this would cause the inspection to fail).

Float Gage X – Horizontal Shift Compensation

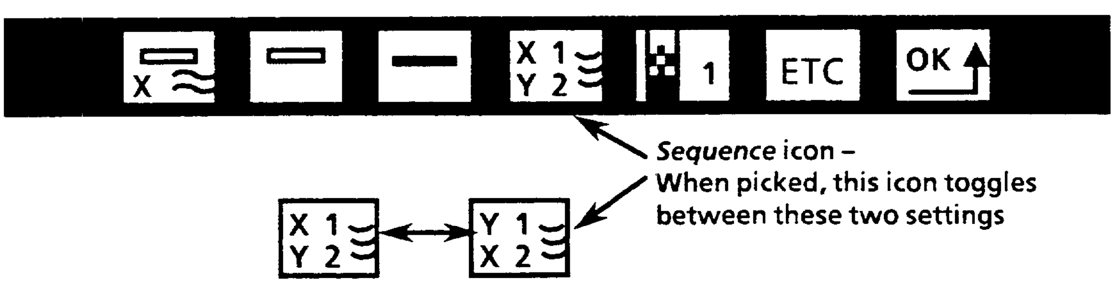
Set up float gage X using the same general procedure you might use to set up any other line gage – first, access the *Line Gage* menu.

Listed below is an example procedure which lists the basic steps for setting up the float gage (**Note:** The steps listed below assume the VIM2 module is set up as described in the "Quick Start" section of Chapter 6):

Your Action	Comments
Select and enable float gage X.	Pick the *Gage Function* and *Gage Enable* icons to select and enable float gage X.



Your Action	Comments
Ignore the Gage Orientation *icon.*	The float gage orientation cannot be changed. Float gage X is always set to *horizontal*; float gage Y is always *vertical*. With a float gage selected, when you pick the *Gage Orientation* icon, it flashes, but does not change.
Pick the Sequence *icon to assign float gage sequence.*	When either the X or Y float gage are selected, the fourth icon on the first part of the *Line Gage* menu is the *Sequence* icon. When picked, this icon toggles between two selections – "X1/Y2" and "Y1/X2." For example purposes, pick "X1/Y2."

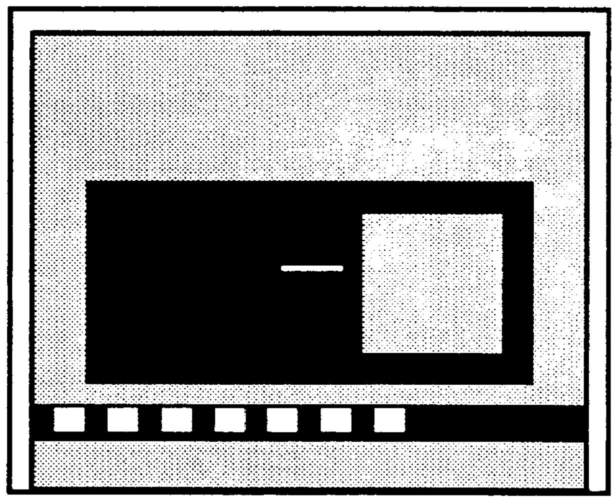


The *Sequence* icon is used to assign float gage sequence when *both* X and Y float gages are used. Float gage sequence is discussed further under "Assigning Float Gage Sequence" later in this chapter.

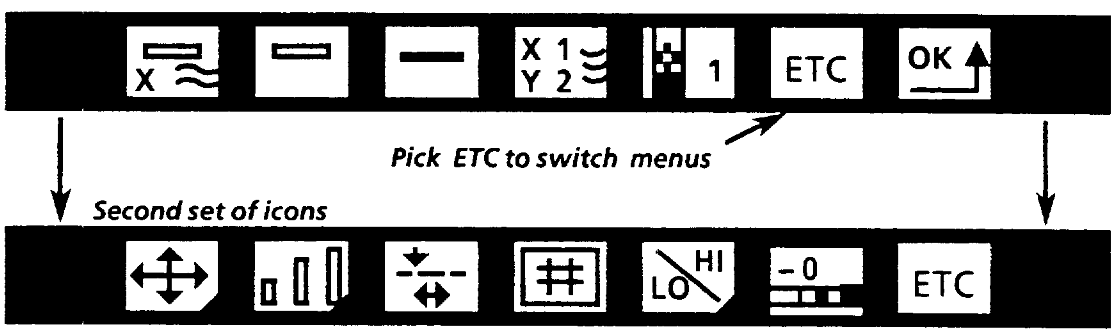
Float Gage X – Horizontal Shift Compensation (continued)

Your Action	*Comments*
Pick the Threshold Image *icon to select threshold image 2.*	As with any line gage, you select which of the four threshold images you will inspect with the selected gage. With threshold image 2 selected, float gage X is displayed on the screen over threshold image 2:

Float gage X displayed on threshold image 2 (example image)

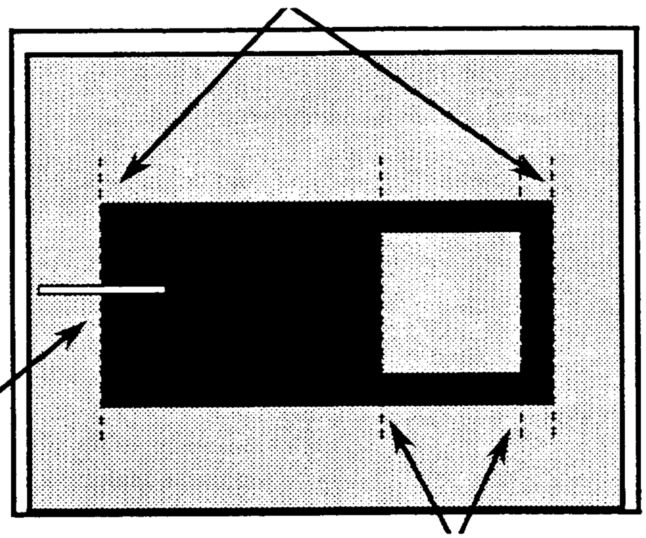


Pick the ETC *icon to access the second set of* Line Gage *menu icons.*	When you pick the *ETC* icon, the second part of the *Line Gage* menu is displayed (Note: Pick *ETC* on the second menu to redisplay the first).

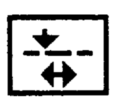


Float Gage X – Horizontal Shift Compensation *(continued)*

Your Action	*Comments*
Choose a straight, vertical edge to locate (on the selected threshold image), in order to gage the workpiece position.	When using the horizontal float gage X, look for a straight, vertical edge to locate. For example, in threshold image 2, there are several straight, vertical edges. In our example, we will *locate the left edge of the black rectangle.*



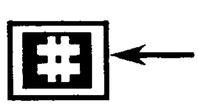
Using the Gage Move *and* Gage Size *menus, place the float gage over the left edge of the black rectangle.*	To measure the left edge of the black rectangle, place the float gage so it crosses this edge, allowing the gage to overlap on each side of the edge, as shown above.
Pick the Gage Function *icon and select the first icon.*	In order to measure the position of the left edge, select the first icon; recall that the upper function measures the position of the "left edge of the largest blob."



Upper Function: Locate left edge of largest blob.

Note: You could also select other functions, such as the fourth icon, which locates the "left edge of the leftmost blob."

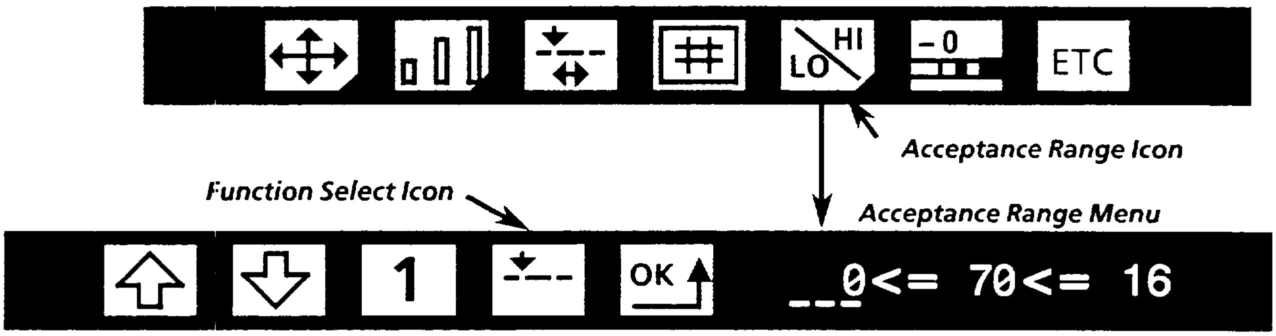
Pick the Color *icon to select black.*



Float Gage X – Horizontal Shift Compensation *(continued)*

With the float gage in position, and with the *Gage Function* and *Color* selected, now set the acceptance range for the float gage.

Your Action	Comments
Pick the Acceptance Range *icon.*	Picking the *Acceptance Range* icon displays the *Acceptance Range* menu.



Pick the Function Select *icon to select the* upper *function.*

When you select the *upper* function, the current measured result for the upper function is displayed.

Set the upper and lower limits for the upper *function.*

Setting the limits may require several attempts at trial and observation before an optimum range is achieved. Or, you might perform a trial run in learn mode, and then display the float gage statistics in order to help establish acceptance range limits (see "Accessing Statistics" in this chapter).

Remember to set both the upper and lower limits in consideration of these limitations:

- The acceptance range should not be so broad as to allow any line gage or window to be shifted off of the image (this constitutes an "acquisition error").
- Do not set the upper or lower limits of a float gage's upper function to their respective extreme value – this would not allow the float gage ever to fail; the shift compensation would take place, but only as far as the float gage could detect.

Note: If the amount of shifting that actually occurs shifts any line gages or windows off the screen, you may have to reposition the line gages or windows, or rearrange the workstage, to better accomodate the position shift.

Pick the Function Select *icon to select the* lower *function.*

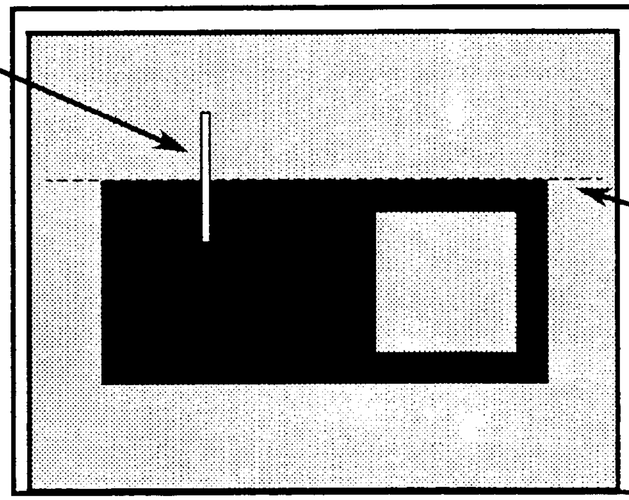
When you select the *lower* function, the current measured result for the lower function is displayed.

Set the upper and lower limits for the lower function.

Set the upper and lower limits to extreme values so that the float gage cannot fail due to a lower function measurement.

Float Gage Y – Vertical Shift Compensation

Set up float gage Y using generally the same procedure described for setting up float gage X. The difference is, of course, you place the *vertical* float gage Y to locate the position of a horizontal edge, so that it measures the amount of shift of that edge per inspection.



Note: Remember that the float gage acceptance range should not be so broad as to allow any line gage or window to be shifted off of the image (this constitutes an "acquisition error"). Or, if "mode 1" (same frame compensation) of brightness compensation is selected, do not allow any line gages or windows to shift into the top 48 rows of the image.

Using the Learn Mode and other Run Mode Functions
The VIM2 module features a variation of its *run* mode called the *learn* mode. In *learn mode,* the VIM2 module accumulates *statistics for each enabled inspection tool,* based on a number of trial inspections. Use of the *learn* mode can be helpful in determining the optimal *acceptance range* for the float gages and other vision tools.

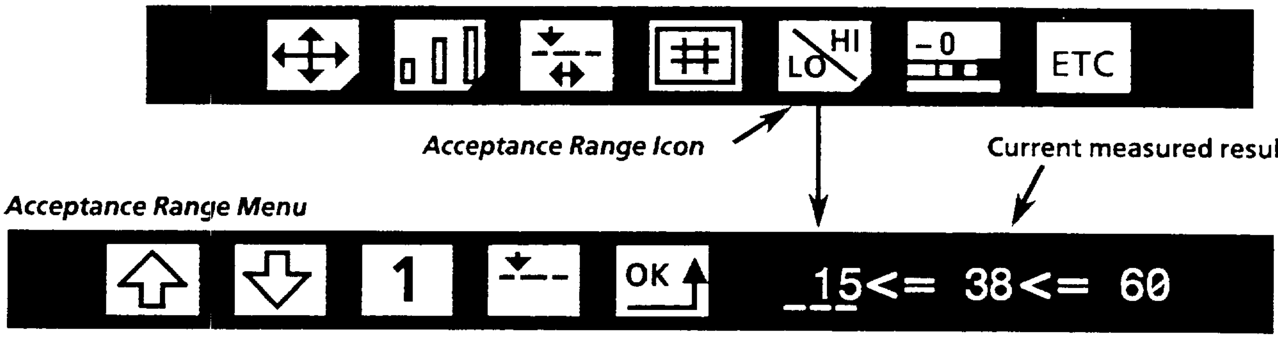
The VIM2 module also features a number of run mode *arm* options, which can be helpful in refining your configuration. See Chapter 13 for more information regarding use of the *learn* mode and other *run* mode options.

Accessing Statistics

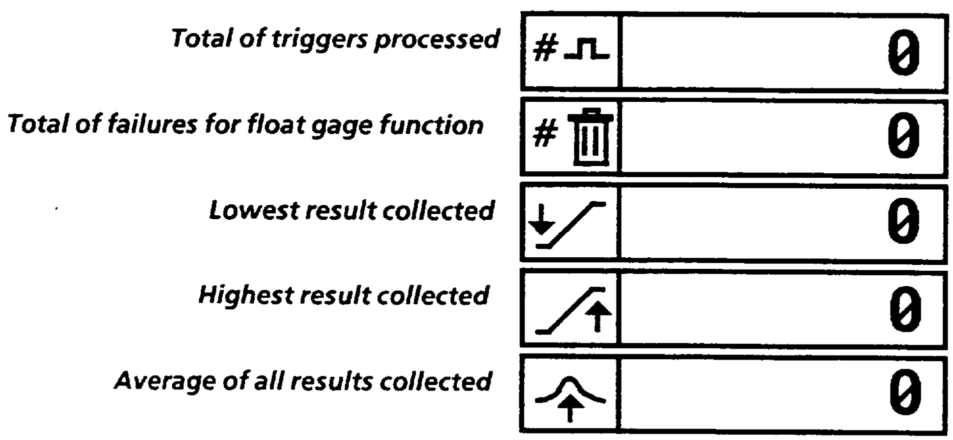
When you run the VIM2 module in *learn mode,* you accumulate statistics for all enabled visions tools and probes. To observe the collected statistics for a particular float gage, first access the *Acceptance Range* menu for the float gage, as described on the next page.

Accessing Statistics
(continued)

Your Action	*Comments*
Pick the Acceptance Range *icon on the* Line Gage *menu.*	Picking the *Acceptance Range* icon displays the *Acceptance Range* menu.



Pick the current measured result.	On the right side of the *Acceptance Range* menu are three numbers, separated by "< =" signs. The *middle* number is the *current result* of the float gage. When you pick this number, the statistics are displayed (see diagram below), and the float gage disappears.

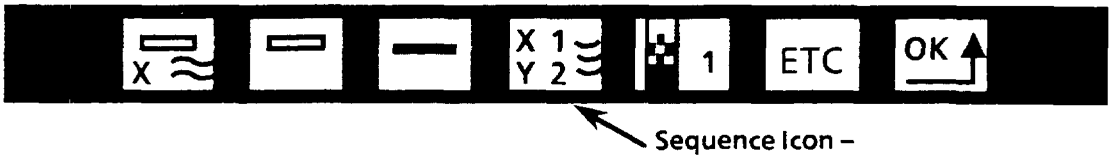


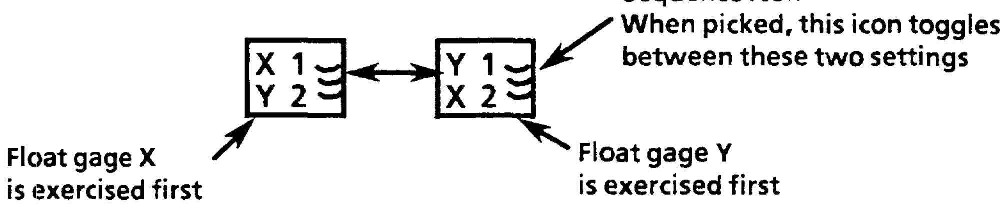
Pick the current measured result again to remove the statistics display.	Picking the *current measured result* again removes the statistics display and redisplays the line gage.
	Note: The results of *failed* inspections are not included in the *lowest*, *highest*, and *average* values. These statistics are based on passing inspections only. Therefore, for preliminary trial runs, you may want to set the acceptance range to extreme limits initially, for the purpose of collecting statistics for all inspections.

Assigning Float Gage Sequence

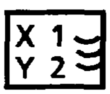
Whenever you configure a float gage, you must set the float gage sequence. You do this by picking the *Sequence* icon. When you use only one float gage, either X or Y, the setting of the *Sequence* icon does not matter. However, when you use both the X and Y float gages, the setting of the *Sequence* icon *does* matter.

When picked, the *Sequence* icon toggles between two selections (*X1/Y2*, or *Y1/X2*), as shown below:



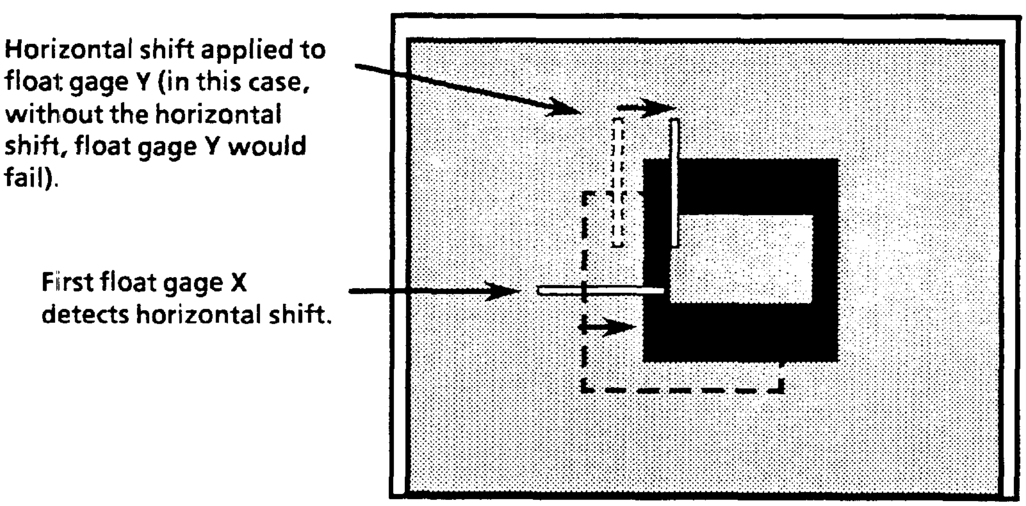


The *Sequence* icon setting determines which float gage is exercised first. The amount of shift determined by the first float gage is applied to shift the position of the second float gage. The two *Sequence* selections are illustrated below:

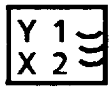


X1/Y2 – In general, select *X1/Y2* if you expect more horizontal shift than vertical, or if the width of the feature measured by float gage Y is more limited than the height of the feature measured by float gage X.

With X1/Y2 selected: Float gage X results are read first by the VIM2 module, then the horizontal shift is applied to float gage Y. At this point, both X and Y results are used to reposition the line gages and/or windows:



Assigning Float Gage Sequence
(continued)

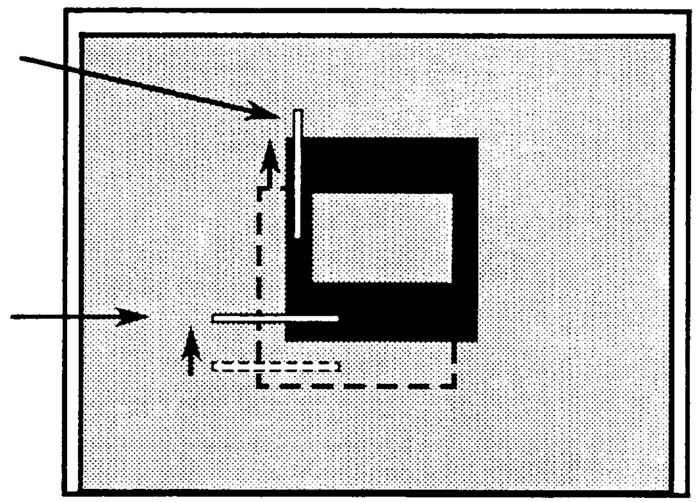


Y1/X2 – In general, select *Y1/X2* if you expect more vertical shift than horizontal, or if the height of the feature measured by float gage X is more limited than the width of the feature measured by float gage Y.

With Y1/X2 selected: Float gage Y results are read first by the VIM2 module, then vertical shift is applied to float gage X. At this point, both X and Y results are used to reposition the line gages and/or windows:

First float gage Y detects vertical shift.

Then the vertical shift is applied to float gage X (in this case, without the vertical shift, float gage X would fail).



Observing Shift Compensation in Action

Once you have set up your configuration using shift compensation, if you place the VIM2 module in *run* mode (see Chapter 13), you can observe shift compensation in action. For example, if you are using a particular window in your configuration (and enable shift compensation for that window), you can select the display mode which shows that window. Then if, for an inspection, the workpiece location shifts, the position of the window shifts accordingly; the shift in workpiece location and in window position will be observable on the monitor (**Note:** The shift compensation will be applied to the *displayed* window, whether or not the shift compensation has been enabled for the window).

Shift Registration

The VIM2 module has a capability called *shift registration* which allows you to permanently adjust line gages and windows according to a new nominal part position during setup. Use shift registration during setup when you change the nominal workpiece position for your inspection, and you want to readjust the line gage and window positions accordingly.

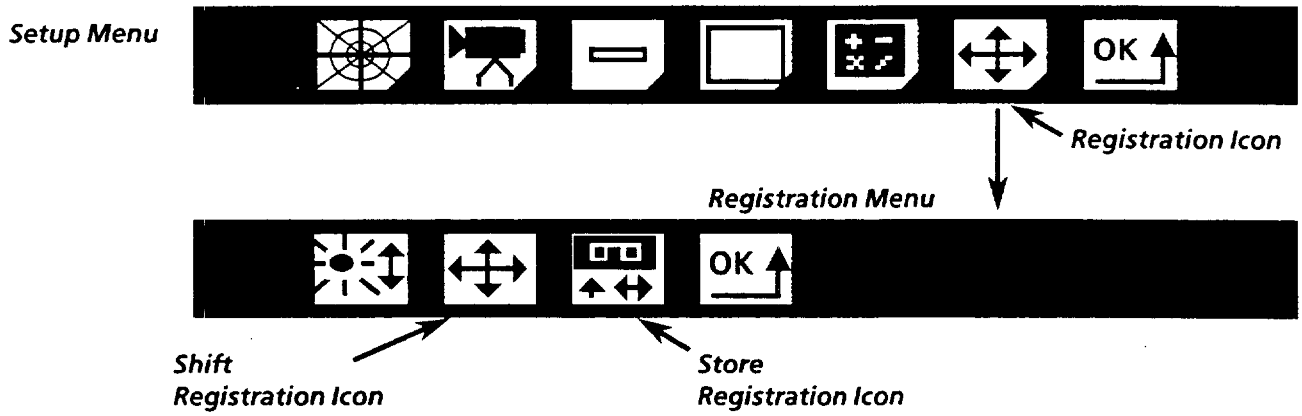
Shift *registration* works just like shift *compensation*, except that, instead of adjusting line gages and windows for a single inspection, shift *registration* permanently repositions the line gages and windows in your configuration. To use shift registration, you must first configure the VIM2 module for shift compensation, as described earlier in this chapter.

In order to make use of shift registration, line gages and windows must be set to *Float* status (i.e., shift compensation *enabled*), as with shift compensation. The positions of any line gages or windows which are set for *Anchor* status (shift compensation *disabled*), and the positions of *train-thru-the-lens* windows, are not affected by shift registration.

Note: Shift registration is a viable option only when the workpiece position shift is within the feasible limits of the X/Y shift compensation; that is, when the amount of position shift detected does not force the shift any of the vision tools off the inspected image. Or, if "mode 1" (same frame compensation) of brightness compensation is selected, do not allow any vision tools to shift into the top 48 rows of the image.

In order to use *Shift Registration*, return to the *Setup* menu, and pick the *Registration* icon.

Your Action	Comment
Pick the Registration *icon on the* Setup *menu to access the* Registration *menu.*	The *Registration* icon is the second to last icon on the *Setup* menu.



Shift Registration
(continued)

After you have accessed the *Registration* menu, applying shift registration requires just two steps – picking the *Shift Registration* icon, then picking the *Store Registration* icon.

Your Action	Comment
Pick the Shift Registration *icon on the* Registration *menu.*	After you have accessed the *Registration* menu, picking the *Shift Registration* icon (the icon color is *inverted* when picked) causes shift registration to take effect immediately. That is, if the float gages detect a workpiece position which is different from the learned nominal position, the newly detected position becomes the nominal position. Each line gage or window which is configured for *float* (shift compensation *enabled*) is then repositioned in the same amount and direction as the workpiece shift. The shift of tool positions is displayed on the monitor. **Note:** If you decide not to use the new line gage and tool positions, pick the *Shift Registration* icon again, or simply pick the *OK* icon to exit the *Registration* menu without picking the *Store Registration* icon. The shift registration will be cancelled. If you decide to use the new nominal workpiece position, pick the *Store Registration* icon.
Pick the Store Registration *icon to save the new parameters.*	Picking the *Store Registration* icon stores the new nominal position and line gage and window positions as part of the current working configuration.

Here are some additional points regarding the use of the *Store Registration* icon:

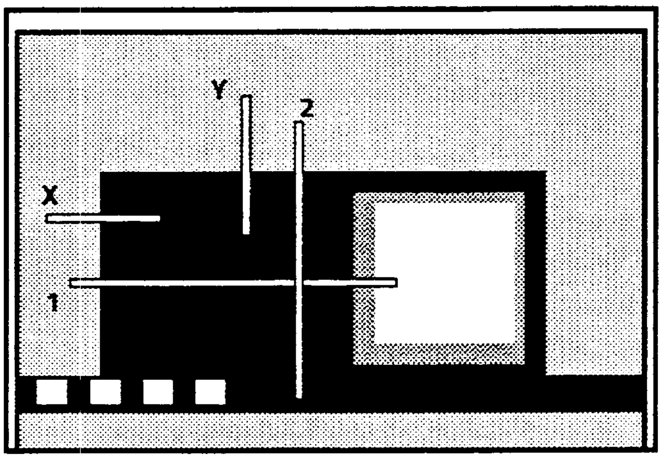
- Picking the *Store Registration* icon will also store the new lighting parameters if *brightness registration* is enabled (see Chapter 8 for brightness registration information).
- If brightness registration is already enabled when you are engaging shift registration, shift registration will be implemented based on the brightness compensated image(s).
- Picking the *Store Registration* icon before enabling shift (or brightness) registration will have no effect, except that the *Store Registration* icon will flash.
- If the amount of shift will move any of the tools offscreen (or, when *same frame compensation* is selected, into the top 48 rows of the image), picking the *Shift Registration* icon will have no effect, except that the *Shift Registration* icon will flash.
- Acceptance range limits for the registered tools are not affected by shift registration. This means that you will have to return to the *Line Gage* menu and adjust the acceptance ranges for those line gages which measure position (blob center position, blob edge position, etc.).

Shift Registration
(continued)

Example Shift Compensation

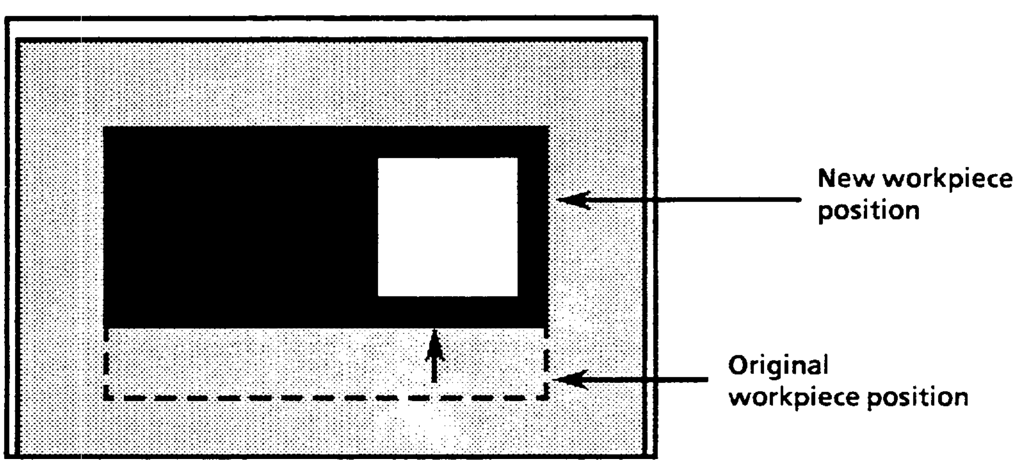
Note: This example refers to threshold image 2 acquired by using the target pattern included with this manual and following the "Quick Start Procedure" in Chapter 6.

Suppose, for example, that the black rectangle in threshold image 2 (see diagram below) represents a workpiece in the learned nominal position. Suppose that you place two line gages and a window as shown over the threshold image.



You also use float gages X and Y, and you set the float gage sequence to Y1/X2, so that any shift detected by the Y gage is applied to the X gage. You enable shift compensation for each line gage and window.

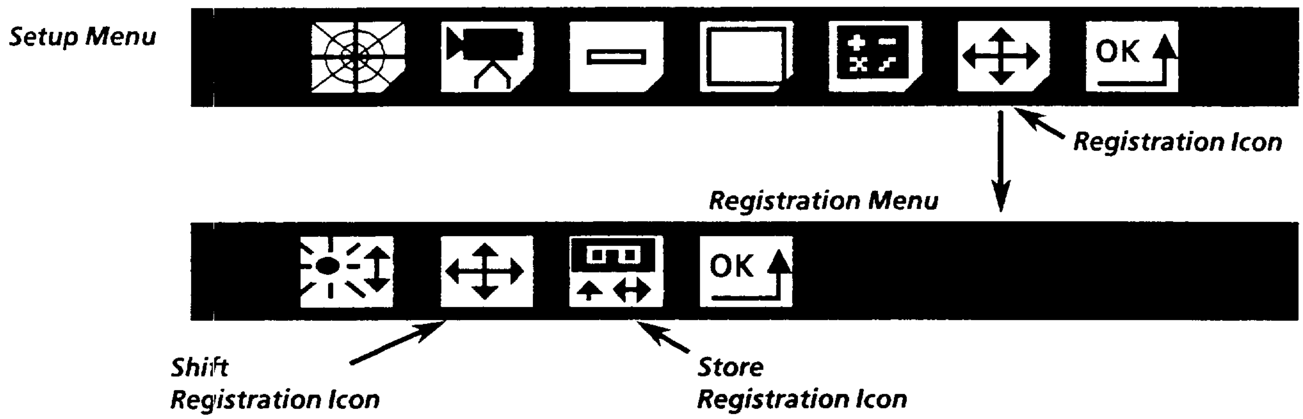
Suppose now that you change the nominal workpiece position for your inspection to a position slightly above that of the old position, so that it appears shifted, as shown below, on the monitor:



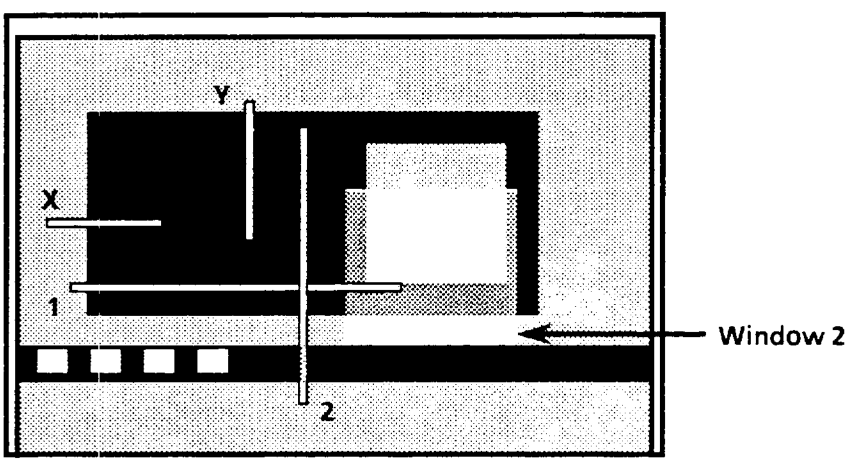
Shift Registration (continued)

Example Shift Compensation *(continued)*
With the workpiece placed in the new position, use shift registration as follows:

Your Action	*Comment*
Pick the Registration *icon on the* Setup *menu to access the* Registration *menu.*	The *Registration* icon is the second to last icon on the *Setup* menu.



With the *Registration Menu* displayed on the monitor, all the enabled tools are displayed on the screen as well – in their original positions:



As you can see, the original positions of the line gages and window are no longer adequate for the most part, and must be adjusted to comply with the position change.

Shift Registration *(continued)*

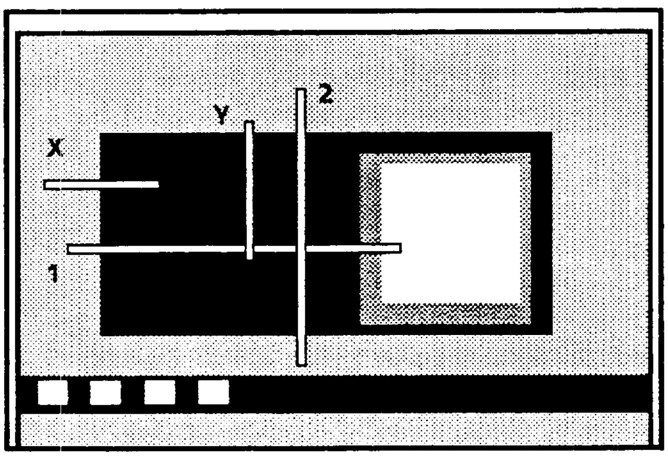
Example Shift Compensation *(continued)*

Your Action	*Comment*
Pick the Shift Registration *icon on the* Registration *menu.*	After you have accessed the *Registration* menu, picking the *Shift Registration* icon (the icon color is *inverted* when picked) causes shift registration to take effect.

Registration Menu

Shift Registration Icon

Line gages and window are repositioned in the same amount and direction as the workpiece shift. The shift of tool positions is displayed on the monitor.



The positions of line gages and window are shifted. Note the position of the Y gage has not changed in this case.

Note the position of the Y gage has not changed in this case, but the X gage is shifted vertically. This is because the float gage sequence selection was set to *Y1/X2*, so that the vertical change detected by the Y gage is applied to the X gage.

Shift Registration (continued)

Example Shift Compensation *(continued)*
If you decide not to use the new line gage and tool positions, pick the *Shift Registration* icon again, or simply pick the *OK* icon to exit the *Registration* menu without picking the *Store Registration* icon (as described below). The shift registration will be cancelled.

Your Action	Comment
Pick the Shift Registration *icon again.*	Picking the *Shift Registration* icon a second time cancels the shift registration (the *Shift Registration* icon color reverts to its original color). All line gages and windows revert to their original positions.

Registration Menu

Shift Registration Icon

Your Action	Comment
Pick the Shift Registration *icon again.*	You can enable the shift registration again by simply pick the *Shift Registration* icon again. If you decide to use the new nominal workpiece position, pick the *Store Registration* icon while the *Shift Registration* is enabled.
Pick the Store Registration *icon to save the new parameters.*	Picking the *Store Registration* icon stores the new nominal position and line gage and window positions as part of the current working configuration.

Registration Menu

Store Registration Icon

Chapter 12 Using Math Tools

Chapter Objectives

The objective of this chapter is to show how to use the *Math Tool* menu icons to configure math tools as part of a VIM2 module configuration for an inspection application (**Note:** This chapter assumes you have already read and understand the previous chapters in this manual, especially Chapters 9, 10, and 11, which discuss line gages and windows).

Some Questions and Answers about Math Tools

This section introduces the topic of *math tools* by posing some questions about math tools, and answering them.

What are math tools?
Math tools are really formulas, which you create, to combine or adjust the results of one or more inspection tools, through either arithmetic or logical operations.

How many math tools are there?
There are 12 math tools in all – numbered 1-12 – available for you to configure.

How do math tools work?
Math tools use an *operator* (such as +, –, ×, or logical *OR*) to perform an *operation* on one or more *operands* (e.g., line gage results, other math tool results, and/or *constants)* in order to obtain a *result.* This result is then compared to a preset *acceptance range*, just as window and line gage results are, in order to help provide the basis for accept or reject decisions (**Note:** The logical operators *AND* and *OR* are exceptions).

What kinds of math tool "operators" are there?
Math tool operators include:
+ (add): add together the operands
– (subtract): subtract one operand from another
× (multiply): multiply one operand times another
÷ (divide): divide one operand by a constant
(logical) *AND*: *AND* result status of the listed operands
(logical) *OR*: *OR* result status of the listed operands
Minimum: the minimum result amongst the listed tools
Maximum: the maximum result amongst the listed tools

How do you set up a math tool?
Using the *Math Tool* menu *icons*, you first select the math tool number (1-12), and enable the math tool. You then configure the math tool: you select the operator (such as +, –, ×, etc.); you select the operands (line gages, and/or windows, and/or math tools, and/or constants) which will be used in the math tool; and you then set an *acceptance range* for the math tool results – during inspection this range helps provide the basis for an accept or reject decision.

Some Questions and Answers about Math Tools *(continued)*

What can you do with math tools?
Using math tools you can, for example:
- Use addition to sum tool results
- Use multiplication or division to scale tool results
- Use subtraction to find the difference (distance or area) between two tool results
- Use addition and division to average tool results
- Use *AND* or *OR* operators to group the *accept / reject* results of different tools
- Use maximum or minimum to compare results

When do you set up a math tool?
Set up a math tool after you have:
- Set up the workstage, and have a workpiece in the nominal or ideal position (see Chapter 4), so that the workpiece features to be inspected are shown clearly in the field of view (and on the monitor).
- Set up all the line gages and windows (see Chapters 9 and 10 for information), or other math tools you will use as the operands when configuring the math tool.

Accessing the Math Tool Menu

To begin to set up a math tool, first access the *Math Tool* menu as described below.

Your Action	*Comments*
Pick the Setup *icon on the* Main *menu to access the* Setup *menu.*	If the *Main* menu is displayed, access the *Setup* menu by picking the *Setup* icon (the "open padlock"). If the *Setup* menu is already displayed (as shown below), skip this step.

Setup Menu — Math Tool Icon

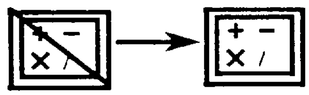
Pick the Math Tool *icon.*	Picking the *Math Tool* icon on the *Setup* menu displays the first part of the *Math Tool* menu:

Note: You cannot access the *Math Tool* menu unless at least one window is enabled, or at least one line gage (numbered from 1-22 only) is enabled.

Accessing the Math Tool Menu
(continued)

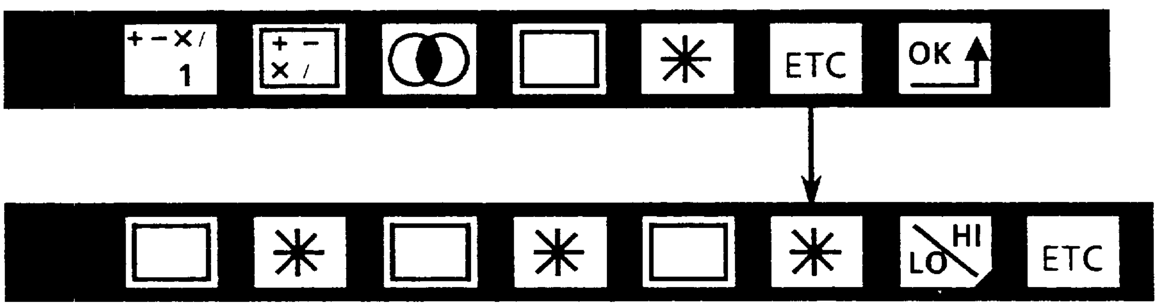
Your Action	*Comments*
Pick the second (enable/disable) *icon to* enable *the math tool.*	Picking the *enable/disable* icon toggles the icon between *enabled* and *disabled* (indicated by the diagonal slash across the icon. When this icon is set to *enabled*, this allows you to pick other icons on the *Math Tool* menu.

When you pick this icon –

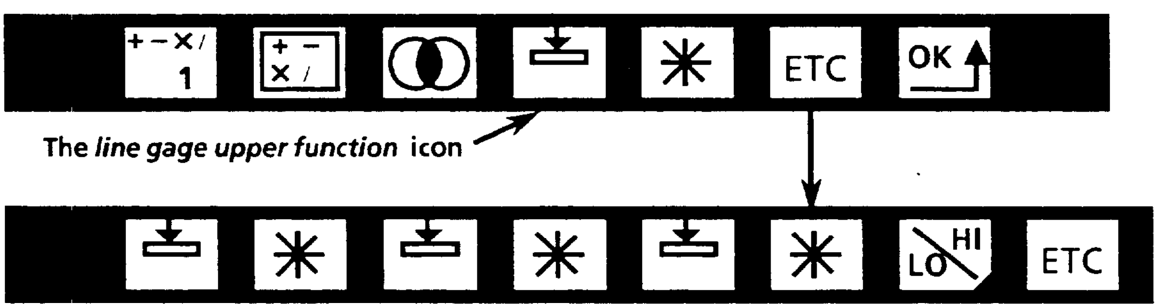


the diagonal slash disappears.

Pick the ETC *icon.*	Picking the *ETC* icon displays the second part of the *Math Tool* menu (picking *ETC* on the second part of the menu redisplays the first part of the menu).



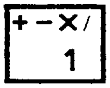
Note: If the only tools enabled are line gages (other than math tools), the first and second part of the *Math Tool* menu appear as follows, when first accessed, after you have enabled the math tool (i.e., *line gage upper function* icons replace the *window* icons):



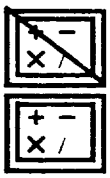
Accessing the Math Tool Menu
(continued)

Math Tool Menu

Below are descriptions of the icons on the first part of the *Math Tool* menu (other than the *ETC* and *OK* icons):

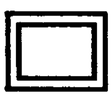


Math Tool Select – Pick this icon to select the math tool from a choice of numbers 1-12. Numbers 1-12, then back to 1 again, etc., are displayed in sequence as you repeatedly pick the icon, or hold the light pen tip down against the icon.

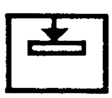


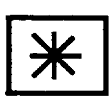
Math Tool Enable – Pick this icon to enable or disable the selected *math tool*. This icon, when picked, toggles between *enabled* or *disabled* (*disabled* is indicated by diagonal slash across the icon). You must enable a selected math tool in order to use the remaining *Math Tool* icons to configure a math tool. Also, a math tool must be enabled in order to be part of your inspection configuration.

Operator – Pick this icon to select the type of operator the math tool will use. Picking the icon repeatedly toggles the icon through eight different operator icons, in the following order – *AND* (shown at left), *OR*, *Minimum*, *Maximum*, *Add*, *Subtract*, *Multiply*, and *Divide*.



First Operand Type – Pick this icon to select which type of tool the math tool will use as the first operand. Picking the icon repeatedly toggles this icon through the available selection of operand types – *window* (shown at top left), *line gage upper function, line gage lower function*, and/or *math tool* (a *constant* operand is also available for operands other than the first). This icon works in conjunction with the *First Operand Number* icon. **Note:** If no windows are enabled, the *Line Gage Upper Function* icon is displayed (shown at left) when the math tool is *enabled.*

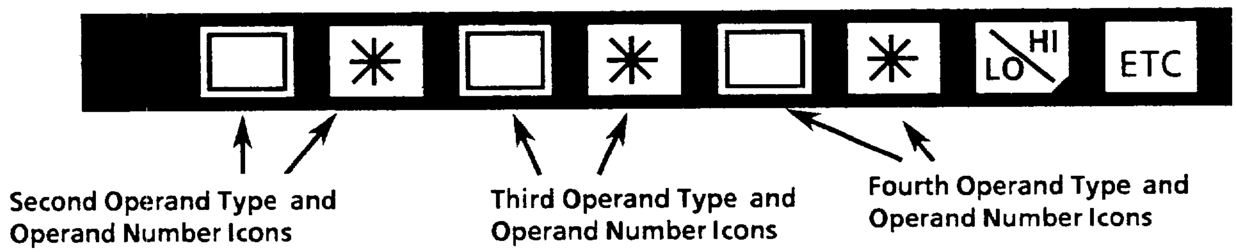


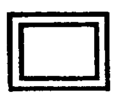


First Operand Number – This icon works in conjunction with the *First Operand Type* icon. Pick this icon to select the specific number of the tool whose type is indicated by the *First Operand Type* icon. Picking this icon repeatedly toggles this icon through the available numbers for the tool – all numbers of tools of the selected type which are *enabled* comprise the selectable operand numbers. **Note:** The asterisk-shaped icon shown at left selects *all tools* of the type indicated by the *First Operand Type* icon.

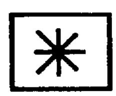
Accessing the Math Tool Menu *(continued)*

The second part of the *Math Tool* menu consists of three more *operand type* and *operand number* icons, the *Acceptance Range* icon, and the *ETC* icon. Below are descriptions of these icons (other than the *ETC* icon):

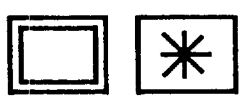




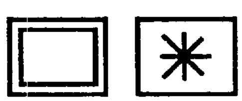
Second Operand Type – Pick this icon to select which type of vision tool the math tool will use as the *second operand.* This icon works in conjunction with the *Second Operand Number* icon to select the second operand; this icon works just like the *First Operand Type* icon, with some exceptions.



Second Operand Number – This icon works in conjunction with the *Second Operand Type* icon to select the *second operand.* This icon works just like the *First Operand Number* icon, with some restrictions.



Third Operand Type and **Number** – Pick these icons to select the third operand of the vision tool. These icons work just like the *Second Operand Type* and *Number* icons, with some exceptions.



Fourth Operand Type and **Number** – Pick these icons to select the fourth operand of the vision tool. These icons work together just like the *Third Operand Type* and *Number* icons, with some exceptions.

Acceptance Range – Pick this icon to access the *Acceptance Range* menu for the selected math tool. Use the *Acceptance Range* menu to set the upper and lower limits for the math tool results, and to read the current math tool result.

Note: The next section, "Math Tool Components," discusses the *Operator Type, Operand Type,* and *Operand Number* icons further.

Math Tool Components

When you configure a math tool, you create a formula consisting of a combination of *operator* and *operands* that act together to produce a *result*. The three terms, *operator operand*, and *result*, are defined below:

Operator – A symbol (such as +) or a term (such as *AND*) that represents an arithmetic or logical operation performed upon one or more operands (all eight of the operators are described under "The Operators" heading in this section).

Operand – A quantity or value (such as a constant, or the measured result from a specific window, or the accept / reject result from a line gage function) upon which the operator acts (all of the operands are described under "The Operands" heading in this section).

Result – The value generated by the math tool as a consequence of the operator acting upon the operands. This result is compared to a preset *acceptance range*, just as window and line gage results are, in order to help provide the basis for an *accept* or *reject* decision (**Note:** The logical operators *AND* and *OR* are exceptions to this).

To review the math tool components, first access the *Math Tool* menu as described on pages 12-2 and 12-3.

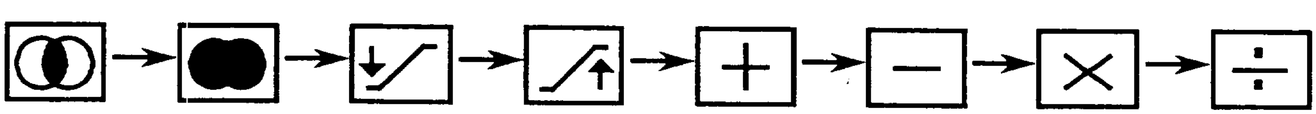
The Operators

For each math tool you configure, you select a single *operator*; the choice of operator determines the type of operation the math tool will perform involving the selected operands. Select the operator by picking the *Operator* icon.

Math Tool Menu

Operator Icon

Your Action	*Comments*
Pick the Operator *icon.*	When the math tool is first selected, the *Operator* icon is set to *AND*. Picking the icon repeatedly will cycle the icon through eight operator icons, as shown below:



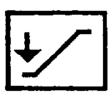
AND → OR → Minimum → Maximum → Add → Subtract → Multiply → Divide

The Operators *(continued)*

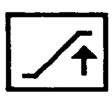
Here are brief descriptions of the operators (the operators are each described more fully later in this chapter, under the "Math Tool Examples" heading).

AND – The *AND* operator checks the *decision status* (*accept/ reject*) of each of the selected tools (operands). The *AND* result is always 0*; the *AND decision status* is:
0 – (*accept*) if all selected tools had *accept* decisions
1 – (*reject*) if at least one tool had a *reject* decision

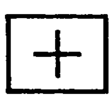
OR – The *OR* operator checks the *decision status* (*accept / reject*) of each of the selected tools (operands). The *OR* result is always 0*; the *OR decision status* is:
0 – (*accept*) if at least one tool had an *accept* decision
1 – (*reject*) if all tools had *reject* decisions
***Note:** For *AND* and *OR*, the *decision status* is displayed as the current result value in the acceptance range display. The range limits have no meaning and are ignored.



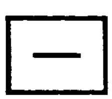
Minimum – The *Minimum* operator compares the *numerical* (*measured*) results of each of the selected tools (operands); the *Minimum* result indicates the lowest value measured amongst the results of the selected tools.



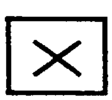
Maximum – The *Maximum* operator compares the *numerical* (*measured*) results of each of the selected tools (operands); the *Maximum* result indicates the highest value measured amongst the results of the selected tools.



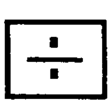
Add – The *Add* operator *adds* the *numerical* (*measured*) results of each of the selected operands; the *Add* result indicates the sum of the measured results of the operands.



Subtract – The *Subtract* operator subtracts the *numerical* (*measured*) result of one operand from that of another (only two operands are allowed, and the second can be a *constant*); the *Subtract* result indicates the *difference* (in absolute value) between the results of the two operands.



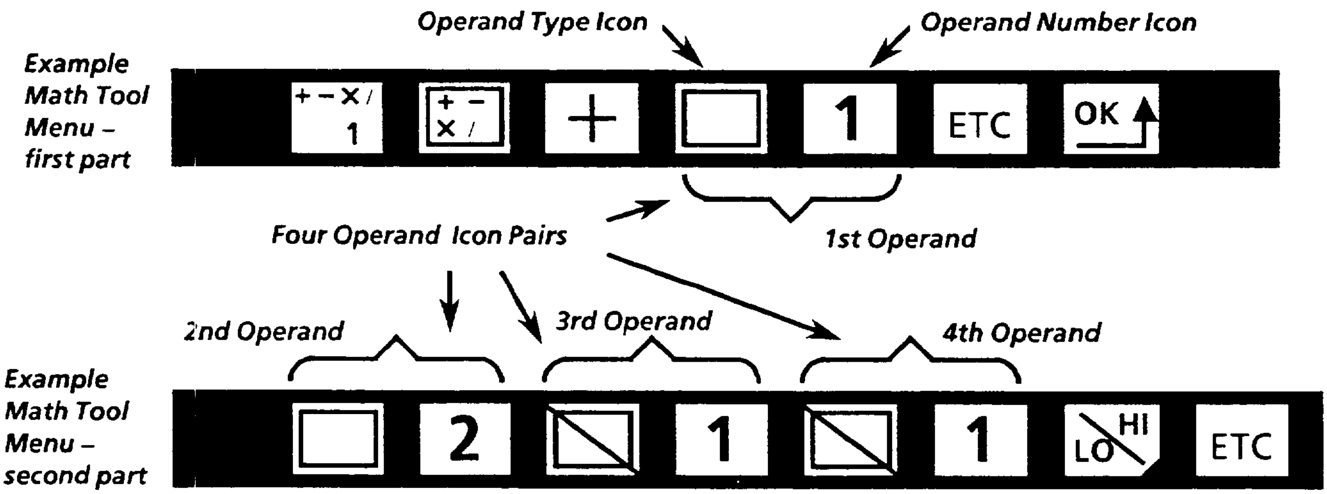
Multiply – The *Multiply* operator multiplies the *numerical* (*measured*) result of one tool (operand) by that of another (only two operands are allowed; the second can be a constant); the *Multiply* result is the *product* of the operands.



Divide – The *Divide* operator divides the *numerical* (*measured*) result of one tool (operand) by a constant (only two operands are allowed, and the second *must* be a constant); the *Divide* result indicates the *quotient* of the first operand divided by the second.

The Operands

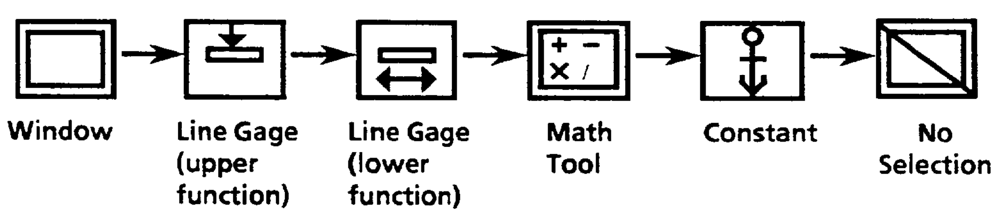
After selecting the math tool *operator*, you select the math tool *operands* by using two types of icons – the *Operand Type* and *Operand Number* icons (there are four of each of these icons in the *Math Tool* menu). The two icon types work together to select the operand; you might consider these eight icons to be four *Operand icon pairs*. Diagrammed below is a typical *Math Tool* menu (both parts), showing the four *Operand* icon pairs:



Operand Type Icons – Picking an *Operand Type* icon selects the *type* of operand – *Window, Line Gage Upper, Line Gage Lower, Math Tool, Constant,* or *No Selection.*

Note: The selection of *operand type* is subject to restrictions. Not all types are available in all cases (see "Operand Type and Number Restrictions" on page 12-10).

Your Action	Comments
Pick the First Operand Type *icon.*	Picking the icon repeatedly cycles the icon through the *available Operand Type* icons (not all operands are available in all cases). All of the *Operand Type* icons are shown below:



Note: When you pick either an *operand type* icon or *operand number* icon on the *Math Tool* menu, the selected line gage or window is displayed on the monitor over its respective threshold image. If you select a math tool as the operand, all *enabled* windows are displayed over threshold image 1; if no windows are enabled, all *enabled* line gages are displayed over threshold image 1. If the third or fourth operands are

The Operands *(continued)*

selected, then disabled, the second operand will be displayed. If the second operand is a *constant*, the first operand is displayed. If the third or fourth operand is a *constant*, the second operand is displayed.

Operand Number Icon – After selecting the *Operand Type*, picking the *Operand Number* icon selects the specific tool number (or constant value, if *constant* is operand type). The range of selectable numbers depends on the *Operand Type* icon selected –

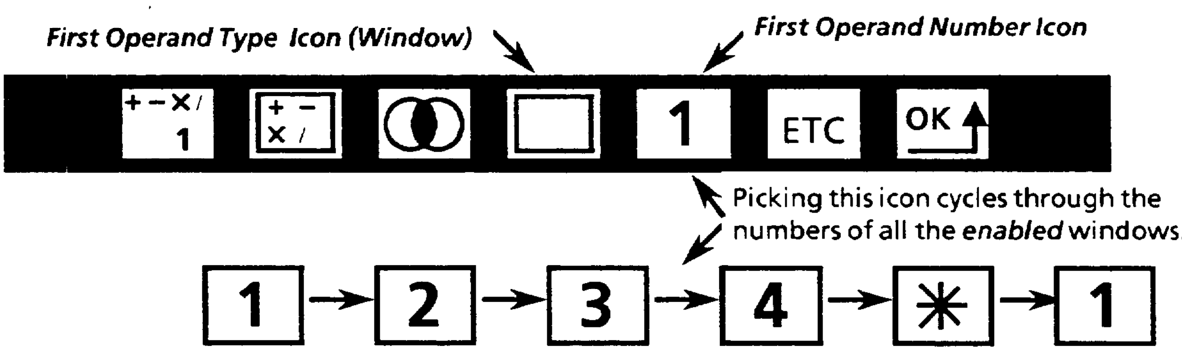
- Window (1-4, *)
- Line Gage Upper or Lower (1-22, *)
- Math Tool (1-12)
- Constant (1-30, 50, or 100)

where "*" selects *all* enabled *tools of the selected type.*

Note: The selection of *operand number* is subject to restrictions. Not all numbers are available in all cases (see "Operand Type and Number Restrictions" on the next page). For example, you are not allowed to choose a number to select a specific tool unless that tool is enabled.

Your Action	***Comments***
Pick the First Operand Number *icon.*	Picking the icon repeatedly cycles the icon through the *available* operand number icons (the illustration shown below assumes *all four windows* are *enabled*):

Example Math Tool Menu – first part



The Operands *(continued)*

Operand Type and Number Restrictions – You will encounter the following restrictions when selecting the *Operand Type* and *Operand Number* icons:

- The VIM2 module *does not* allow you to select an *operand type* if no tools of that type are currently enabled (for example, if no windows are enabled, the *Window* icon is not an option for any of the four *Operand Type* icons).
- You are not allowed to select either the *constant* or the *No Selection* icon as the *First Operand Type* icon. You are not allowed to select the *No Selection* icon as the *Second Operand Type* icon.
- If you select either the *AND* or *OR* operator, you are not allowed to select the *constant* icon as an operand type.
- You are not allowed to select a *Fourth Operand Type* icon if the *Third Operand Type* icon is *No Selection.* Also, the VIM2 *does not* allow you to pick *No Selection* as the *Third Operand Type* if a *Fourth Operand Type* other than *No Selection* is selected.
- You are not allowed to pick an *operand number* representing a tool which is disabled. For example, if only windows 1 and 2 are currently enabled, and *Window* is the selected operand type icon, you are allowed only to select *1*, *2*, or, in some cases, the *all tools* (*asterisk*) icon.
- You can select the *all tools* (*asterisk*) icon only for the *AND, OR, Minimum* and *Maximum* operators.
- You can select the *all tools* (*asterisk*) icon only for the *First Operand Number* icon; once you select *all tools* for the *First Operand Number* icon, the VIM2 does not allow you to select any other operands for that math tool (all other operand icon pairs appear the same as the first).
- If you select *Math Tool* icon as an operand type, you cannot select the *all tools* icon for operand number. Also, you are not allowed to select the number for the same math tool you are configuring. And, you are not allowed to select a particular math tool as an operand if that particular math tool contains the math tool you are configuring as an operand, or if it contains a tool which contains the math tool, etc.
- If you select the *Multiply* operator, you are allowed only to select two operands (you are allowed to use only the first and second operand icon pairs). You are not allowed to use the *all tools* icon as an operator number.
- If you select the *Divide* operator, you are allowed to select two operands only (you use only the first and second operand icon pairs). You can not use the *all tools* icon as the operator number. For the second operand type you are only allowed to use the *constant* icon.

Setting the Acceptance Range

Once you have selected the operator and operands, you can access the *Acceptance Range* menu for the math tool; by accessing the *Acceptance Range* menu, you can observe the current *result* value of the math tool, and set the upper and lower limits (the *acceptance range)* for the math tool.

As with the line gage and window acceptance ranges, when the result of the math tool is not within the specified acceptance range for an inspection, the math tool fails, resulting in an inspection *reject* decision (**Exception:** Range limits do not apply with either the *AND* or *OR* operator).

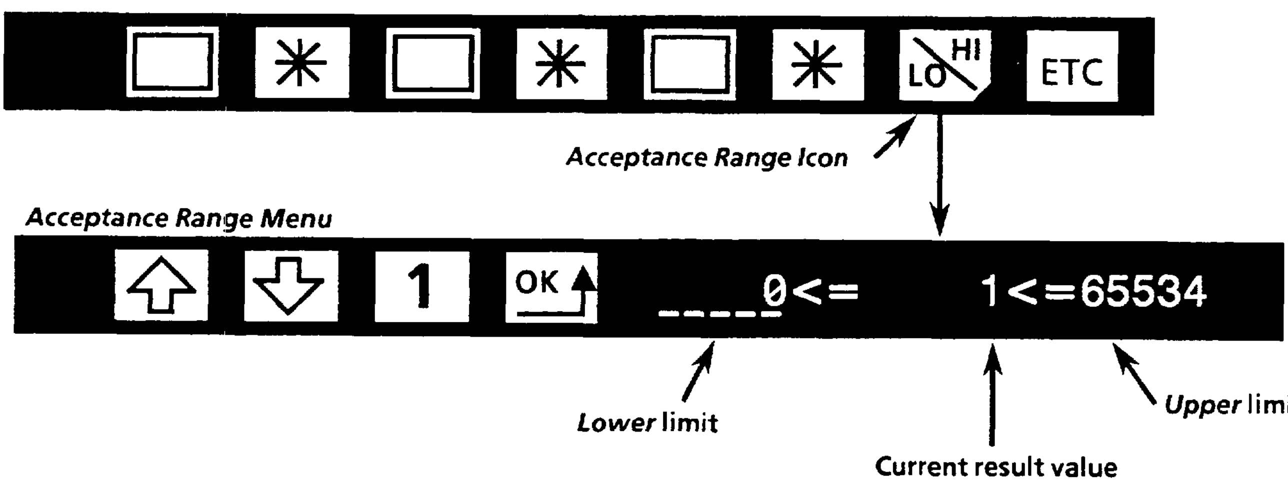
The math tool result range is subject to these limitations:

- For *logical* operations *(AND* or *OR)*, the math tool result is always 0 (zero), and the range limits do not apply. The *accept / reject* decision (0 =*Accept*, 1 = *Reject)* is instead displayed as the current result value in the acceptance range display.
- For all other operations, the math tool results are limited to a range from a lower limit of 0 to an upper limit of 65,534. If a result of greater than 65,534 is achieved, the *result* value reads 65,535, and the math tool fails.

Note: If an operand fails, the math tool result will still be computed as usual, unless the failed tool is a line gage which was unable to detect an edge. In this case, the math tool result is 65,535, and the math tool fails.

To demonstrate how to set the acceptance range for the math tool, access the *Acceptance Range* menu, as described below:

Your Action	Comments
Pick the Acceptance Range *icon on the* Math Tool *menu.*	When you pick the *Acceptance Range* ("HI/LO") icon on the *Math Tool* menu, the *Acceptance Range* menu appears:



Setting the Acceptance Range
(continued)

Your Action — **Comments**

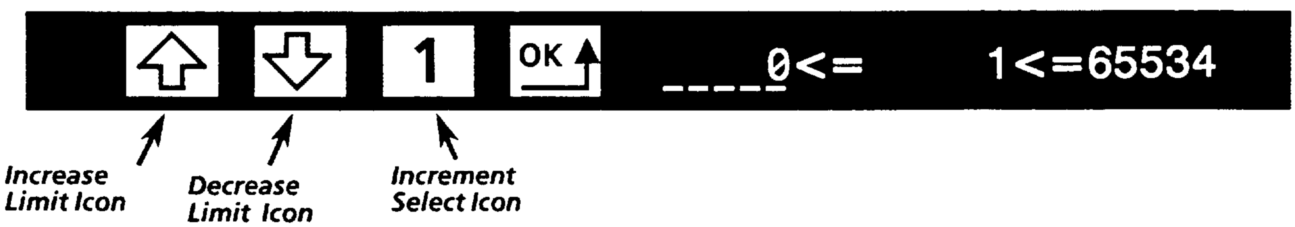
Observe the math tool result.

On the right side of the *Acceptance Range* menu are three numbers, separated by "< =" signs. The *middle* number is the current result value of the math tool.

*Select the limit (*upper *or* lower*) you want to change.*

The *lower limit* is the *left* number of the three numbers displayed on the right side of the menu, the *upper limit* is the number on the *right* (**Note:** Default values are displayed for the limits initially). To select either limit, pick that limit with the light pen. The selected limit is *underlined.*

Acceptance Range Menu



Select the increment.

Pick the *Increment Select* icon to set the amount by which the selected limit will change when you pick the *Increase Limit* or *Decrease Limit* icon. Picking the *Increment Select* icon, cycle through the increment choices – 1, 10, 100, and 1000.

Increase or decrease limit.

To *increase* the selected limit by the amount shown on the *Increment Select* icon, pick the *Increase Limit* icon. You can increase the limit to a maximum of 65,534.

To *decrease* the selected limit by the amount shown on the *Increment Select* icon, pick the *Decrease Limit* icon. You can decrease the limit to a value of 0.

Note: The VIM2 module does not allow you to set the upper limit to a value lower than the lower limit, or to set the lower limit to a value higher than the upper limit.

When finished, pick the OK *icon to display the* Math Tool *menu.*

Picking the *OK* icon at this point returns the *Math Tool* menu.

Setting the Acceptance Range (continued)

Using the Learn Mode and other Run Mode Functions

The VIM2 module features a variation of its *run* mode called the *learn* mode. In *learn mode*, the VIM2 accumulates *statistics for each enabled inspection tool,* based on a number of trial inspections. Use of the *learn* mode can be helpful in determining the optimal *acceptance range* for inspection tools and probes.

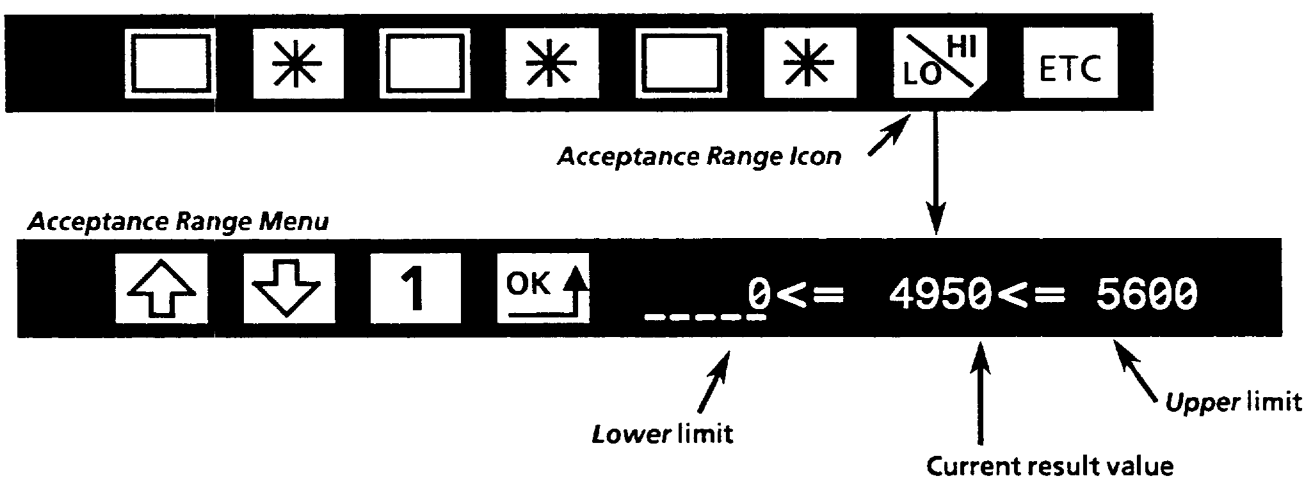
The VIM2 module also features a number of other run mode options, which can be helpful in refining your configuration.

See Chapter 13 for more information regarding use of the *learn* mode and other *run* mode options.

Accessing Statistics

When you run the VIM2 module in *learn mode*, you accumulate statistics for all enabled visions tools and probes. To observe the collected statistics for a math tool, first access the *Acceptance Range* menu on the *Math Tool* menu:

Your Action	*Comments*
Pick the Acceptance Range *icon to access the* Acceptance Range *menu.*	Picking the *Acceptance Range* icon on the *Math Tool* menu displays the *Acceptance Range* menu.



Pick the current result.	On the right side of the *Acceptance Range* menu are three numbers, separated by "< =" signs. The *middle* number is the *current result* of the math tool. When you pick this number, the statistics are displayed (see diagram next page), and the current tool display disappears from the screen.
Pick the current result again to remove the statistics display.	Picking the current result again removes the statistics from the screen, and the current tool display reappears.

Accessing Math Tool Statistics
(continued)

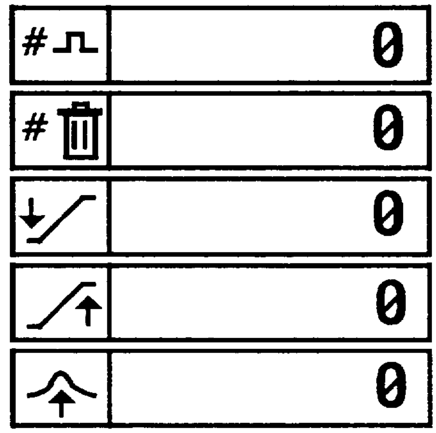
Total of triggers processed

Total failures for this math tool

Lowest result collected

Highest result collected

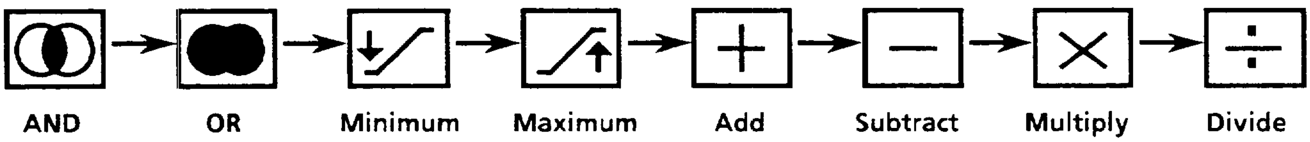
Average of all results collected



Note: The results of *failed* inspections for the math tool are not included in the *lowest*, *highest*, and *average* values. These statistics are based on passing inspections only. Therefore, for preliminary trial runs, you may want to set the acceptance range to extreme limits initially, for the purpose of collecting statistics for all inspections.

Math Tool Examples

This section provides examples demonstrating each of the eight types of math tool operation – *AND*, *OR*, *Minimum*, *Maximum*, *Add*, *Subtract*, *Multiply*, and *Divide*.



For example purposes, these procedures refer to the target pattern included with this manual. See the "Quick Start Procedure" in Chapter 6 to set up the VIM2 module, peripheral equipment, and target pattern.

Note: The VIM2 *Decision* outputs (including the *Decision* swingarm output, *Decision* LED and *Decision* discrete bit) always indicate a *reject* decision when any one of the enabled math tools, line gages, or windows fails, and indicate an *accept* decision when none of the tools fail. In order distinguish an individual math tool (or other tool) *accept or reject* result from the general inspection *accept / reject* decision, you must obtain the individual tool result from the results block (this means you must use a PLC and/or serial host). See Chapters 14 and 15, and Appendix C, for information regarding the use of the results block.

Using AND

The *AND* operator performs a logical *AND* of the *accept / reject* status of the selected operands (*constants* are not allowed as operands). For the *AND* operator, an *accept* decision (tool result within *acceptance range*) is a logic *true*; a *reject* (tool result *not* within acceptance range) is a logic *false*. If all of the selected operands are *true* (*accept*) the math tool result status is *accept*. If at least one of the operands is *false (reject)*, the math tool result status is *reject*.

The stored numerical result (in the results block) of the *AND* operation is always 0 (zero); measured results do not apply to the *AND* operator. Note, however, that the math tool *accept / reject* status is displayed when you access the acceptance range. The value listed is always either 0 or 1:

- 0 – (*accept*) if all tools had an *accept* decision
- 1 – (*reject*) if at least one tool had a *reject* decision

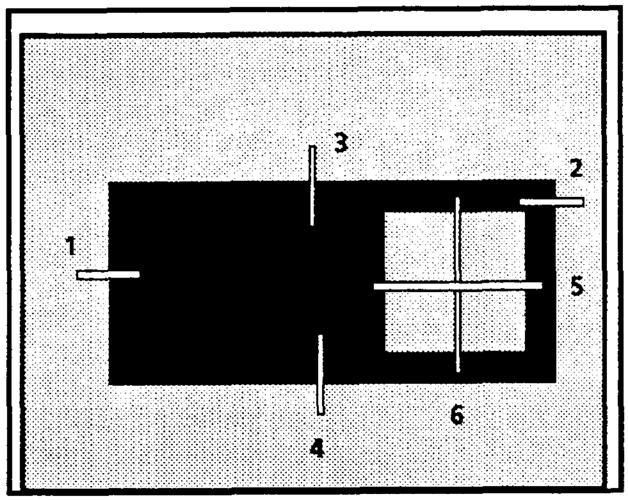
Acceptance Range – The acceptance range has no meaning for the *AND* operation; the range limits are ignored.

Using AND
(continued)

Example AND Operation

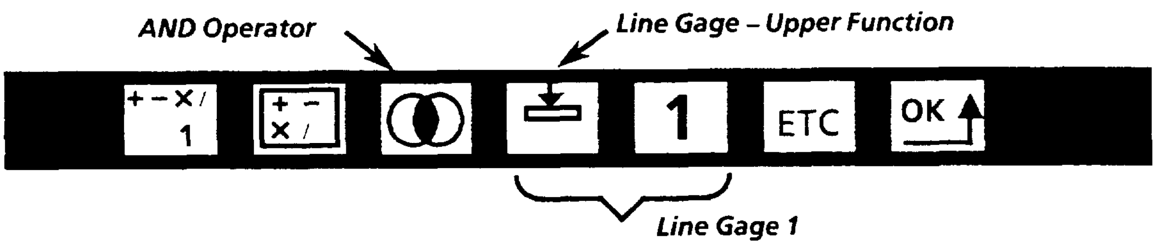
Suppose you want to measure two distinct workpiece features, such as the large black rectangle and small white square in threshold image 2 (shown below). For example, locate the edges of the *black rectangle* using upper (edge) functions of line gages 1-4; measure the width of the *white square* with lower (width) functions of line gages 5 and 6.

Line Gages 1-6 shown on threshold image 2 (when configuring the line gages, only one line gage is displayed at a time).

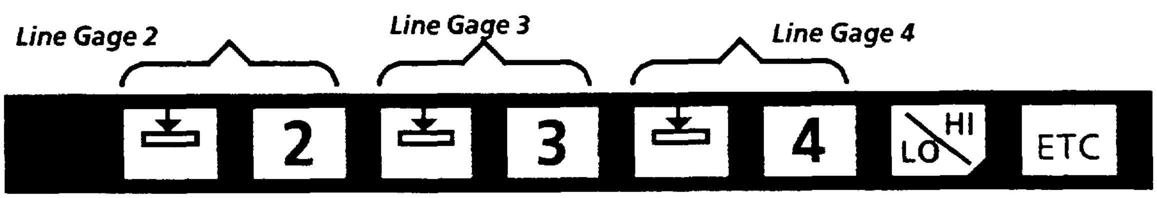


You can group all the results for the black rectangle measurements (upper functions of line gages 1-4), for example, using math tool 1, and the *AND* operator:

Example Math Tool Menu – first part



Example Math Tool Menu – second part



The math tool shown above *AND*s together the results of the upper functions of line gages 1, 2, 3, and 4. That is, if any of the upper functions of the four line gages fails, the math tool fails, and the decision status is 1 (*reject*); if none fail, the math tool passes, and the decision status is 0 (*accept*). Then, by simply checking for a 1 in the decision status of math tool 1 in the results block, you can find out if an inspection failure was due to one of the four measurements of the large rectangle.

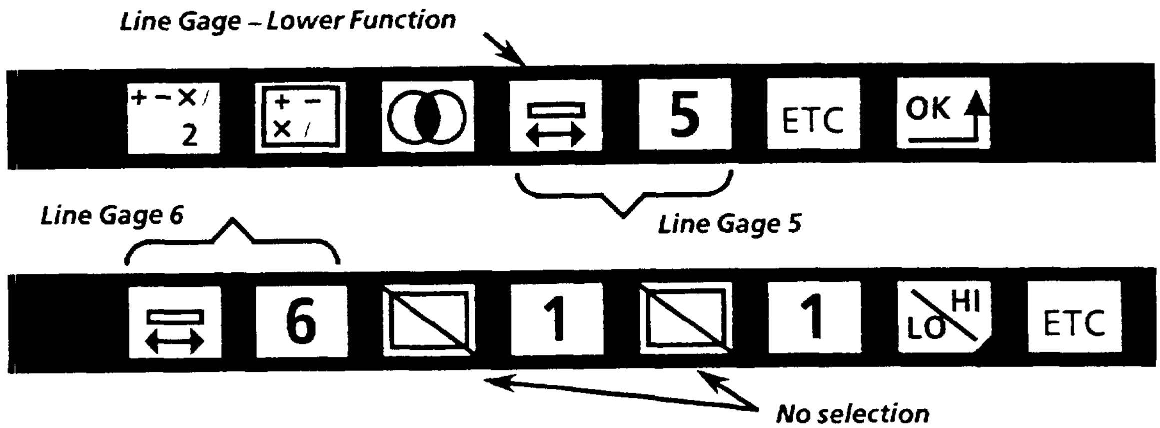
Using AND (continued)

Example AND Operation *(continued)*

In a similar fashion, you can *AND* together the results of the two line gages which measure the white square. You can group the results for white square measurements (lower functions of line gages 5 and 6), for example, using math tool 2 and the *AND* operator, as shown below:

Example Math Tool Menu – first part

Example Math Tool Menu – second part



The above math tool 2 *AND*s together the decision status of the lower functions of line gages 5 and 6. That is, if either of the two line gages fails, the math tool fails, and the decision status is 1 (*reject*); if neither fails, the math tool passes, and the decision status is 0 (accept). By simply checking for a 1 in the decision status of math tool 2 in the results block, you can find out if an inspection failure was due to one of the measurements of the white square.

Using OR

The *OR* operator performs a logical *OR* of the *accept / reject* status of the selected operands (*constants* are not allowed as operands). For the *OR* operator, an *accept* decision (tool result within *acceptance range*) is a logic *true*; a *reject* (tool result *not* within *acceptance range*) is a logic *false*. If at least one of the selected operands is *true* (*accept*) the math tool result status is *accept*. If all of the operands are *false (reject)*, the math tool result status is *reject*.

The stored *numerical* result of the *OR* operation is always 0 (zero); measured results do not apply to the *OR* operator. Note, however, that the *OR accept / reject* status is displayed when you access the acceptance range. The value listed is always either 0 or 1:

- 0 – (*accept*) if any selected tool had an *accept* decision
- 1 – (*reject*) if none of the tools had an *accept* decision

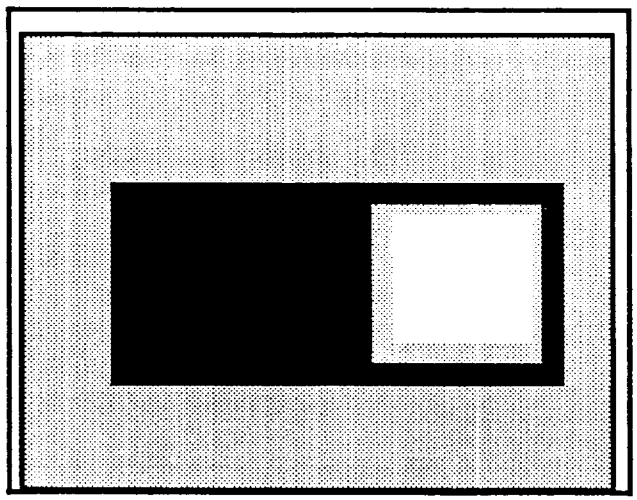
Acceptance Range – The acceptance range has no meaning for the *OR* operation, the range limits are ignored.

Using **OR** *(continued)*

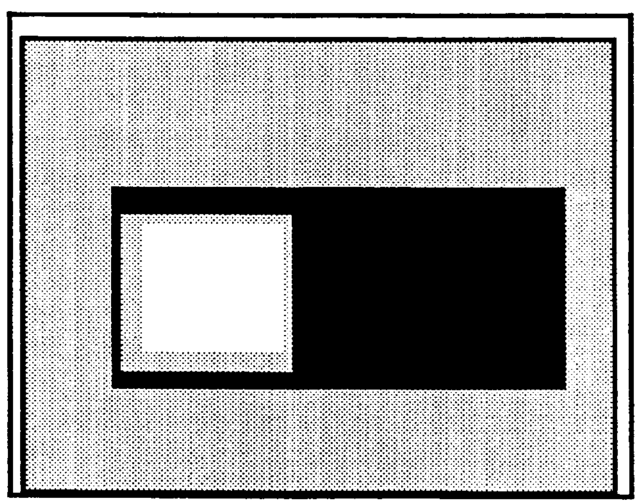
Example OR Operation

Suppose you want to measure the size of a distinct workpiece feature, such as the small white square in threshold image 2 (shown below), using a rectangular window to count white pixels. Suppose the workpiece can be positioned so that the small white square might end up on either the right side of the large black rectangle, or on the left side (the workpiece is rotated 180 degrees).

In this case, you decide to use two rectangular windows (windows 2 and 3), positioning one over each of the two possible locations of the white square (**Note:** Threshold images 2 and 3 must be set up identically to show the white square box; windows 2 and 3 have the same acceptance range; window 3 is set up with the workpiece rotated so the white square appears on the left side of the black rectangle).



Window 2 positioned on threshold image 2

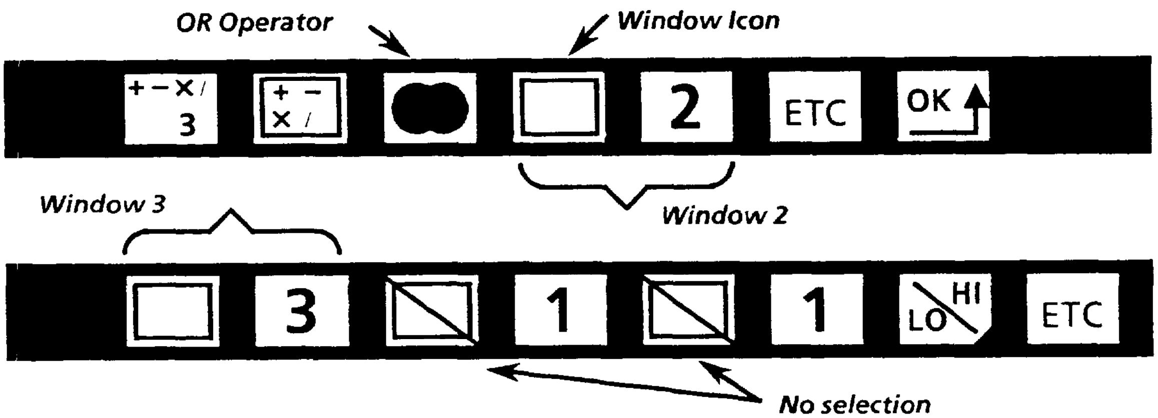


Window 3 positioned on threshold image 3 with workpiece rotated

Using OR *(continued)*

Example OR operation *(continued)*

In using windows 2 and 3 to measure the same feature in two different possible locations, only one of the two windows can pass during any one inspection (the white square can only be in one of the two positions). However, if *both* windows fail during the inspection, then the white square box fails to meet the measurement criteria. You can use a math tool with an *OR* operator to combine the window 2 and 3 results into a single *accept / reject* result, as shown below:



Example Math Tool Menu – first part

Example Math Tool Menu – second part

The above math tool 3 *OR*s together the results of windows 2 and 3. That is, if either one of the windows passes, the math tool 3 result is 0 (*accept*); if both fail, the math tool 3 result is 1 (*reject*). By checking the result of math tool 3 in the results block, you can find out if either window passes, or if both failed. And, if either window passes, you know that the white square box passes the inspection criteria.

Using Minimum

The *Minimum* operator compares the selected operands (the operands are either the measured results of selected vision tools, or a selected constant value). The math tool outputs the minimum value among the selected operands.

Example Minimum Operation

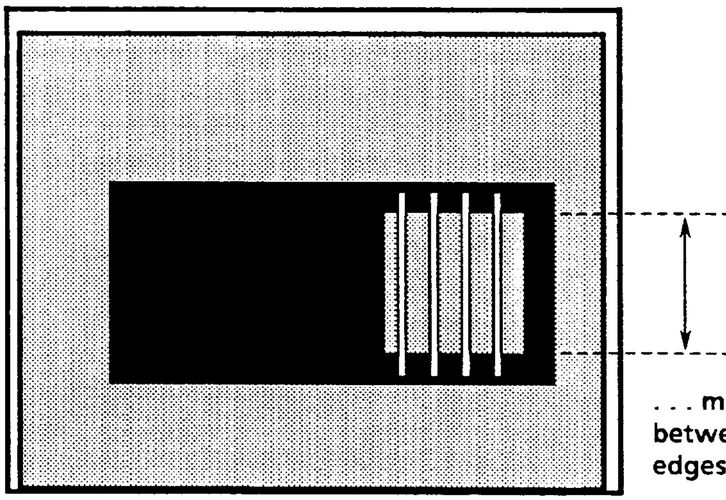
Suppose you want to measure the distance between the edges of two workpiece features, and that this distance must be above a certain minimum for the workpiece to be acceptable. To do this, you can use several line gages to measure the "gap" between the two edges, then use math tool with the *Minimum* operator to determine the lowest result among the selected line gages. You can then set the *acceptance range* of the math tool to reject the workpiece if the "gap" is too small at any point along the edges.

Using Minimum *(continued)*

Example Minimum Operation *(continued)*
In this case, for example, you decide to use line gages 1-4 to measure the distance across the white square, at various points, in threshold image 2 (see diagram below). Then you use a math tool to find the minimum result among the four line gages. If this result is too small, the part is rejected.

Note: In this case, the line gages are each set to *vertical*, and the *Color* icon of each is set to *white* pixels. The function icon of each is set to the *first* function icon – see page 9-18 – so that the line gage lower function counts the number of pixels across white square. The *acceptance ranges* for each of the line gages' functions is set to extreme values, so that the line gages will **not** fail individually (instead, the acceptance range of the math tool with the *Minimum* operator is set to distinguish between acceptable or rejected workpieces).

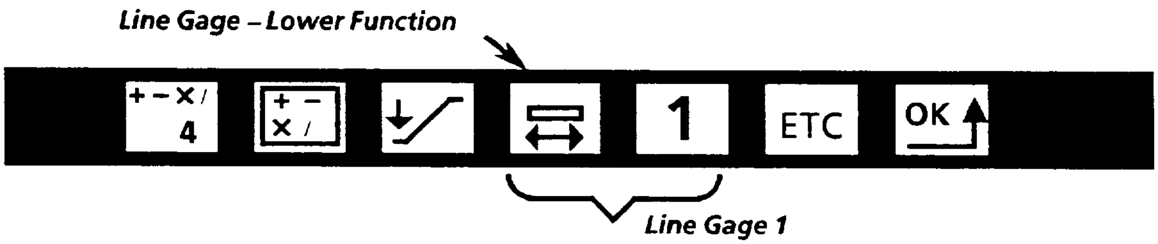
Line Gages 1-4 shown on threshold image 2 . . .



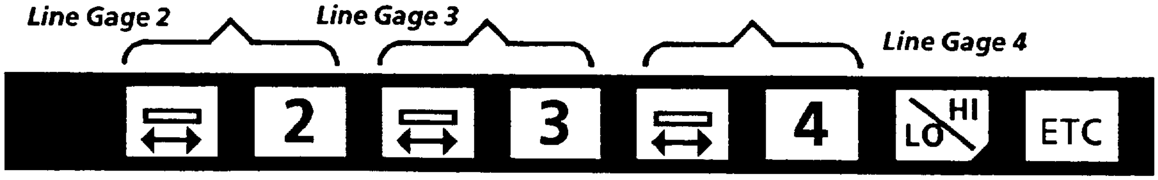
. . . measuring the "gap" between the two edges

The math tool 4 shown below compares the results of the lower functions of line gages 1-4. The math tool result is the lowest of the four line gage results. You set the acceptance range so that if the math tool result is too small, the math tool fails, and the VIM2 outputs a *reject* decision.

Example Math Tool Menu – first part



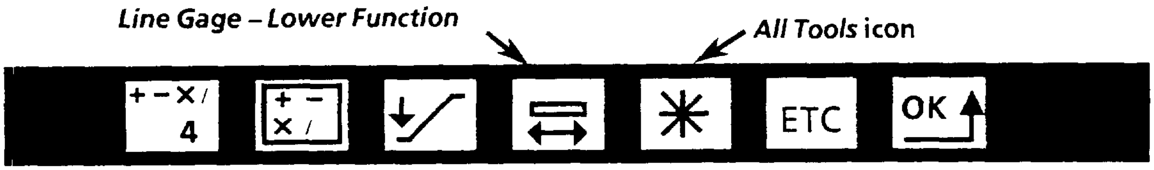
Example Math Tool Menu – second part



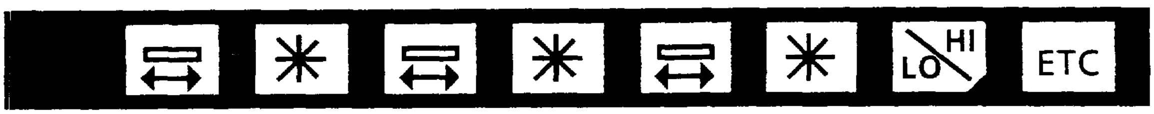
Using Minimum *(continued)*

Example Minimum Operation *(continued)*
Note: In the example, if the line gages 1-4 are the only line gages enabled for this inspection, you can use the *all tools* (*asterisk*) icon as the *First Operand Number* icon in order to *select all line gage lower functions.* You select the *all tools* icon as the *First Operand Number* icon; all other *Operand Type* and *Operand Number* icons are automatically set the same way, as shown below:

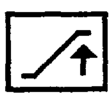
Example Math Tool Menu – first part



Example Math Tool Menu – second part



Using Maximum



The *Maximum* operator compares the selected operands (the operands are either the measured results of selected vision tools, or a selected constant value). The math tool outputs the maximum value among the selected operands.

Example Maximum Operation
Suppose you want to measure the height of a workpiece feature and that the height can vary in places, but if the height exceeds a certain measurement, the workpiece must be rejected. To do this, you can use several line gages to measure the height of the feature, then use math tool with the *Maximum* operator to determine the maximum result among the line gages. You can then set the acceptance range of the math tool to reject the workpiece if the height is too great at any point along the feature.

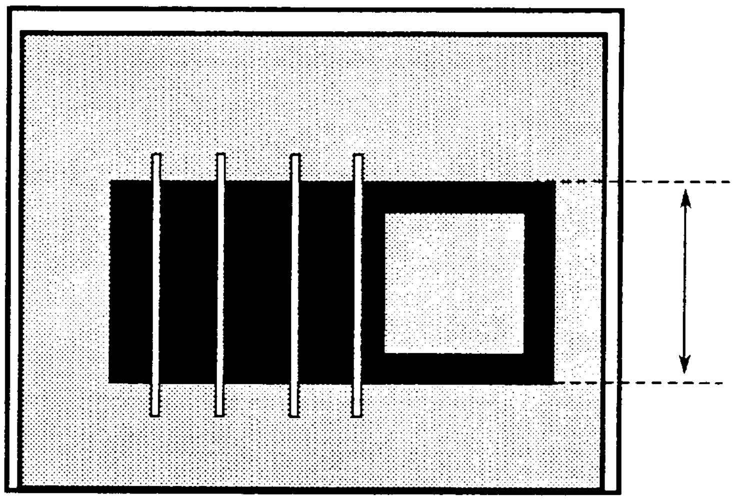
In this case, for example, you decide to use line gages 1-4 to measure the height of the black rectangle, at various points, in threshold image 2 (see diagram next page). Then you use a math tool to find the maximum result among the four line gages. If this result is too high, the part is rejected.

Using Maximum *(continued)*

Example Maximum Operation *(continued)*

Note: In this case, the line gages are each set to *vertical*, and the *Color* icon of each is set to *black* pixels. The function icon of each is set so that the line gage lower function counts the number of pixels across the black rectangle. The *acceptance range* for each of the line gages' functions is set to extreme values, so that the line gages will **not** fail individually (instead, the acceptance range of the math tool with the *Maximum* operator is set to distinguish between acceptable or rejected workpieces).

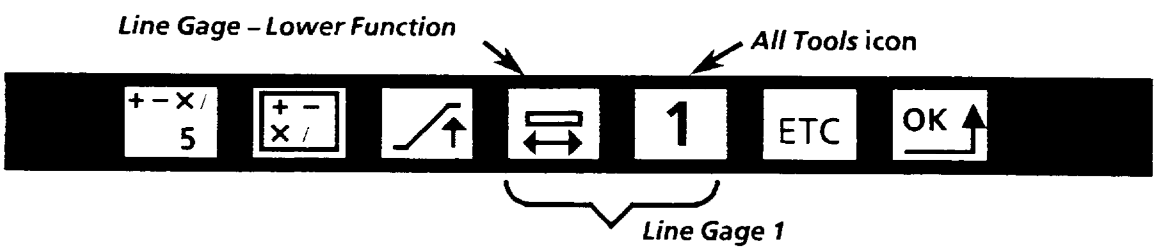
Line Gages 1-4 shown on threshold image 2 . . .



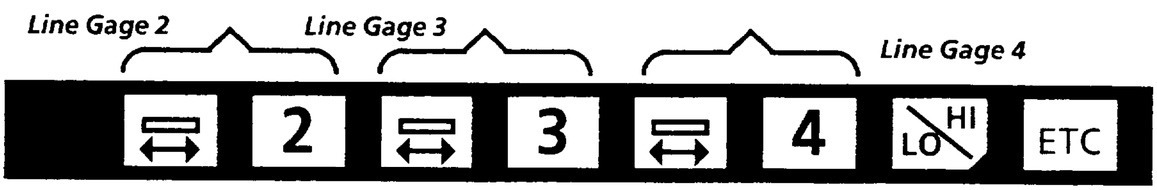
. . . measuring the height of the black rectangle

The math tool 5 shown below compares the results of the lower functions of line gages 1-4. The math tool result is the maximum value among the line gage results. You set the acceptance range so that if the math tool result is too high, the math tool fails, and the VIM2 outputs a *reject* decision.

Example Math Tool Menu – first part

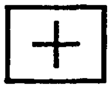


Example Math Tool Menu – second part



Note: As noted with the *Minimum* example on page 12-21, if line gages 1-4 are the only line gages enabled for this inspection, you can select the *all tools* (*asterisk*) icon for the *First Operand Number* icon to select all line gage lower functions.

Using Add



The *Add* operator *adds* the numerical results of all the selected operands; the math tool *Add* result indicates the sum of the numerical results of all the operands.

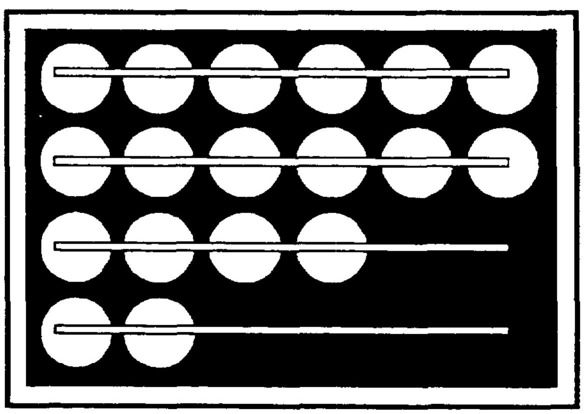
Example Add Operation

Suppose that your process includes packing cans into cartons, and that you must count the the number of cans in each carton at a certain point. The number of cans in a carton determines the line speed of the conveyor.

In this case, you use line gages 1-4, each counting the number of cans in a row in the carton (see diagram below). Then you use a math tool to total the results of the four line gages. The math tool total (the total number of cans) is used as a factor in determining the conveyor line speed.

Note: In this case, the line gages are each set to *horizontal.* The cans appear white in the monitor image, against a black background, so the *Color* icon of each of the line gages is set to *white.* The function icon of each line gage menu is set to the *ninth* function icon – see Chapter 9 – so that the line gage upper function counts the *number of blobs* (each can is counted as a *blob* by the line gages). The *acceptance range* for each of the line gages' functions is set to extreme values, so that the line gages will **not** fail individually.

Line Gages 1-4 shown . . .



. . . counting the number of cans (white blobs) in the carton

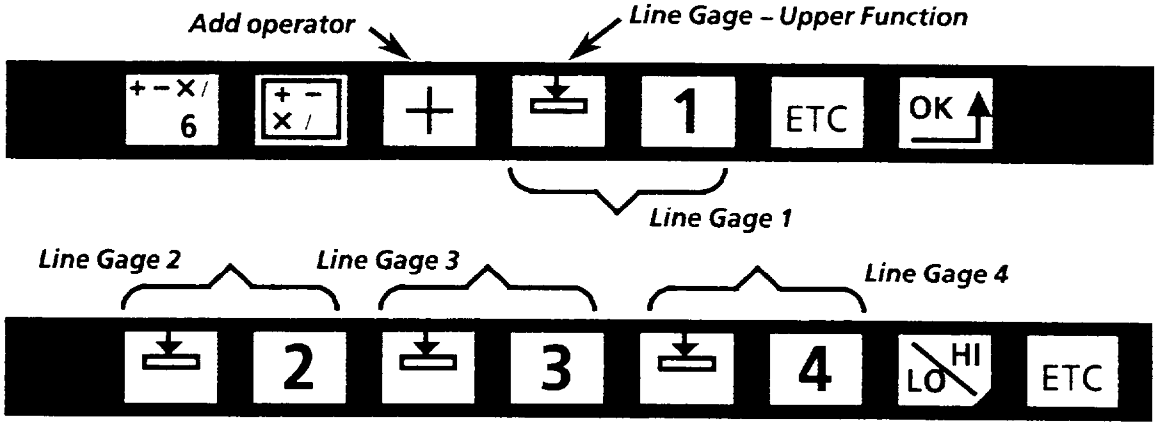
Using Add (continued)

Example Add Operation *(continued)*

The math tool 6 shown below adds the results of the lower functions of line gages 1-4. The math tool result is the total of the four line gage results.

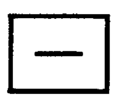
Example Math Tool Menu – first part

Example Math Tool Menu – second part



Note: You can select a *constant* as the *operand type* for any but the *First Operand Type* icon. Also, you are not allowed to use the *all tools* (*asterisk*) icon for the *Operand Number* icon with the *Add* operator.

Using Subtract



The *Subtract* operator subtracts the *numerical* (*measured*) result of one operand from that of another (only two operands are allowed, and the second can be a *constant*); the *Subtract* result indicates the *difference* (in absolute value) between the results of the two operands.

Example Subtract Operation

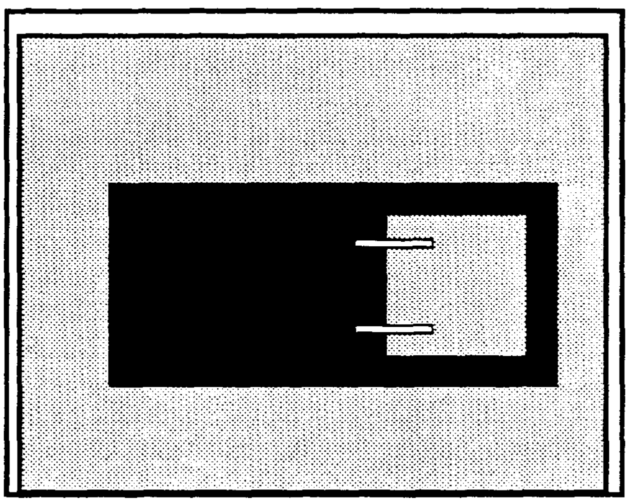
Suppose you want to determine whether a label is attached correctly on a workpiece (that is, with the side of the label exactly perpendicular to the horizontal axis of the field of view of the camera). To do this, you can, for example, use two horizontal line gages to locate the edge of the label, one near the label top, and one near the bottom. Then you can use a math tool with the *Subtract* operator to find out if there is any difference in line gage edge location (**Note:** If the side of the label is perpendicular, then the two line gage results are identical; subtracting one from the other would yield a zero result). You can then set the acceptance range of the math tool to reject the workpiece if the *result* indicates the label is crooked beyond an acceptable tolerance.

Using **Subtract**
(continued)

Example Subtract Operation *(continued)*
In this case, for example, let's suppose that the white square in threshold image 2 represents a label (see diagram below). Suppose you decide to use line gages 1 and 2 to locate the edge of the label near the top and bottom. Then you use a math tool to *subtract* one result from the other. If the label is perfectly straight, the result is 0. Suppose you want to allow some variance -so that, if the result is, say, greater than 3, the part is rejected. You would set the math tool acceptance range so the upper limit is 3, the lower limit 0.

Note: In this case, the line gages are each set to *horizontal*, and the *Color* icon is set to *white* pixels. The function icon of each line gage menu is set to the *first* function icon – see Chapter 9 – so that the line gage upper function locates the left edge of the label. The *acceptance range* for each of the line gages' functions is set to extreme values, so that the line gages will **not** fail individually (instead, the acceptance range of the math tool with the *Subtract* operator is set to distinguish between acceptable or rejected workpieces).

Line gages 1 and 2 shown on threshold image 2 . . .



. . . locating the position of the left edge of the white square.

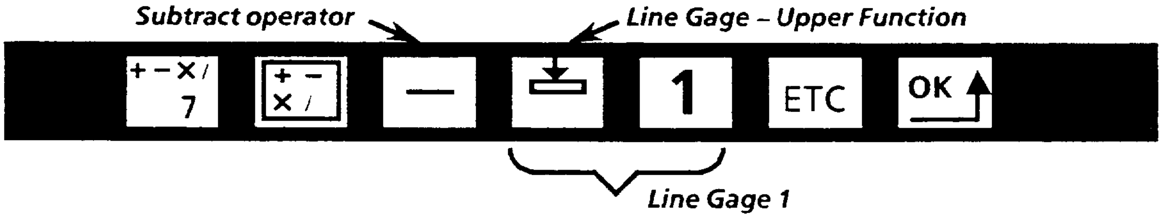
Note: When using the *subtract* operator, you are allowed to select only *two* operands; the *Third* and *Fourth Operand Type* icons are automatically set to *No Selection*. You are allowed to select a *constant* as the *Second Operand Type* icon, if desired. Also, you are not allowed to select the *all tools* (*asterisk*) icon for the *Operand Number* icon with the *Subtract* operator.

The math tool 7 shown on the next page subtracts the result of the upper function of line gage 2 from the result of the upper function of line gage 1. The math tool result is the difference in absolute value between the two results. You set the *acceptance range* so that if the math tool result is too high, the math tool fails, and the VIM2 outputs a *reject* decision.

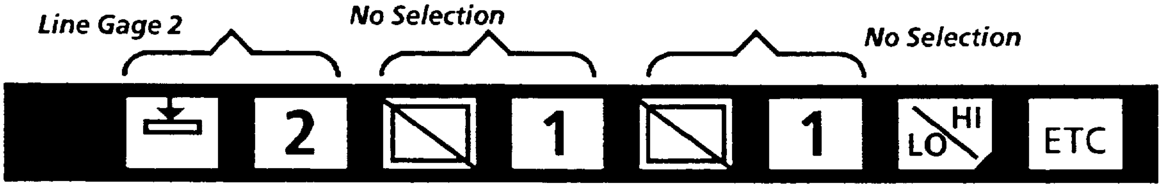
Using Subtract (continued)

Example Subtract Operation *(continued)*

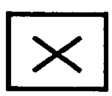
Example Math Tool Menu – first part



Example Math Tool Menu – second part



Using Multiply



The *Multiply* operator multiplies the *numerical* (*measured*) result of one operand by that of another (only two operands are allowed, and the second can be a *constant*); the *Multiply* math tool result indicates the *product* of the two operands.

Example Multiply Operation

Suppose you want to determine the area of a rectangular workpiece feature, and you have already used all of the four windows available. Using two line gages, and the math tool with the multiplier operator, you can compute the area of the rectangular feature. To do this, you can, for example, use a vertical line gage to measure the feature height, and use a horizontal line gage to measure the feature width. Then you can use a math tool with the *Multiplier* operator to multiply the two line gage measurements, and compute the feature area. You can then set the math tool *acceptance range* to accept or reject the workpiece based on the result.

In this case, for example, let's suppose we want to measure the area of the white square in threshold image 2 (see diagram on the next page). Suppose you decide to use line gages 1 and 2 to measure the height and width. Then you use a math tool to *multiply* one result by the other.

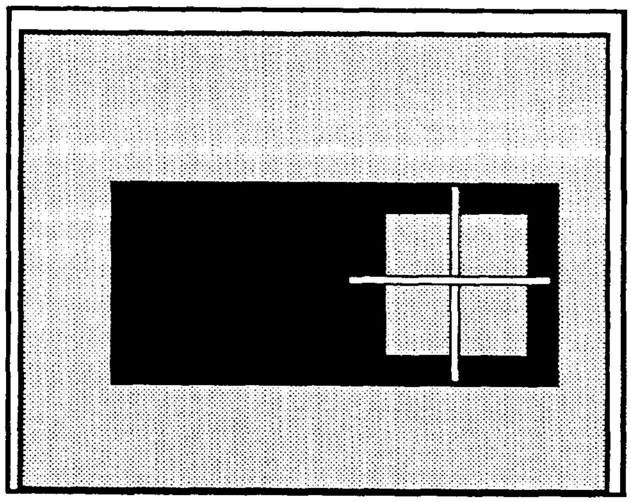
Note: When using the *multiply* operator, you are allowed to select only *two* operands; the *Third* and *Fourth Operand Type* icons are set to *No Selection*. You are allowed to select a *constant* as the *Second Operand Type* icon, if desired. Also, you are not allowed to select the *all tools* (*asterisk*) icon for the *Operand Number* icon with the *Multiply* operator.

Using Multiply *(continued)*

Example Multiply Operation *(continued)*

In this case, one line gage is set to *horizontal*, and the other to vertical. The *Color* icon of each line gage is set to *white* pixels. The function icon of each line gage menu is set to the *first* function icon – see Chapter 9 – so that the line gage lower function counts the white pixels across the white square. The *acceptance range* for each of the line gages' functions is set to extreme values, so that the line gages will **not** fail individually (instead, the acceptance range of the math tool with the *Multiply* operator is set to distinguish between acceptable or rejected workpieces).

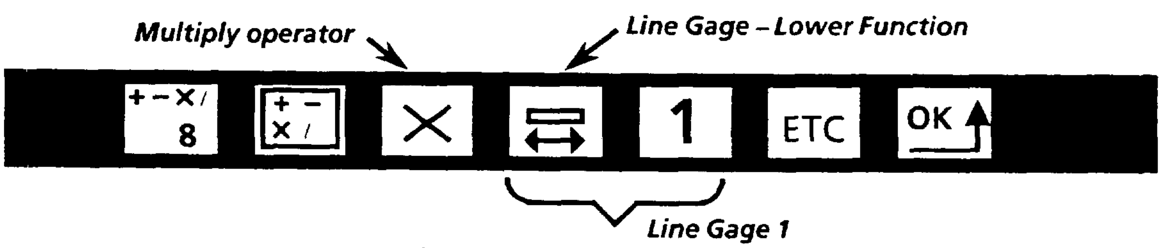
Line gages 1 and 2 shown on threshold image 2 . . .



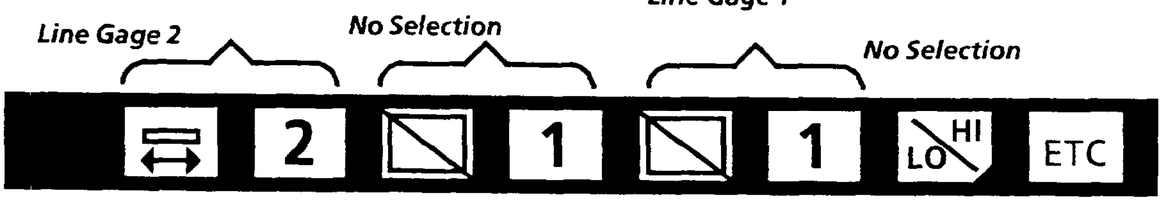
. . . measuring the height and width of the white square.

The math tool 8 shown below multiplies the result of the lower function of line gage 1 by the result of the lower function of line gage 2. The math tool result is the product of the two results. You set the *acceptance range* so that if the math tool result is too high, or too low, the math tool fails, and the VIM2 module outputs a *reject* decision.

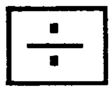
Example Math Tool Menu – first part



Example Math Tool Menu – second part



Using Divide



The *Divide* operator divides the *numerical* (*measured*) result of one tool (operand) by a constant (only two operands are allowed, and the second *must* be a constant); the *Divide* math tool result indicates the *quotient* of the first operand divided by the second (a *constant*).

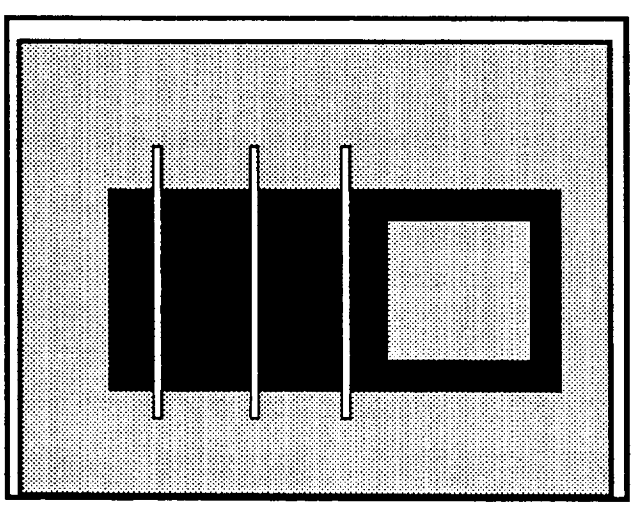
Example Divide Operation

Suppose you want to determine the *average* of three line gage measurements of a certain workpiece feature. You can do this by using three line gages, and by using a math tool with the *Add* operator to total the results from the line gages. Then, using a second math tool, you can use the *Divide* operator to divide the result of the first math tool by three, and thus obtain the average measurement of the three line gages. You can then set the math tool *acceptance range* to accept or reject the workpiece based on the result.

In this case, for example, let's suppose we want to *average* three measurements of the height of the black rectangle in threshold image 2 (see diagram below). Suppose you decide to use line gages 1-3 to measure the height. You use math tool 9 to *add* the three results. Then, you use math tool 10 to *divide* the result of the first math tool by three to obtain the average result.

In this case, the three line gages are set to *vertical*, and positioned over the black rectangle in threshold image 2 (see diagram below). The *Color* icon of each line gage is set to *black* pixels. The function icon of each line gage menu is set to the *first* function icon – see Chapter 9 – so that the line gage lower function counts the black pixels across the black rectangle. The *acceptance range* for each of the line gages' functions is set to allow some tolerance, so that a line gage will fail individually if a measurement is too extreme.

Line gages 1 -3 shown on threshold image 2 . . .



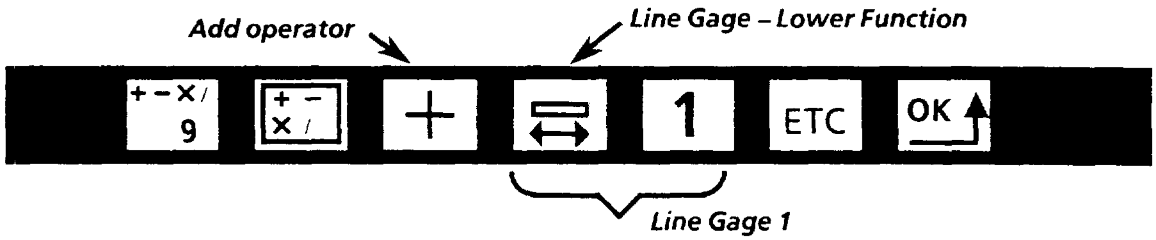
. . . measuring the height of the black rectangle.

Using Divide *(continued)*

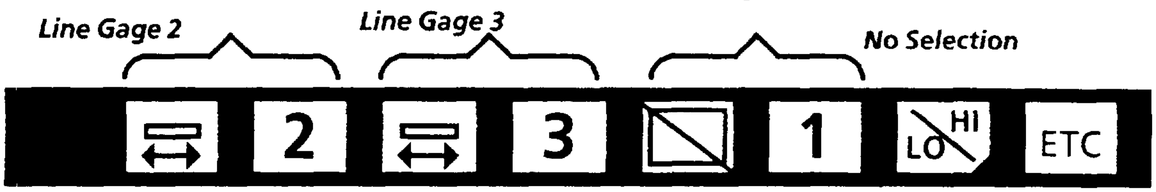
Example Divide Operation *(continued)*

The math tool 9 shown below adds the results of the lower functions of line gage 1-3. The math tool result is the total of the three results. You can set the *acceptance range* to extreme values so that the math tool will not fail.

Example Math Tool Menu – first part

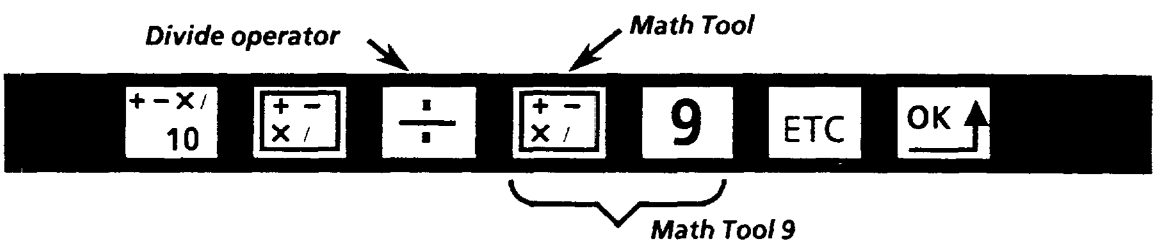


Example Math Tool Menu – second part

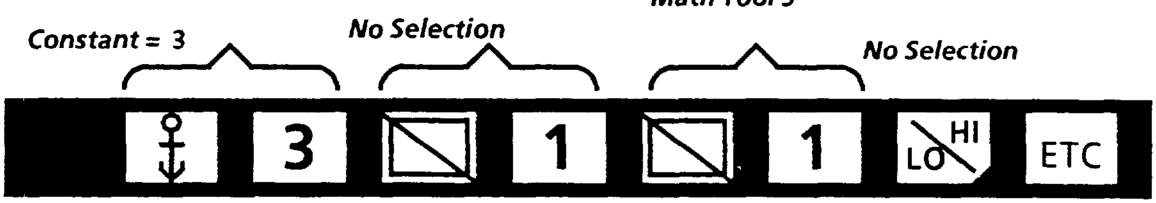


The math tool 10 shown below divides the result of math tool 9 by the *constant* 3. The math tool result is the average of the measurement of the three line gages. You set the *acceptance range* so that if the math tool 10 result is too high, or too low, the math tool fails, and the VIM2 module outputs a *reject* decision.

Example Math Tool Menu – first part



Example Math Tool Menu – second part



Note: When using the *Divide* operator, you are allowed to select only *two* operands; the *Third* and *Fourth Operand Type* icons are set to *No Selection*. The *Second Operand Type* must be a *constant* (*anchor icon*). Also, you are not allowed to select the *all tools* (*asterisk*) icon for the *Operand Number*.

(This page intentionally left blank)

Chapter 13 Run Modes and Archiving

Chapter Objectives

The objectives of this chapter are to show how to use the *Main* menu icons to select the VIM2 module's *run mode* options, how to transfer configurations to and from EEPROM using the *Archive* menu icons, and how to exit the *setup* mode and begin operation in the *run* mode..

Run Modes and Archiving

Using the *Main* and *Archive* menu icons, you can perform the following:

Select arm mode: Select from four different arm modes, which allow you to halt operation upon reject, to freeze the image upon reject, to halt operation upon each inspection, or to continue normal operation upon reject.

Select standard or learn mode: Select *learn* mode for trial inspections, to enable the VIM2 module to collect statistics based on results of each vision tool and probe. Or select *standard* mode to operate without collecting statistics.

Reset statistics: Set all statistics to 0 (zero).

Store and retrieve configurations from EEPROM: Use the *Archive* menu to store up to two different configurations in the VIM2 module's EEPROM, or use a configuration already stored in EEPROM as the working configuration. You can also use a *default* configuration.

Select start-up configuration: Select which of the two stored EEPROM configurations is used as the working configuration upon powerup of the VIM2 module.

Begin run mode operation: Exit the *setup* mode and begin the inspection operation in the *run* mode.

Accessing the Main Menu

To begin to set up the run mode parameters and/or archive configurations, access the *Main* menu as described below.

Your Action	***Comments***
Pick the OK *icon on the* Setup *menu to access the* Main *menu.*	If the *Setup* menu is displayed, picking the *OK* icon accesses the *Main* menu (**Note:** If the *title banner* is displayed, pick any bright area on the monitor screen).

Setup Menu

OK Icon

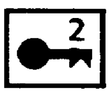
Accessing the Main Menu
(continued)

Below are brief descriptions of each of the *Main* menu icons:

Main Menu

Archive– Pick this icon to access the *Archive* menu. Use the *Archive* menu to store your configuration as one of two EEPROM configurations. Or load a stored configuration from EEPROM as the working configuration. Or load the default configuration as the working configuration.

Setup – Pick this icon to access the *Setup* menu. Use the *Setup* menu and subsequent menus to perform the majority of the VIM2 configuration tasks.



Powerup Configuration – Pick this icon to select which of the two stored EEPROM configurations will be used as the working configuration upon powerup. This icon, when picked, toggles between configuration 1 and configuration 2.

Standard/Learn – Pick this icon to select whether the VIM2 module will run in *standard* mode or in *learn* mode (in *learn* mode, the VIM2 module collects a number of different statistics for each vision tool and probe). This icon, when picked, toggles between the *standard* mode icon (the *factory*) and the *learn* mode icon (the *open book*).

Arm Mode – Pick this icon to select the *diagnostic* mode. This icon, when picked, cycles through the four different arm mode icons – *GO on reject*, freeze *on reject*, *halt on reject*, and *halt on each inspection.*

Reset Statistics– Pick this icon to set the stored values of all statistics collected to 0 (zero).

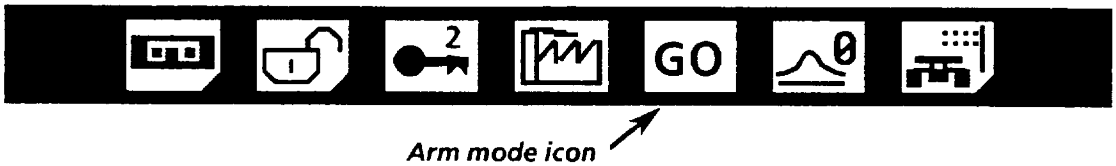
Run – Pick this icon to access the *Run* menu. Use the *Run* menu to exit setup mode and go to run mode, and/or access the *Archive* menu to store your configuration in EEPROM before exiting the setup mode.

Using the Arm Modes

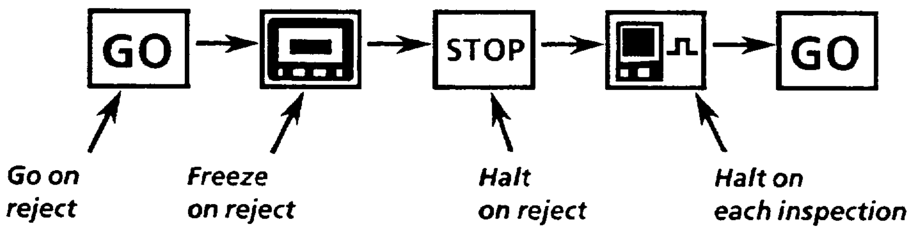
The VIM2 module has *arm* modes to aid in refining VIM2 configurations for inspection processes.

There are four arm modes available. Select the mode by picking the *Arm Mode* icon on the *Main* menu.

Main Menu



Your Action	*Comment*
Pick the Arm Mode *icon four times to toggle through the four arm modes.*	This icon, when picked, cycles through the four different arm mode icons – *GO on reject*, *freeze on reject*, *halt image on reject*, and *halt image on each inspection.*



When you exit the *setup* mode and begin *run* mode operation, the VIM2 module operates in the selected arm mode. In general, when you intend to use the VIM2 module in actual system operation, select either *GO on reject* mode, or *Freeze on reject* mode. When you intend to run a preliminary trial operation, select *any* of the four modes, as appropriate.

Here are descriptions of each of the four arm modes:

Go on Reject – This is the mode to select for normal system operation. The troubleshooting features (*halt image* or *stop*) are deactivated and system operation continues normally regardless of the *accept / reject* decision status.

Freeze on Reject – With this mode selected, when an inspection fails, the VIM2 module retains the onscreen image, but continues inspecting. The frozen image may or may not be the image that caused the failure. To resume the updating of the monitor image, depress the light pen tip once.

Using the Arm Modes
(continued)

Halt on reject – With this mode selected, whenever an inspection fails, the VIM2 module stops acquiring and processing images, and retains the image of the failed inspection. Using this mode, you can, after a failure, exit the *run* mode, and when you enter the *setup* mode, the *image acquisition* icon (on the *Image* menu) is set to *halt image*. All four threshold images from the last acquired image will be available for your review (see Chapter 8 for information on *halt image*). This allows you to, for example, check the results and statistics of each line gage and window you configured, and adjust the configuration if desired.

CAUTION: The VIM2 module stops inspecting when a reject occurs with either *halt on reject, or halt on each inspection* selected; steps must be taken in order to resume operation.

Note: The *Busy* output is held high, and the *Busy* LED is *On*, while the inspection is halted.

To resume the inspection operation, depress the light pen tip four times. The monitor display will cycle through the four threshold images (and windows), then resume inspections.

Halt on each inspection – With this mode selected, whenever an inspection takes place, the VIM2 module stops operation afterward, and retains the image of the inspection (whther or not it failed). Using this mode, you can, after each inspection, either resume operation as described above, or exit the *run* mode and enter the setup mode. When you enter the *setup* mode, the *image acquisition* icon (on the *Image* menu) is set to *halt image*. All four threshold images from the last acquired images will be available for your review (see Chapter 8 for information on *halt image*).

Note: To review the halted image in *setup* mode:

1. Unlock the VIM2 module (in standalone mode, pick the monitor screen with the light pen; with PLC master host, assert the unlock discrete bit; with serial master host, send the *Unlock* command).
2. Pick the monitor screen with the light pen to enter the *setup* mode (the *Main* menu is displayed). In *setup* mode, you can review inspection results, and statistics, and adjust your configuration as required.

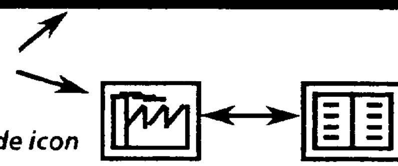
Using the Learn Mode

The VIM2 module has a *learn* mode to aid in refining your VIM2 configuration. While running in *learn* mode, the VIM2 collects statistics for each passing inspection (results for inspections for each tool/probe which fails an inspection are not included in the statistics for the respective tool/probe).

You select either the *learn* mode, or *standard* mode (no statistics collected) by picking the *Standard / Learn* icon on the *Main* menu. When picked, the *Standard / Learn* icon toggles between *standard* mode and *learn* mode.

Main Menu

Standard / Learn icon



Standard mode icon *Learn mode icon*

To use the *learn* mode, configure the VIM2 module as appropriate for your application, and select *learn* mode as part of your configuration before exiting to the *Run* mode. While running in *learn* mode, the VIM2 module collects statistics for the following vision tools:

- Brightness probe (see Chapter 8)
- All *enabled* line gages (see Chapters 9 and 11)
- All *enabled* windows (see Chapter 10)
- All *enabled* math tools (see Chapter 12)

You can review the statistics collected for a specific tool or probe by accessing the *Acceptance Range* menu for that tool or probe, and then picking the *current measured result* (see "Setting the Acceptance Range" in Chapters 8, 9, 10, 11 and/or 12). The statistics for the particular tool or probe are displayed on the right side of the screen, as shown below:

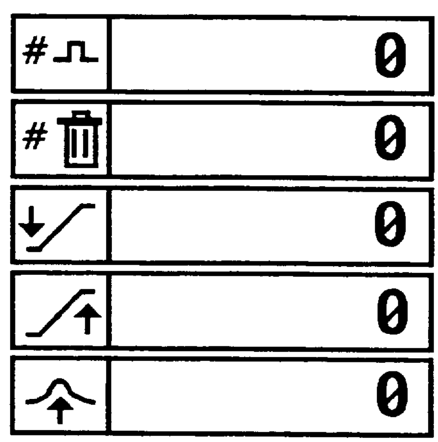
Total of triggers processed

Number of failures per tool/probe

Lowest result collected

Highest result collected

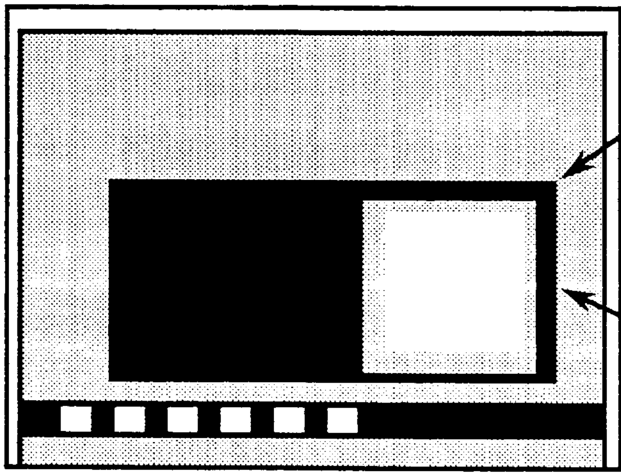
Average of all results collected



Using Learn Mode Statistics

In order to illustrate the use of the *learn* mode, an example is provided below.

Example: Using Learn Mode
Suppose you are going to measure the size of a workpiece feature by counting white pixels, using window 2 and threshold image 2, as shown below.



Before actually running the inspection, suppose you want to measure a random selection of 50 sample good parts to help define the window acceptance range. To do this, after staging and lighting the workstage, configuring threshold image 2, the brightness probe, window 2, and so on, you select the *learn* mode, then exit the *setup* mode, and enter the *run* mode. In *run* mode, with *learn* mode selected, the VIM2 module collects statistics as you inspect the 50 parts (**Note:** For information about accessing the statistics through block transfer, refer to Chapters 14 and/or 15).

To display the statistics for the window, do the following:

Your Action	***Comments***
With the VIM2 module in run *mode, pick the monitor screen* twice *to access the* Main *menu.*	The first pick takes the VIM2 module out of *run* mode, and displays the *title banner*. The second pick puts the VIM2 module in *setup* mode, and displays the *Main* menu. **Note:** If the VIM2 module has a PLC master host or serial port master host, the module must first be unlocked to access *setup* mode (see Chapter 14 or 15 for unlock information).

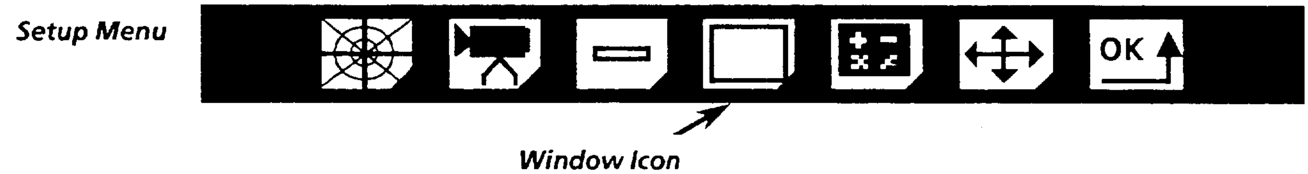
Main Menu

Setup Icon

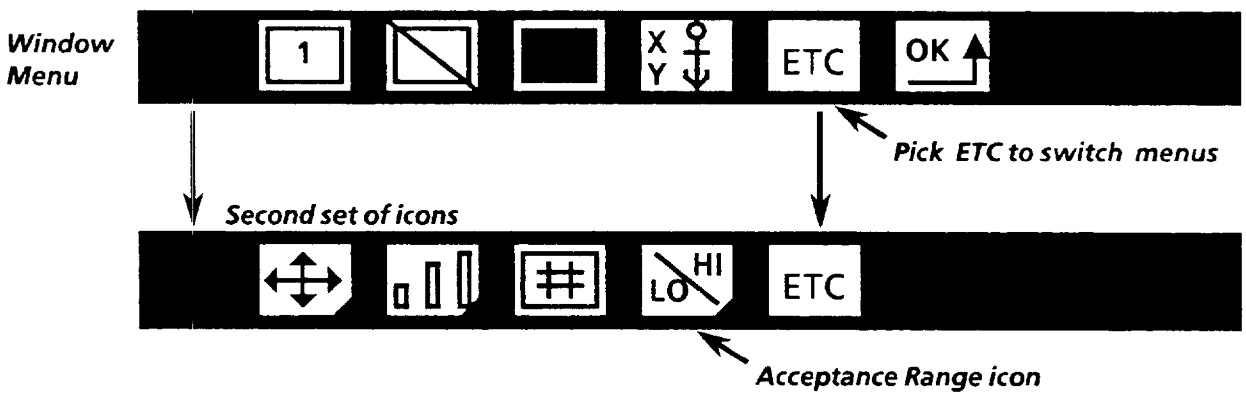
Using Learn Mode Statistics (continued)

Example: Using Learn Mode *(continued)*

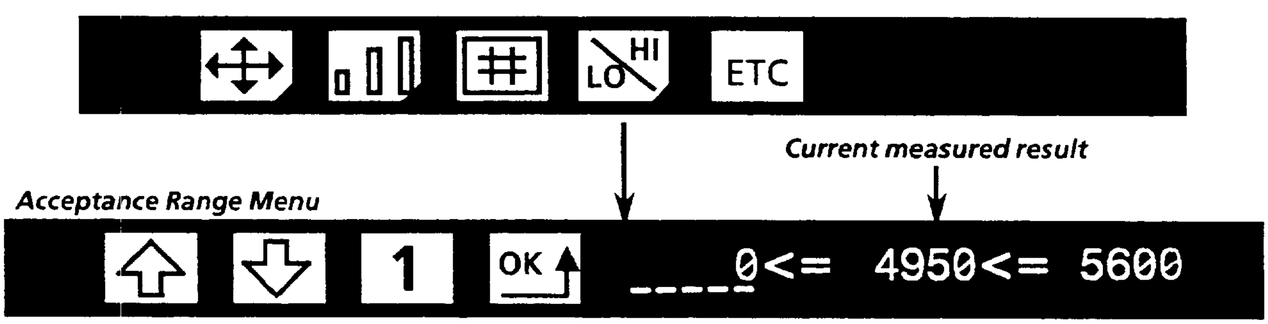
Your Action	*Comments*
Pick the Setup *menu icon to access the* Setup *menu.*	The *Setup* icon is the second icon from the left on the *Main* menu.



Pick the Window *menu icon to access the* Window *menu.*	
Pick the ETC *icon to access the second set of* Window *menu icons.*	When you pick the *ETC* icon, the second part of the *Window* menu is displayed. The *Acceptance Range* icon is found in the second set of icons.



Pick the Acceptance Range *icon on the* Window *menu.*	When you pick the *Acceptance Range* ("HI/LO") icon on the *Window* menu, the *Acceptance Range* menu appears:

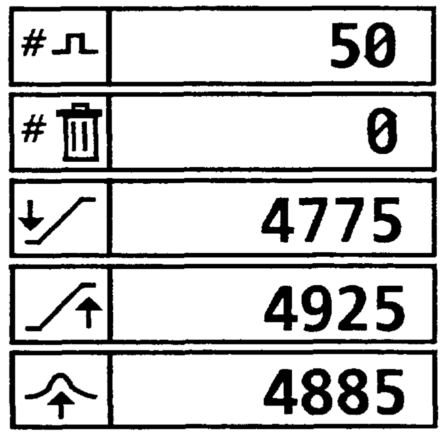


Pick the current measured result.	On the right side of the *Acceptance Range* menu are three numbers, separated by "< =" signs. The *middle* number is the *current measured result* of the window. When you pick this number, the statistics are displayed. To remove the statistics from the screen, pick the *middle* number again.

Using Learn Mode Statistics *(continued)*

Example: Using Learn Mode *(continued)*

Your Action	*Comments*
Observe and assess the collected statistics.	In this example, when the statistics are displayed, we see that 50 parts were inspected, and no parts failed inspection for this window. The lowest measured value of any of the 50 parts was 4775 (number of white pixels). The highest measured value was 4925. The average value was 4885.



As a result of using the *learn* mode and observing the statistics for the window, you now have guidelines for low (4775) and high (4925) range limits for the window.

Resetting the Statistics

The VIM2 module stores the collected statistics so long as you do not download a configuration from a host or from one of the two EEPROM configurations. If you change the configuration in either of these two ways, the VIM2 module resets all statistics values to 0 (zero). If you change the configuration only through means of the light pen while in *setup* mode, the statistics are not reset.

If you want to reset the statistics, do this while in *setup* mode by picking the *Reset Statistics* icon on the *Main* menu.

Your Action	*Comments*
Pick the Reset Statistics *icon on the* Main *menu.*	Picking the *Reset Statistics* icon immediately resets all statistics values to 0 (zero).

Main Menu

Reset Statistics icon

Archiving Configurations

The working configuration* of the VIM2 module is held in volatile RAM, and is lost if the power to the VIM2 module is disconnected for any reason. Using the *Archive* menu, you can store two different configurations in EEPROM on board the VIM2 module. The two EEPROM configurations (EEPROM configuration #1 and EEPROM configuration #2) are retained when power is disconnected from the VIM2; thus you can store two different configurations in EEPROM for safe keeping. And you can download either EEPROM configuration for use as the working configuration.

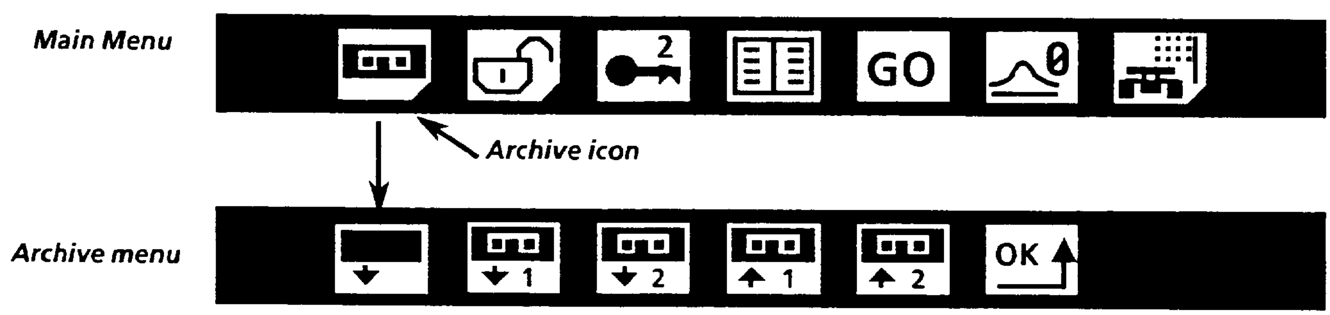
***Note:** The working configuration is the configuration used by the VIM2 module during operation. When you configure the VIM2 module with the light pen and monitor, you are configuring the RAM (working) configuration.

You can also transfer and store additional configurations if you are using a PLC or serial host. See Chapters 14 and 15 for more information.

Default Configuration – You can download the *default* configuration as the working configuration by picking the *Default Configuration* icon on the *Archive* menu. The *default* configuration is the initial configuration set within the VIM2 module when you power it up the first time.

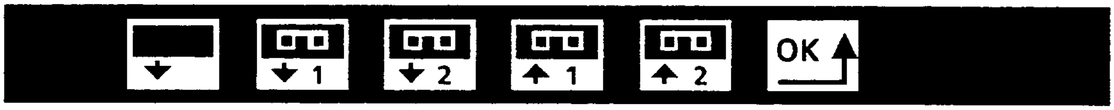
In order to review the *Archive* menu icons, access the *Archive* menu by picking the *Archive* icon on the *Main* menu.

Your Action	Comments
With the Main *menu displayed, pick the* Archive *icon to display the* Archive *menu.*	



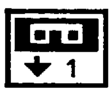
Archiving Configurations
(continued)

Archive menu

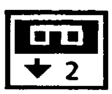


Here are brief descriptions of each of the *Archive* menu icons:

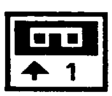
Load Default Configuration – Pick this icon to load the *default* configuration into RAM. The *default* configuration is the configuration used by the VIM2 module when you power it up the first time. In the default configuration, for example, among other things, all line gages and windows are disabled, all four threshold levels are set to a value of 128, windows and line gages are set in default positions in the center of the screen, and windows are set to rectangular. **Note:** When you download the default configuration into RAM, the current RAM (working) configuration is lost.



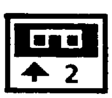
Download EEPROM #1 – Pick this icon to load the configuration stored as EEPROM configuration #1 into RAM for use as the working configuration. **Note:** When you download a configuration from EEPROM into RAM, the current RAM (working) configuration is lost.



Download EEPROM #2 – Pick this icon to load the configuration stored as EEPROM configuration #2 into RAM for use as the working configuration. **Note:** When you download a configuration from EEPROM into RAM, the current RAM (working) configuration is lost.



Save to EEPROM #1 – Pick this icon to save the working configuration for non-volatile storage to EEPROM configuration #1. **Note:** When you save a configuration to EEPROM #1, the previously saved EEPROM #1 configuration is lost.

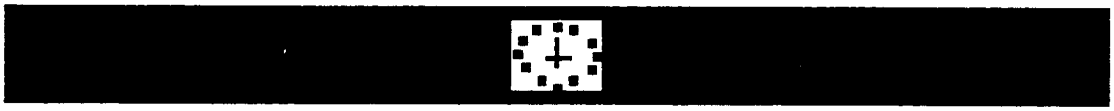


Save to EEPROM #2 – Pick this icon to save the working configuration for non-volatile storage to EEPROM configuration #2. **Note:** When you save a configuration to EEPROM #2, the previously saved EEPROM #2 configuration is lost.

It is suggested that you always back up your working configuration, either by saving it to one of the two EEPROM configurations, or by storing it through transfer to a host device (PLC or serial host).

Archiving Configurations *(continued)*

When you save a configuration into EEPROM, normally a few seconds elapse while the configuration is being stored. During this time, this *clock* icon appears:



CAUTION: The working configuration and both EEPROM configurations will be lost if you turn off power to the VIM2 module while the clock icon is displayed.

When the clock icon disappears, the save is complete. If you shut off power to the VIM2 module while saving a configuration to EEPROM, when power is reapplied, the *Config Fault* LED goes ON, indicating a configuration fault has occurred. The *Config Fault* LED remains ON until you use the light pen and monitor to enter the *setup* mode and then exit to *run* mode, or until you reconfigure the VIM2 module by sending a configuration from a host device (see Chapters 14 and 15).

If you use the light pen and monitor to enter the *setup* mode, after a configuration fault has occurred, you will notice that the working configuration, as well as both the EEPROM configurations, have been changed to the *default* configuration.

Selecting Startup Configuration

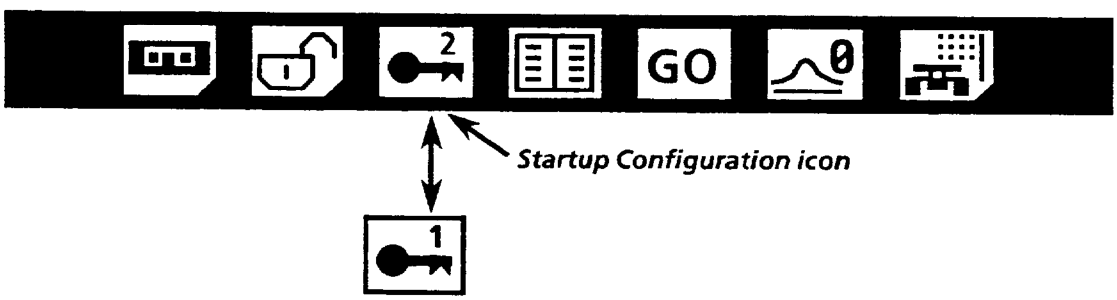
When the VIM2 module is powered down, its working configuration, stored in volatile RAM, is lost. When you power up the VIM2 module, it automatically loads one of the two available EEPROM configurations (see "Archiving Configurations," this chapter) into RAM for use as the working configuration.

To prevent loss of the working configuration, store the working (RAM) configuration as one of the two EEPROM configurations (see *Archiving Configurations* this chapter), then select that EEPROM configuration as the startup configuration. That way, should the VIM2 module ever lose power during operation, the VIM2 module, when powered back up, will automatically load the same working configuration into RAM as the one that was in RAM before the power loss.

You select either EEPROM configuration #1 or EEPROM configuration #2 as the *startup* configuration by picking the *Startup Configuration* icon on the *Main* menu.

Your Action	Comments
With the Main *menu displayed, pick the* Startup Configuration *icon to change the startup configuration.*	When picked, the *Startup Configuration* icon toggles between *1* (EEPROM Configuration #1) and *2* (EEPROM Configuration #2).

Main Menu



Exiting to Run Mode

Once you have configured the VIM2 module for a trial run or for the actual inspection operation, and you are ready to begin VIM2 module inspection operations, exit the *setup* mode, and enter the *run* mode. In the *run* mode, the VIM2 module awaits trigger signals, and, when triggered, performs inspections as configured.

⚠ **WARNING:** As soon as *run* mode is entered, the VIM2 module will begin responding to trigger signals. Any equipment cannected to the swingarm outputs may be energized by the resulting VIM2 output activity. **Do not** enter *run mode* unless assured that subsequent outputs will not result in damage to person or property.

To exit the *setup* mode, and enter the *run* mode, pick the *Run* icon on the *Main* menu.

Main Menu

Run icon

Run Menu – When you pick the *Run* icon, the following confirmation menu appears:

Archive icon *Run icon*

When the above icons appear, pick one of the three icons:

Your Action	*Comments*
Pick the Archive *icon, if desired, to access the* Archive *menu.*	With the *Archive* menu displayed, you can store the working configuration to EEPROM. Or, if desired, you can download one of the two EEPROM configurations, or the default configuration, for use as the working configuration.
Pick the OK *icon to return to the* Main *menu,* if desired.	If you change your mind about going to *run* mode, picking the *OK* icon returns the *Main* menu.
Pick the Run *icon to exit the* setup *mode and go to* run *mode.*	Picking the *Run* icon causes the VIM2 module to go to run mode (**Note:** If the VIM2 module is configured for master host operation, the VIM2 module goes to *standby* mode – title banner displayed – until locked by the master host).

Changing the Run Mode Display

While the VIM2 module is in *run* mode, you can change the monitor display. To do this, you depress the light pen tip – away from the monitor screen – to cycle through and select from six different display options:

(1) Show threshold image #1 and window #1

(2) Show threshold image #2 and window #2

(3) Show threshold image #3 and window #3

(4) Show threshold image #4 and window #4

(5) Show threshold image #1 and windows #1-4

(6) Show "live" analog image (direct – camera to monitor)

Using a Host Device to Change Display Mode – Note that you can also change the display mode using either a PLC or serial host device.

If the VIM2 module is integrated with a PLC, you can cycle through the display options by asserting the *light pen emulation* discrete bit as necessary (see Chapter 14).

Note: If both the light pen and the *light pen emulation* discrete bit are asserted simultaneously, only one is effective.

When using a host connected to the VIM2 module's RS-485 port, you can select a *specific* display mode by sending the *Change Display Mode* command (see Chapter 15).

Note: If the serial host's *Change Display Mode* command is received by the VIM2 module, and a simultaneous light pen pick or PLC assertion of the *light pen emulation* discrete bit occurs, only the display selection indicated by the *Change Display Mode* command is effective.

Chapter 14 Integration with the PLC

Chapter Objectives

This chapter provides guidelines for integrating the VIM2 module with your PLC system. Guidelines are included for:

- Using block transfer "read" and "write" commands within a PLC program for:
  - Receiving configurations from the VIM2 module.
  - Downloading configurations to the VIM2 module.
  - Receiving comprehensive inspection results and "learn mode" statistics from the VIM2 module.
- Using the "discrete bit" interface between the PLC and the VIM2 module for discrete error or result information, and for helping to define the desired block transfer "read" and "write" information.

Integrating the VIM2 Module with the PLC

The VIM2 module may be integrated with your process to provide feedback for process management and closed-loop process control. This integration can be done through interface with the discrete I/O of the VIM2 swingarm terminal, and /or through communication with a PLC.

The VIM2 module has a discrete I/O interface which you can directly connect to through the swingarm terminals (discussed in Chapter 5). Discrete outputs include *Decision* (inspection accept/reject) and *Busy*. There is also a strobe output for inspections with strobe lighting. For triggering inspections with an external device, there is a *Trigger* input. You can use the discrete swingarm I/O with or without the use of a host.

You can also transfer discrete information through direct discrete bit interface with the PLC. Discrete bit communications can transfer *accept/reject* decision and *error* condition signals; however, no measurement data is communicated.

For more complete information communication, you can use block transfer "reads" and "writes" to transfer information between the VIM2 module and the PLC.

Defining Your Interface Requirements

Selecting the best approach for you VIM2 module/process interface (discrete communication or block transfer) depends upon your process and its feedback requirements. First you must determine if you need a single (*discrete*) *accept/reject* decision per inspection, or if you need more specific (*block transfer*) information, such as a list of tool-specific accept/reject decisions, a list of tool-specific numerical measurements, etc. These choices are discussed below:

Using Discrete Bit Communication

The VIM2 module communicates a workpiece *accept/reject* decision through the discrete *Decision* output. You access the decision output through either or both the PLC controller (discrete bits) or through the swingarm *Decision* terminals.

Note: Other information, also transmitted through PLC controller discrete bits, are discussed later in this chapter.

The *Decision* output signals a reject if any of the acceptance range tests for the line gages, windows, math tools or brightness probe fail.

Discrete *decision* information is ideal if you only need to identify unacceptable units from production, and do not need information regarding the specific cause of the rejection. This is practical for inspection of workpieces simply for completeness and/or correctness of assembly, and for elimination of any unacceptable part from production, regardless of the specific part flaw.

Using Block Transfers

You can use block transfers to collect specific inspection result values or statistics from the VIM2 module, and you can find out which specific inspection tool(s) failed per inspection. This can provide a more comprehensive information profile for your inspection.

You can also use block transfers to collect and store the configuration data from the VIM2 module, or to download ("write") a stored configuration from the PLC to the VIM2 module.

The block transfer approach requires appropriate use of PLC ladder logic and block transfer programming, as well as an understanding of binary, binary coded decimal (BCD), and hexadecimal numbering.

Note: The transfer of blocks requires more time than discrete bit communication, and may add to the inspection cycle time in high-speed applications. This is a consideration mainly for high-speed applications (over 5 parts/second).

Defining Your Interface Requirements *(continued)*

For example, specific measurement feedback you can collect from the VIM2 module through a "results" block transfer include discrete bit accept/reject results for these tools:

- Brightness probe
- Windows 1-4
- X and Y float gages
- Line gages 1 through 22
- Math tools 1-12

The results blocks also provide measurements such as:

- Brightness probe gray level, from 0 to 255
- Window area measurements, from 0 to 61, 696 pixels
- X & Y gage measurements, from 0 to 255
- Line gage measurements, from 0 to 255
- Math tool results, from 0 to 65,535

The commands required for the collection of the above data by a PLC controller are discussed in this chapter (see Chapter 15 for similar information regarding a host connected through the RS-485 serial port). The *management* and *processing* of such data, however, is beyond the scope of this manual. Many software, hardware, and communications products are available for this purpose. Contact your local Allen-Bradley PLC supplier for information on data management and information processing options for the PLC controller.

Discrete Bit Communications

This section describes how to transfer discrete data through discrete bit transfer with the PLC.

Eight discrete bit lines are used for communicating discrete bit information between the PLC and the VIM2 module. You can use these data lines for transferring information to the VIM2 module (*discrete bit outputs*) or for collecting information from the VIM module (*discrete bit inputs*). The *discrete bit outputs* are used by the PLC for controlling communications and operation of the VIM2 module. The *discrete bit inputs* are used for communicating discrete results and module status information from the VIM2 module to the PLC.

Discrete Bit Communications (continued)

Addressing the Discrete Bits

Tables 14.A and 14.B list the discrete bit assignments for communication between the PLC and the VIM2 module. Figures 14.1 and 14.2 illustrate addressing discrete bits to include rack and module/group numbers.

Note: The bit addressing shown in this chapter is based on "two-slot addressing." For other addressing schemes (one-slot and half-slot), refer to Appendix F.

Table 14.A
VIM2 Module Discrete Bit Inputs to the PLC

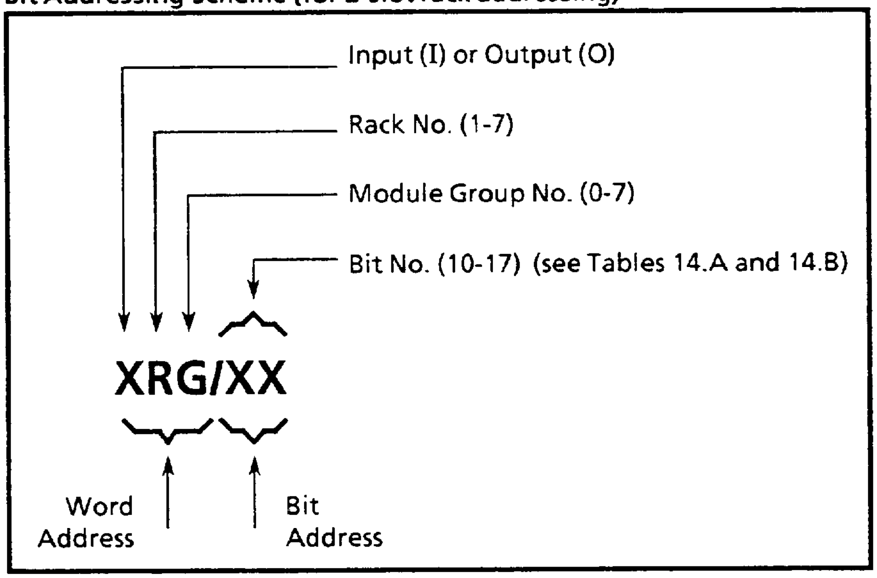
INPUT BIT ADDRESS	FUNCTION
10	**Module Fault** (0 = Running OK, 1 = Fault)
11	**Configuration Fault** (0 = Configuration OK, 1 = EEPROM/CONFIG. invalid)
12	**Busy** (0 = Ready, 1 = Busy)
13	**Decision** (0 = Accept, 1 = Reject)
14	**Probe Error** (0 = OK, Normal Operation, 1 = Error, Probe Out of Range)
15	**X/Y Float Error** (0 = OK, Normal Operation, 1 = Error, Out of Range)
16	Reserved
17	**Configuration Busy** (0 = Ready, 1 = Configuration Download in progress)

Table 14.B
PLC Discrete Bit Outputs to the VIM2 Module

OUTPUT BIT ADDRESS	FUNCTION
10	**Unlock** (0 = Lock the Module/Disable Progrmg., 1 = Unlock/Enable Progrmg.)
11	**Last Block** (0 = Not last block, 1 = Last Block)
12	**Select Configuration** (0 = Configuration 1, 1 = Configuration 2)
13	**Select Block Type** (0 = Configuration/Results, 1 = Statistics)
14	**Light Pen Emulation** (0 = no assertion, 1 = assertion)
15	**Trigger** (0 = Stand By, 1 = Initiate an Inspection Cycle)
16	**Binary/BCD Results** (0 = Standard Binary Number Format, 1 = BCD Format)
17	**Configuration Storage** (0 = RAM Storage, 1 = EEPROM Storage)

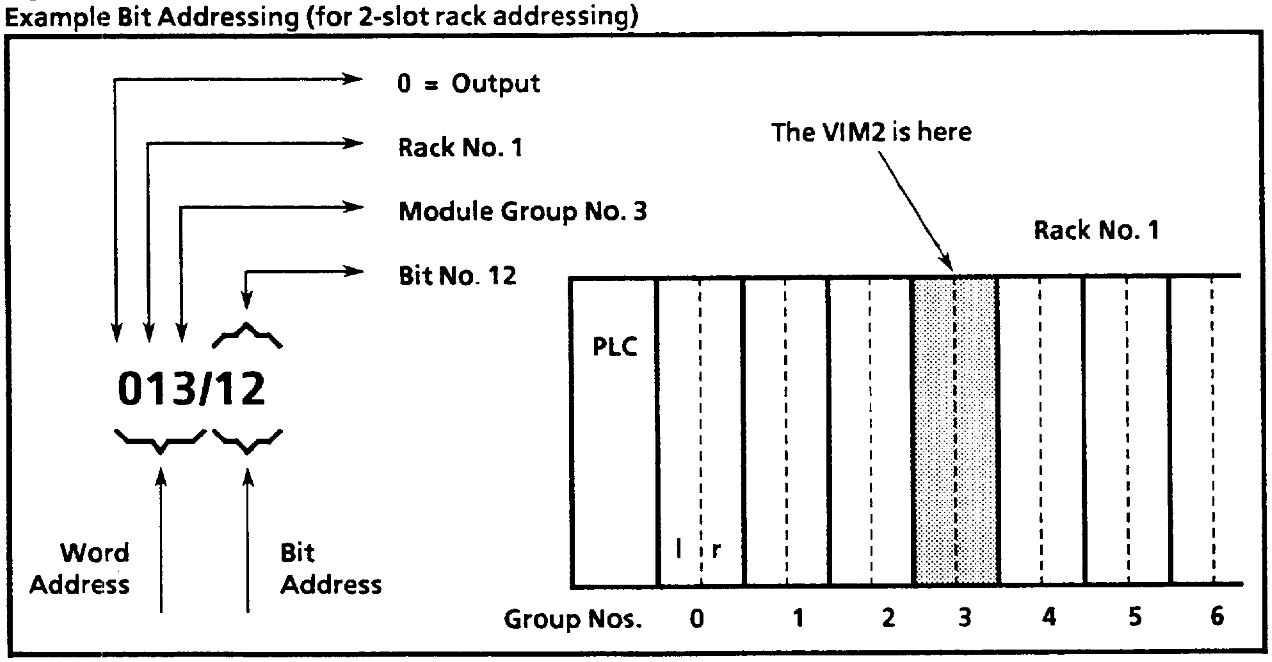
Discrete Bit Communications
(continued)

Figure 14.1
Bit Addressing Scheme (for 2-slot rack addressing)



The PLC controller addresses and controls I/O modules through ladder logic programming. The ladder logic rungs must be programmed to read inputs and write to outputs. These read and write operations are controlled through instruction addresses programmed on the ladder rungs. The address assignments typically appear as shown in Figure 14.1 and Figure 14.2.

Figure 14.2
Example Bit Addressing (for 2-slot rack addressing)



Discrete Bit Communications *(continued)*

Discrete Bit Inputs to the PLC Controller
There are eight discrete input bits available through the PLC controller. These include (see Table 14.A):

- **Decision** – This bit indicates the result status of the inspection, where 0 = Accept, and 1 = Reject. For inspection results, read this bit immediately after the *Busy* bit goes from "1" to "0." When this bit = 1, the *Decision* LED goes *On*. When this bit status goes to "1," it remains at "1" until an inspection "accept" decision is made by the VIM2 module.

- **Busy** – This bit indicates the VIM2 module inspection status, where 0 = Ready (for a trigger), and 1 = Busy (conducting an inspection). This bit is set just after image acquisition has begun. Monitor the status of this bit to determine when to read the *Decision* bit; read the *Decision* bit immediately after the *Busy* bit goes from "1" to "0." When this bit = 1, the *Busy* LED goes *On* (**Note:** This bit is also set when the VIM2 module is in setup mode).

- **Module Fault** –This bit indicates the hardware operating status of the VIM2 module, where 0 = OK, and 1 = Fault. When this bit = 1, no triggers are accepted. Monitor the status of this bit to determine whether the VIM2 module has a fault and is not functioning. When this bit = 1, the *CPU Fault* LED goes *On*. Also, the *Decision* and *Busy* outputs go *Off*.

- **Configuration Fault** – This bit indicates the VIM2 module configuration status, where 0 = OK, and 1 = Fault. A fault means that either of the two configurations in the VIM2 EEPROM is not valid, or that the downloaded configuration data is not valid, or that the start-up configuration number is not valid. A loss of power during a configuration download to the VIM2 module will also cause a configuration fault. If the VIM2 module indicates a configuration fault after a configuration block transfer, it aborts the transfer attempt, and reloads the transfer buffer with the previous working RAM configuration. To clear this bit, configure the VIM2 module through the light pen interface, or by downloading a valid configuration from the host. When this bit = 1, the *Config. Fault* LED goes *On*.

- **Probe Error** – This bit indicates the status of the brightness probe for an inspection, where 0 = reading within range, and 1 = out of range. If this bit = 1, the *Decision* bit also = 1 (reject), and no other vision tools are evaluated. When this bit = 1, the *Acq. Error* LED goes *On*. Monitor the status of this bit to determine when the workstage lighting brightness is out of acceptable range.

Discrete Bit Communications
(continued)

- **X/Y Float Gage Error** – This bit indicates the status of the X and Y float gage readings for an inspection, where 0 = both gages are within range, and 1 = either gage is out of range for this inspection. If this bit = 1, the *Decision* bit also = 1 (reject). When this bit = 1, the *Acq. Error* LED goes *On*. Monitor the status of this bit to determine when the workpiece positioning is out of acceptable range.

- **Configuration Busy** – This bit indicates the status of the configuration download process, where 0 = Not Busy, and 1 = Busy (the VIM2 is currently receiving a configuration block from the host). This bit is also set while the VIM2 module processes a block transfer command (see "Command Block" this chapter). While this bit = 1, the VIM2 cannot perform inspections, or receive another configuration block. Monitor the status of this bit to determine when to resume inspection triggers, or when to send the next configuration block.

Discrete Bit Outputs to the VIM2 Module

There are eight discrete output bits available through the PLC controller. These include (see Table 14.B):

- **Unlock** – Assert this bit to unlock the VIM2 module in order to enable manual (or block transfer) configuration of the VIM2 module. Otherwise set this bit to 0 to prevent any attempt at configuring the VIM2. This bit is active only if the PLC is the selected master host.

- **Last Block** – Assert this bit just before you send the last block of a configuration block transfer to the VIM2 module. Otherwise *set this bit to 0.* When this bit is asserted, the VIM2 module performs an internal configuration validation, then either accepts the configuration, or signals a *configuration fault* (see *Configuration Fault* bit description). If you do not set this bit to 1 after a block transfer of a configuration, the transferred data will not be loaded into the VIM2 module.

- **Select Configuration** – Use this bit to select the configuration number when you are transferring the configuration to or from a VIM2 module EEPROM configuration through a block transfer read or write. Also use this bit when you are changing the startup configuration through a command block transfer. Assert this bit to choose Configuration #2; otherwise, set this bit to 0 to choose Configuration #1.

Discrete Bit Communications *(continued)*

- **Select Block Type** – Use this bit to select the type of block (configure or results vs. statistics) you wish to read from the VIM2 module just before you use a block transfer "read" command. Set this bit to 1 to select statistics; otherwise, set this bit to 0. **Note:** This bit is only effective if the VIM2 module is running in "learn mode."

- **Light Pen Emulation** – Assert this bit to change the monitor display when the VIM2 module is in the *run* mode. Asserting this bit has the same effect as depressing the light pen tip (that is, the VIM2 monitor cycles through the six display modes – see Chapter 6 or 13 for more information). **Note:** To repeat light pen emulation, you must set this bit to 1, then to 0, then to 1, then to 0, etc.

- **Trigger** – Assert this bit to trigger an inspection cycle. Asserting this bit has the same effect as changing the trigger input status (on the swingarm) to high; that is, it will initiate an inspection (provided the VIM2 module is not busy or is not being configured). To repeat the trigger, you must set this bit to 1, then to 0, then to 1, then to 0, etc. **Note:** The SWG TRIGGER LED does not respond to the discrete bit trigger signal. Also, this bit is only effective if the PLC is the selected master host, and if host triggering is selected (see Chapter 7).

- **Binary/BCD Results** – Use this bit when reading results blocks to select the type of numerical format for the results data. Set this bit to 1 to select binary-coded decimal (BCD); otherwise, set this bit to 0 to select binary. **Note:** Statistics data is only available in binary format. This bit is ignored when reading statistics blocks.

- **Configuration Storage** – Use this bit when you send a configuration block transfer "read" or "write" to the VIM2 module. Set this bit with a block transfer write to indicate whether the configuration is stored only in VIM2 module RAM ("0"), or also in one of the two EEPROM configurations ("1"). Set this bit to 1 with a block transfer read to indicate the configuration is read from one of the two EEPROM configurations (setting this bit to "0" to select RAM is not a valid option for block transfer reads). **Note:** If you set this bit to "1," you must also indicate which of the two EEPROM configurations you are writing to or reading from by setting or clearing the *Select Configuration* bit.

  Use this bit also when you send the command block transfer to load an EEPROM configuration into RAM (the working configuration). Set this bit to "1" to indicate the downloaded EEPROM configuration is the new designated start-up configuration. Otherwise set to "0."

Bit Manipulation

There are several ways to control the operation of the module using programming tools in the PLC controller. One way is to manipulate bits in the discrete bit communications set, and to control triggering through ladder logic programming.

Many PLC controllers have a Bit Manipulation Menu which can be used to force the VIM2 module control bits – listed in Table 14.B as the output bits. This example uses a PLC 2/05 and T3 programming terminal. Other PLC controllers may be slightly different.

To access the Bit Manipulation Menu, press SEARCH 53 on the T3 terminal. The Bit Manipulation Menu will be displayed as shown in Figure 14.3. The bits are listed in octal format.

Figure 14.3
PLC Bit Manipulation Menu Used to Force Control Bits

```
                     BIT MANIPULATION


                   WORD ADDRESS:  0011


BIT NO : 17 16 15 14 13 12 11 10 07 06 05 04 03 02 01 00
STATUS :  0  0  1  0  0  0  0  0  1  0  0  1  1  1  0  1
FORCE  :





REMOTE PROG MODE
```

In this example the 0011 is used as the word address. The address you use will be determined by the location of the VIM2 module in your I/O chassis.

Bit Manipulation Example 1:

Locking and Unlocking the VIM2 Module
This bit is only effective if the PLC is the selected master host. Move the cursor to the *Unlock* bit (bit 10 in Table 14.B). Press "1" to unlock the module and enable light pen programming. The video monitor will display the title banner. Press "0" to lock the module and go to run mode. The title banner will disappear from the video monitor.

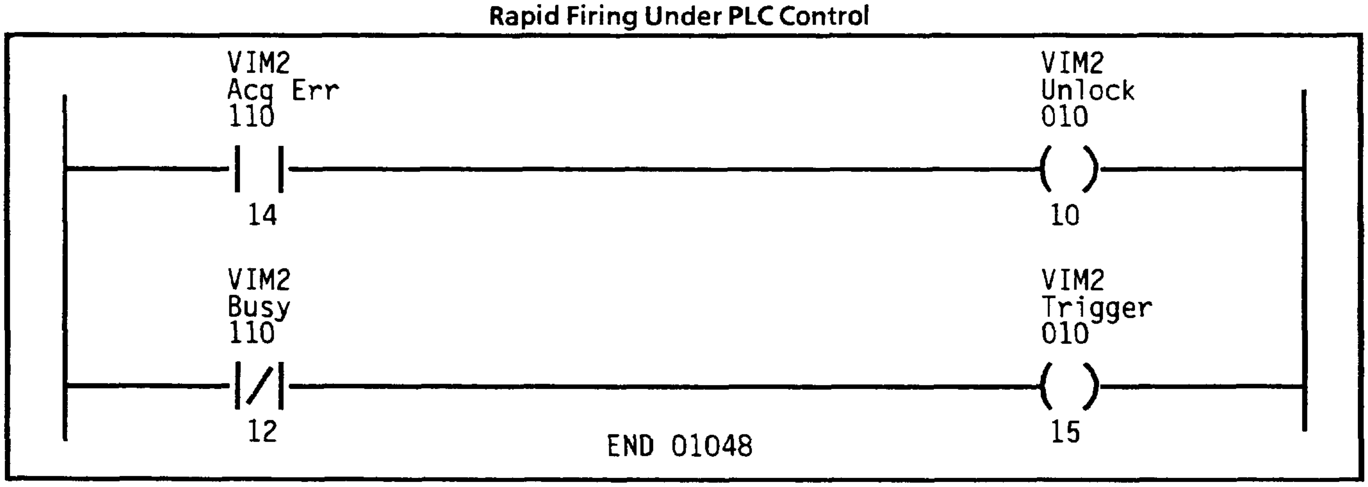
The Unlock output bit is used to control access to the VIM2 configuration data. When the operator wants to enter the setup mode, the PLC controller must set this bit high. When the operator unlocks the VIM2 module with the light pen the set-up session is started.

Note: As long as the *Unlock* bit is set low (0), the module's configuration data cannot be modified.

Bit Manipulation Example 2:

Rapid PLC System Triggering of the VIM2 Module
A simple two-rung program can be used to rapidly trigger the VIM2 system inspection cycle (see Figure 14.4). This program is only effective if the VIM2 module is set so the PLC is the master host, and so the run mode trigger source is HOST TRIG (see Chapter 7). In this program, the *Busy* bit is used to drive the *Trigger* bit. This forms an oscillator which rapidly and continuously triggers the module.

Figure 14.4
Rapid Firing Under PLC Control



The oscillation is stopped when the module is unlocked by rung 1 due to an ACQ ERROR (brightness probe out-of-range). The low range limit for the brightness probe should be set to a number greater than 10 to enable the shut-off (see Chapter 8).

To test this while the program is running, disconnect the camera cable; the ACQ ERROR light comes on due to brightness out-of-range and inspections stop. To restart the program, force the *Unlock* bit back to zero.

Block Transfers

The PLC can exchange data with VIM2 module using block transfer commands – block transfer "read," and block transfer "write." Block transfers are controlled from the PLC controller using ladder logic programming. There are four types of blocks which can be transferred – *results, statistics, configuration*, and *command.*

A block of data can contain up to 64, 16-bit words of data. You define the type of block, for the most part, by indicating the block length (number of words). Other requirements for the specifying different blocks are discussed individually for each block type in the following sections.

Tables indicating block transfer data assignments are provided in Appendices B, C, and D for configuration, results, and statistics data. The data in data blocks may be assigned in either bit or word increments. Some information, such as an accept/reject decision, requires only one bit, while other information, such as a measured value for a tool, requires a byte of information (8 bits) or a full word.

Configuration Blocks

Configuration blocks contain information regarding the VIM2 module configuration. This information can be read from the VIM2 module, or downloaded to the VIM2 module (see Appendix B for the contents of the configuration blocks).

The VIM2 module can store two configurations in its EEPROM memory, and another configuration (the working configuration) in its volatile RAM memory. You select which configuration you read or write through the use of discrete bits. There are four blocks per configuration; you select which block of a configuration you are writing or reading by specifying the block length (that is, number of words – 30, 62, 63, or 64):

Configuration Block One - 30 Words
- Trigger status
- Strobe status
- Run-time image freeze status
- Probe position and acceptance range data
- Window configuration data for windows 1-4

Configuration Block Two - 62 Words
- X and Y Float line gage configuration data
- Configuration data for line gages 1-10

Configuration Block Three - 63 Words
- Configuration data for line gages 11-22

Configuration Block Four- 64 Words
- Configuration data for math tools 1-12

Configuration Blocks (continued)

Reading Configuration Data

To upload configuration data from the VIM2 module to the PLC, use a block transfer "read" command. You can upload any one or all of the configuration blocks, as required. You can only read one block at a time.

You cannot read configuration blocks while the VIM2 module is in *setup* mode. If the PLC is master host, it can read configuration data during either *standby* (unlocked, and not in *setup* mode) or *run* mode. If the PLC is not the master host, it can read configuration data only in *standby* mode.

You can read configuration data from either of the VIM2 module EEPROM configurations. In doing so, for each configuration block read, your program would include these steps:

1. If you are attempting to upload a configuration while the VIM2 module is running in the *learn* mode, clear the *Select Block Type* discrete bit to specify *configuration* blocks (and not *statistics*).

2. Before sending the block transfer command, set the *Configuration Storage* discrete bit to "1" to select EEPROM configuration (setting this bit to "0" to select RAM is not a valid option for block transfer reads).

3. Set or clear the *Select Configuration* discrete bit to specify EEPROM configuration 1 or 2.

4. Select the configuration block you are reading by block length (number of words – 30, 62, 63, or 64).

5. Monitor the *Busy* discrete bit, and wait until *Busy* discrete bit is clear, indicating the VIM2 module status is not busy.

6. Monitor the *Config Busy* discrete bit, and wait until *Config Busy* discrete bit is clear.

7. Send the block transfer read command.

8. Monitor the *Config Busy* bit. When it is clear, the block transfer is complete.

Writing Configuration Data

To download configuration data from the PLC to the VIM2 module, use a block transfer "write" command. You can download any one or all of the configuration blocks, as required. You can only write one block at a time. Writing configuration block is allowed only while the VIM2 module is in *standby* mode, and if the PLC is the selected master host.

You can either write only to the VIM2 module RAM configuration, or you can write to EEPROM configuration 1 or 2 in addition to writing to the RAM configuration. In doing so, for each configuration block write, your program must:

1. Check the *Unlock* discrete bit; set if it is not set.
2. Before sending the block transfer command, set or clear the *Configuration Storage* discrete bit to select either RAM only or RAM and also EEPROM configuration.

   Note: Configuration data stored into EEPROM takes about 5 seconds. When configuration data is stored in RAM only there is no such time requirement.
3. If you select EEPROM in step 2, also set or clear the *Select Configuration* discrete bit to specify EEPROM 1 or 2.
4. Select the configuration block you are writing by block length (number of words – 30, 62, 63, or 64).
5. Clear the *Last Block* discrete bit if you are not sending the last block. Otherwise, set the *Last Block* discrete bit.
6. Monitor the *Busy* and *Configuration Busy* bit status; wait until these bits are clear to send the block transfer write command.
7. Send the block transfer write command.
8. Monitor the *Config Busy* bit. When it is clear, the block transfer is complete.

Results Blocks

The results blocks provide specific results data for each vision tool (see Appendix C for a complete listing of the contents of the results blocks). The results blocks can only be read from the VIM2 module; you cannot write results blocks to the VIM2 module. There are two results blocks – Results Block 1 (59 words, all results except math tools) and Results Block 2 (13 words, math tools only). The PLC can read results only while the VIM2 module is in *run* mode.

Block Length
You select which results block you are reading by specifying the block length (that is, number of words). You specify 13 to select Results Block Two.

To select Results Block One, specify any number of words from 1 to 59 (except 13 or 30).* The PLC will read that amount of consecutive words in Results Block One, starting with the first word.

***Note:** If you specify 30, the PLC reads the configuration block of that length; if you specify 13, the PLC reads Results Block Two (only if a math tool is enabled; otherwise the PLC reads the first 13 words of Result Block One). You cannot read a partial portion of Results Block Two.

The results are provided in either binary, or binary-coded decimal (selectable through the *Binary/BCD Results* bit).

Results Block One - 1-59 Words
(words 1-4, discrete results):
- The brightness probe (in range/out of range)
- Windows 1 through 4 (accept/reject)
- X and Y Float line gages (in range/out of range)
- Line gages 1 through 22 (accept/reject)

(words 6-58, measured values):
- Brightness probe luminance level
- Pixel counts for windows 1 through 4
- Upper and lower function results, X and Y float gages
- Upper and lower function results, line gages 1 through 22

Results Block Two - 13 Words
(word 1, discrete results):
- Math tool (1-12) results (accept/reject)

(words 2-13, actual values):
- Math tool (1-12) results values

Reading Results Data
To upload results data from the VIM2 module to the PLC, use a block transfer "read" command. You can only read one block or partial block at a time.

Results Blocks *(continued)*

In reading each results block, your program must:

1. Before sending the block transfer command, if you are attempting to upload results while the VIM2 module is operating in the "learn" mode, clear the *Select Block Type* discrete bit to specify results blocks (and not statistics).

2. Set or clear the *Binary/BCD Results* discrete bit to select binary or BCD results.

3. Select the block you are reading by block length.

4. Monitor the *Busy* discrete bit status; when it changes from 1 to 0, send the block transfer read command.

5. Monitor the *Config Busy* bit. When it is clear, the block transfer is complete.

Statistics Blocks

The statistics blocks provide data for each vision tool collected while the VIM2 module is running in the "learn" mode. The statistics blocks can be only be read from the VIM2 module, and cannot be written to the VIM2 module (see Appendix D for a complete listing of the contents of the statistics blocks). The PLC can read statistics only while the VIM2 module is running in *learn* mode. There are four statistics blocks; you select which statistics block you are reading by specifying the block length (that is, number of words – 55, 12, 62, or 60):

Statistics Block One - 55 Words
- Number of triggers
- Number of triggers failed
- Number of window failures, windows 1-4
- Number of X and Y float gage failures, upper/lower
- Number of line gage failures, 1-22, upper/lower

Statistics Block Two - 12 Words
- Number of math tool failures, 1-12

Statistics Block Three - 62 Words
(Statistics – minimum, maximum, average)
- Brightness Probe
- Windows 1-4
- X and Y float gages, upper/lower
- Line gages 1-14, upper/lower

Statistics Block Four - 60 Words
(Statistics – minimum, maximum, average)
- Line gages 15-22, upper/lower
- Math tools 1-12

Statistics Blocks (continued)

Reading Statistics Data

To upload statistics data from the VIM2 module to the PLC, use a block transfer "read" command. You can upload any one or all of the statistics blocks, as required. You can only read one block at a time. You can only read statistics while the VIM2 module is running in the "learn" mode. The statistics are provided in binary format only.

For each statistics block read, your program must:

1. Before sending the block transfer command, set the *Select Block Type* discrete bit to specify statistics blocks (and not configuration or results).
2. Select the statistics block you are reading by block length (number of words - 55, 12, 62, or 60).
3. Monitor the *Busy* discrete bit status; wait until the bit is clear, indicating the VIM2 module is not busy.
4. Send the block transfer read command.
5. Monitor the *Config Busy* bit. When it is clear, the block transfer is complete.

Command Block

The PLC can send a "command" block to the VIM2 module using a block transfer "write" command in order to change the working configuration. You select the new working configuration by setting or clearing the *Select Configuration* discrete bit before sending the command block to the VIM2 module. You can also use the command block to reset the VIM2 module statistics *only* (**Note:** The statistics are reset automatically if you use the command block transfer to change the configuration, or if you write a configuration to the VIM2 module). Table 14.C shows the command block.

Table 14.C
Command Block Transfer

Command Block Transfer			
WORD (16 Bits)	BIT (Decimal)	FUNCTION	VALUES
1	0 1-7 8 9-15	Configuration Change Reserved for Future Expansion Reset Statistics Reserved for Future Expansion	0 = No change, 1 = Change 0 0 = No change,* 1 = Reset Statistics 0
2-5		Reserved for Future Expansion	Zero

***Note:** The statistics are reset whenever bit 0 is set to 1, whether or not bit 8 is set to 0.

Writing the Command Block

To download the command block data from the PLC to the VIM2 module, use a block transfer "write" command. The VIM2 module must be in *standby* mode, and the PLC must be the selected master host, in order for the PLC to write the command block. In writing the command block, your program would include these steps:

1. Check the *Unlock* discrete bit, and set if not set to 1.

2. If a configuration change is requested in the command block, set or clear the *Select Configuration* discrete bit to select the new working configuration from either EEPROM configuration 1 or EEPROM configuration 2.

3. If a configuration change is requested in the command block, set or clear the *Configuration Storage* bit to indicate if the selected EEPROM configuration number is to be saved as the new start-up configuration number.

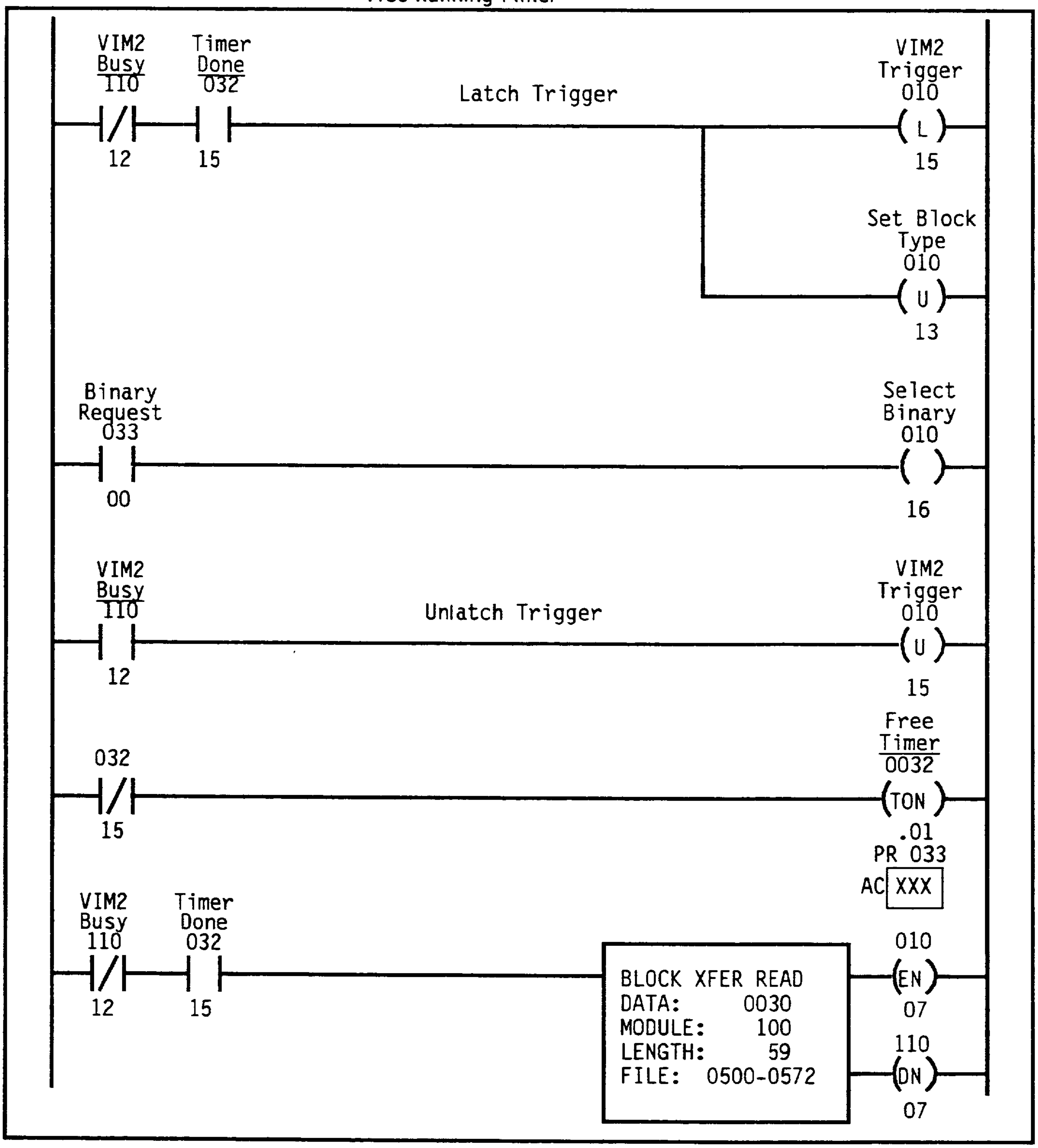
4. Select the command block you are writing by block length (number of words = 5).

5. Monitor the *Config Busy* and *Busy* bits; wait until these bits are clear to send the block transfer write command.

Block Transfer Example

An example of a results block transfer will serve as an example of block transfer programming. The PLC program in Figure 14.5 reads the list of measurement results for each inspection performed.

Figure 14.5
Free-Running Timer



Displaying the Results Blocks

To display the contents of the results blocks, place the blinking cursor on the block transfer instruction and press "DISPLAY 0." Figure 14.6 shows a typical results block with results displayed in binary form. Figure 14.7 displays the same results block in hexadecimal form.

Notice that the display in Figure 14.6 is updated with every inspection cycle.

Figure 14.6
Results Block Display in Binary Format

```
BINARY DATA MONITOR
 BLOCK XFER READ
 MODULE ADDR:  110
 FILE:  1000- 1034

POSITION    FILE DATA
   001  00000000 00011000
   002  00001010 10000000
   003  00000000 00000000
   004  00000000 00000000
   005  00000000 00000000
   006  00000000 00001001
   007  00000000 00000000
   008  00000000 00000001
   009  00001000 00001111
   010  00000000 00010000
   011  00000000 00000000
   012  00000000 00000000
   013  00000000 10111111
   014  00000000 00100101
   015  00000000 10111000

RUN/PROGRAM MODE
```

Results Block Format

The numerical measurements in the results block can be sent to the PLC controller in binary or binary coded decimal (BCD) format. The format is selected by discrete output Bit #16. If the output is set to one, the results will be formatted as four digit BCD numbers. If the output is set to zero, the measurements will be formatted as 16 bit binary numbers.

If you are not familiar with the following counting systems, refer to your PLC programming manual.

Results Block Format
(continued)

Figure 14.7
Results Block Display in Hexadecimal Format

```
          HEXADECIMAL DATA MONITOR
              BLOCK XFER READ
              MODULE ADDR:  110
              FILE:  1000- 1034

           POSITION    FILE DATA
              006         0009
              007         0000
              008         0001
              009         080F
              010         0010
              011         0000
              012         0000
              013         00BF
              014         0025
              015         00B8
              016         0013
              017         00AD
              018         0010
              019         002F
              020         1013

RUN/PROGRAM MODE
```

Block Transfer Numbering Systems

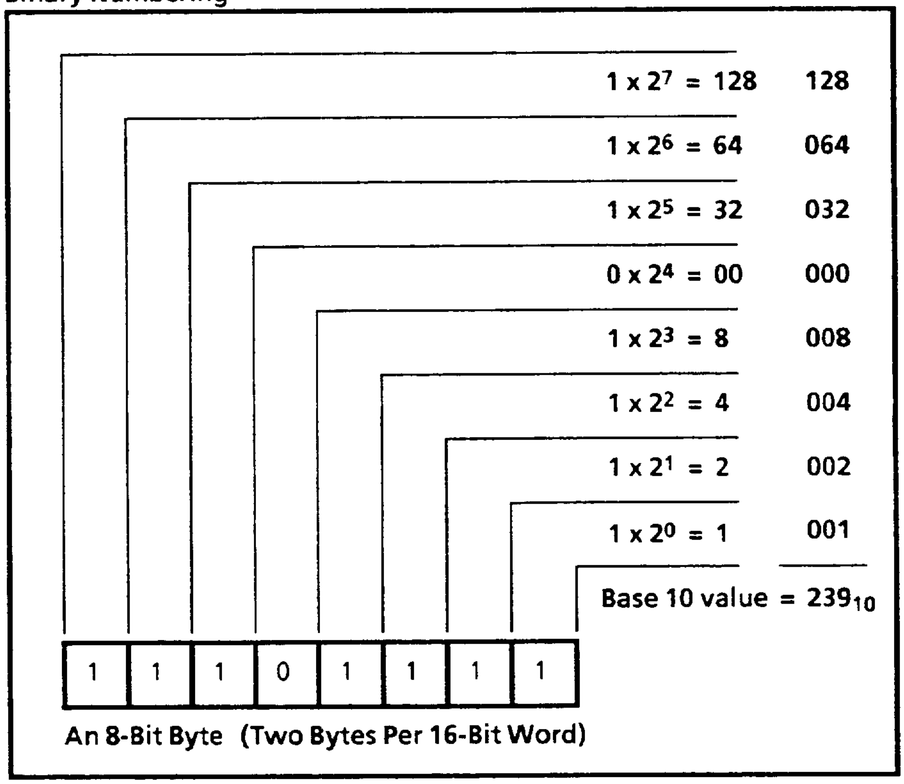
There are four numbering systems used with programmable controllers. They are:

- Binary
- Decimal
- Octal
- Hexadecimal

These numbering systems differ by the counting base used and the resultant differences in place values. The decimal, octal, and hexadecimal numbering systems are represented by binary bit sets at the PLC controller level and converted for display. This coding is referred to as binary-coded decimal (BCD), binary-coded octal (BCO), and hexadecimal.

Block Transfer Numbering Systems
(continued)

Figure 14.8
Binary Numbering



Binary Format

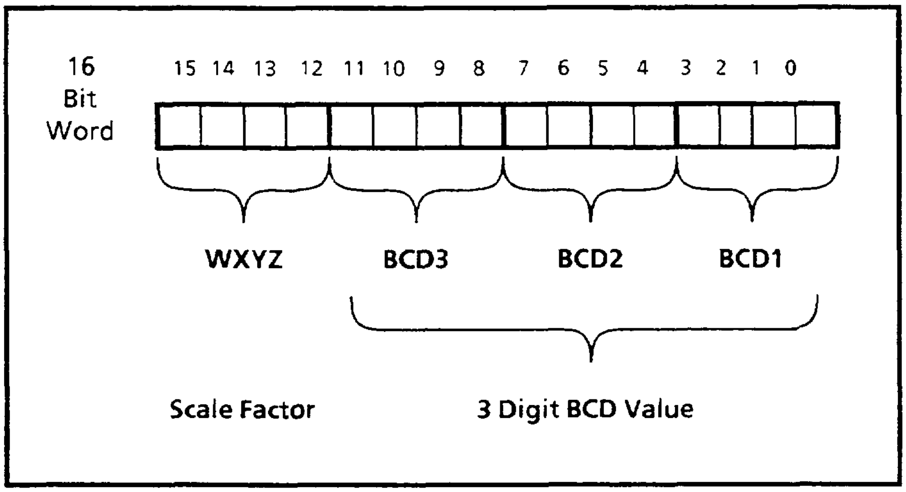
The binary numbering system uses a number set that consists of two digits: the numbers 0 and 1. All information in memory is stored as an arrangement of 0's and 1's. Each digit in a binary number has a certain place value expressed as a power of two. The decimal equivalent of a binary number is computed by multiplying each binary digit by its corresponding place value and adding these numbers together, as shown in Figure 14.8.

When PLC output bit 16 is set to 0, the results block data will be transferred in binary format (unsigned integer). Each block contains up to 64, 16-bit words. Each word contains one or more numbers.

Block Transfer Numbering Systems *(continued)*

The formats of the configuration, results, and statistics blocks are detailed in Appendices B, C, and D.

Figure 14.9
BCD Word Format



BCD Format

BCD words, as applied in the VIM2 system, have two parts: the scale factor and the value. The scale factor is used to allow a greater range of numbers than is possible with only three digits. The bits in the scale factor allow the transfer of numbers that are 10, and 100 times greater than the three digit value represented in the three-digit BCD value. The format of the BCD word is shown in Figure 14.9. The Scaling factors represented by the four scale factor bits (bits 12 through 15) are:

W is Not Used
X is Not Used
Y is set to 1 if the value is 1/100 the actual value
Z is set to 1 if the value is 1/10 the actual value

The remaining 12 bits are assigned in groups of four; one group for each digit. This is because it takes four binary number places to reach a decimal count of 9. The numbers displayed in the BCD mode are three digit unsigned decimal integers of 0 through 9.

The BCD bits are assigned as follows:

BCD3 is the high order digit (bits 8 through 11)
BCD2 is the middle digit (bits 4 through 7)
BCD1 is the low order digit (bits 0 through 3)

***Block Transfer Numbering Systems** (continued)*

You can use *X/Y Float Error* discrete bit in your PLC program to verify that all tool readings are reported in the results block. If both this bit is clear (0), then the results block will contain measurements from all enabled tools. If the *X/Y Float Error* bit is set, results of all tools with *shift compensation* enabled ("float" status selected) will be zero, and the corresponding tool status bits will be set to 1 to indicate failure.

(This page intentionally left blank)

Chapter 15 Using the RS-485 Port

Chapter Objectives

In this chapter we describe how to:

- Configure the VIM2 module for serial communication.
- Use ASCII or DF1 protocol commands to perform functions including uploading or downloading configurations, reading results or statistics data, and locking/unlocking the VIM2 module.

RS-485 Communications

Using the RS-485 port you can link computers which have an RS-485 port to the VIM2 module. You can also link these additional devices, through the use of an RS-232/RS-485 converter, to the VIM2 module's RS-485 port:

- Computers with RS-232 ports
- Operator Interfaces such as Allen-Bradley Industrial Terminals and Computers with serial ports.
- I/O modules such as the BASIC module (Catalog No. 1771-DB) or ASCII module (Catalog No. 1771-DA).

The RS-485 port on the VIM2 module is designed for half-duplex, point-to-point communication only.

Note: To communicate with the VIM2 module RS-485 port, you must also supply the communication driver for your host device. Contact your local Allen-Bradley representative for more information on RS-232/RS-485 converters and/or communication drivers.

ASCII and DF1 Protocols

There are two protocol options when you use the RS-485 communications port, **ASCII** and **DF1**. This chapter describes both of these options. First we describe the ASCII protocol and then the DF1 protocol.

Command Availability

The availability of each serial command depends on the current mode of the VIM2 module (*setup*, *standby*, or *run*), and upon whether or not the serial host is selected as the master host. The stipulations for mode and host status are listed in the descriptions for each command.

What Functions Can Be Performed Over The RS-485 Interface?

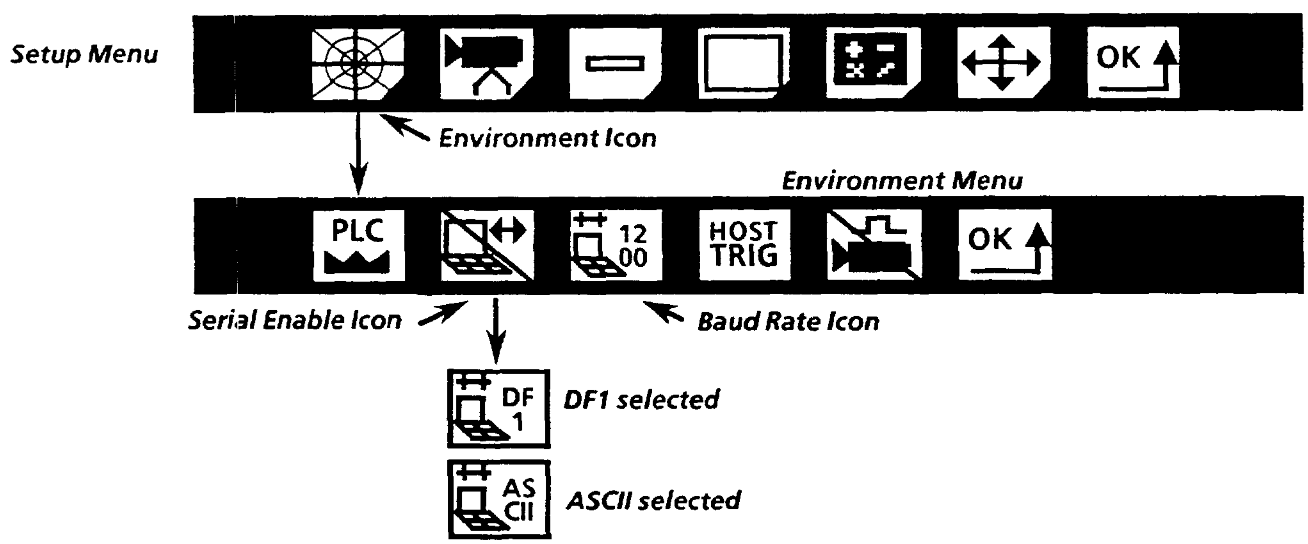
A host device can perform the following functions through the RS-485 port:

- Obtain VIM2 module results information.
- Obtain VIM2 module statistics information.
- Upload or download VIM2 module configurations.
- Transfer configurations between the VIM2 module's RAM configuration and two EEPROM configurations.
- Change run mode display.
- Trigger an inspection.
- Lock or unlock the VIM2 module to control access to the setup mode.

VIM2 Module Configuration Instructions

If you are using the RS-485 port, you must configure the VIM2 module, using the light pen/monitor interface, for serial communications. Begin by enabling and configuring the serial communications port:

Your Action	Comments
Pick the Environment *icon on the* Setup *menu to access the* Environment *menu.*	If the *Main* menu is displayed, first access the *Setup* menu by picking the *Setup* icon (the "open padlock"). Then, with the *Setup* menu displayed, pick the *Environment* icon.



Your Action	Comments
Pick the Serial Enable *icon to select* ASCII *or* DF1 *protocol.*	
Pick the Baud Rate *icon as necessary to set the baud rate for communication.*	Picking this icon repeatedly cycles through the baud rate options – 1200, 2400, 4800, or 9600.

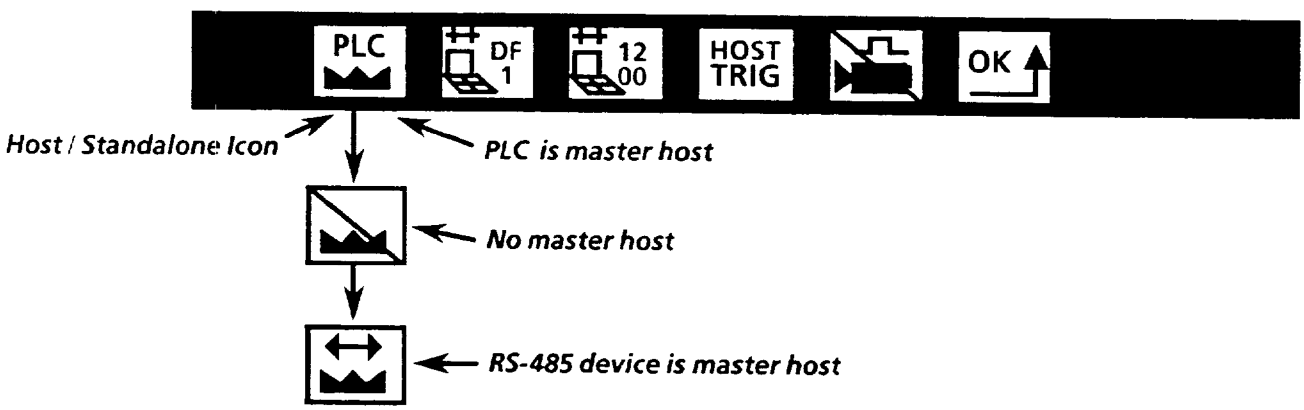
VIM2 Configuration Instructions *(continued)*

Note: When you select RS-485 communications, the data format is fixed as follows:

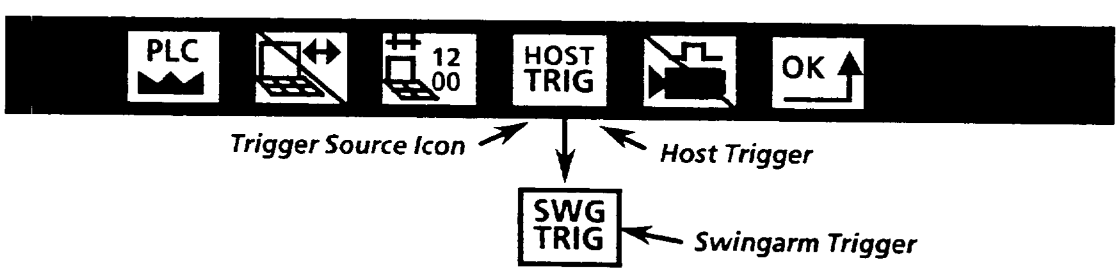
- 8 Data Bits
- 1 Stop Bit
- No Parity

Next, select the *master host*, or select *no master host*, according to whether a PLC, or a serial device, will act as the master host, or if no device will be a master host. The master host is exclusively permitted to perform certain functions, such as downloading configurations or triggering inspections. A subset of the master host functions can be performed by a PLC or serial device which is not the master host, such as the collection of configuration or results data (**Note:** The specific host and mode stipulations which apply when issuing a command from a serial host are included in this chapter, in the description for each command).

Your Action	*Comments*
Pick the Host / Standalone *icon as necessary to select the appropriate mode.*	Picking this icon selects the *master host* (or selects *no master host*). **Note:** The *Serial Host* icon is not available unless you first *enable* the RS-485 port using the *Serial Enable* icon (described on the previous page).



Pick the Trigger Source *icon as necessary to select the appropriate mode.*	Picking the *Trigger Source* icon toggles the icon between *Host Trigger* and *Swingarm Trigger*. **Note:** If the *Host Trigger* icon is selected, a master host must be selected (PLC or serial), since only a master host can issue a trigger.



VIM2 Powerup Sequence with Serial Device as Master Host

When the VIM2 module is powered up, regardless of the master host selection (PLC, serial, or no master host) it first conducts a number of self-diagnostic tests. The VIM2 module also checks the two configurations stored in the EEPROM; if both are valid, the selected Starting Configuration is downloaded from EEPROM to RAM to become the working configuration.

If the serial line has been selected as the master host, the VIM2 module waits for five seconds for a command from the serial host after the diagnostics are complete. If a command from the serial host *is not* received within five seconds, the VIM2 will automatically be "unlocked," and the VIM2 module will go to *standby** mode. The selection of serial line as the master host is maintained by the VIM2 module.

If a command from the serial host *is* received within five seconds, the VIM2 module will be locked and will go to run mode (unless the command is the *Unlock* command, in which case the VIM2 module goes to *standby** mode).

***Note:** *Standby* mode is an interim mode of the VIM2 module which occurs upon leaving *run* mode, and before entering *setup* mode. When the VIM2 module is in *standby* mode, the title banner is displayed on the monitor. The VIM2 module enters *standby* mode when:

- The VIM2 module, in *run* mode, is unlocked by the master host.
- The VIM2 module is in *setup* mode, a master host is present, the module is locked, and you exit *setup* mode by picking the *Run* icon.
- The VIM2 module is in *run* mode and operating as a standalone unit, and you pick the monitor screen once with the light pen.

Serial Line "Wake-up"

The VIM2 module does not continually monitor the serial line for messages; instead, it waits until an incoming signal is received, and then "wakes up" and begins to receive the incoming signal. So part of the initial incoming command is lost. Thus, each time a command is sent, the VIM2 module will respond to the initial command as though it is an invalid command – in DF1 protocol a **NAK** is issued; in ASCII, the **?** character is issued.

In order to accomodate the necessity for serial line wake-up, the serial host must respond to the initial **NAK** (DF1) or **?** (ASCII) by repeating the original command.

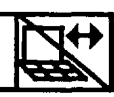
Serial Line Error Tracking

The VIM2 module maintains *three counters* for the purpose of tracking and handling communication errors. You can access the counter data only by sending the *Serial Line Diagnosis* command. You set counter maximums using the *Serial Line Initialization* command. The purpose of these counters and their maximum settings is discussed below.

The first counter tracks the *number of consecutive erroneous messages*, and is incremented each time a message exchange is aborted (for example, when the limit of NAKs, or negative acknowlegements, is surpassed).

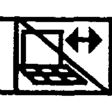
The *consecutive erroneous messages* counter is reset to zero when: a successful message exchange between the serial host and the VIM2 module is achieved; when the serial line driver is initialized (refer to the *Serial Line Initialization* command), when a user-defined maximum number of *consecutive erroneous messages* has been reached (refer to the *Serial Line Initialization* command), or when a user-defined maximum number of *inspections* has been completed since the last reset.

If the *maximum number of consecutive erroneous messages* is reached, a condition called *serial line failure* occurs. When a *serial line failure* occurs, a second counter – which tracks the *number of serial line failures* – is incremented. Other consequences of *serial line failure* depend on whether or not the serial line host is the master host. If the serial line host *is not* the master host, and a serial line failure occurs, the VIM2 module continues operating in run mode, but the serial line communication is shut down, and this icon –



appears on the monitor (unless the image is "frozen," which occurs if the *Freeze on Reject* mode is selected and a reject occurs). While the serial line is shut down, a *third* counter – which tracks the *number of inspections since the last serial line reset* – is incremented. The serial line is reenabled when the *number of inspections since the last serial line reset* counter reaches a user-defined maximum (refer to the *Serial Line Initialization* command).

If the serial line host *is* the master host, and a serial line failure occurs, serial line communication remains enabled. However, this icon appears on the monitor –



and theVIM2 module *halts the inspection process, until a valid message is received* from the serial host.

ASCII Protocol

This section describes the use of ASCII protocol for communication with the VIM2 module, including descriptions of the ASCII commands.

ASCII Protocol Conventions

In describing the ASCII protocol we use the following conventions:

- All ASCII commands begin with the ">" character.
- ASCII commands are shown in bold characters:

  >RCn,CBf [CR]

- Non-printable ASCII characters are shown as follows:

  [CR] = Carriage Return
  [LF] = Line Feed
  _ = Space

- Unless _ is specified, there are no spaces between characters.
- ASCII commands contain only these ASCII characters:

  - Digits 0-9

  - Letters A-Z (upper case), *, –, (*space*), and , (*comma*)

- The **x** is the repeat count, an optional parameter for some commands which indicates the number of times the VIM2 module will repeat the given command. For example:

  >RRx,RD [CR]

  The range for this parameter is between 0 and 255. If you do not include the repeat count, the value of 1 is assumed. If you use the repeat count value of 0 (zero), the command will be executed indefinitely (until another command is received). **Note:** A repeat count of 0 is treated as 1 for Echo commands.

  Note also that when a repeat count is used with any *Read* type command (not including *Read Configuration*), the VIM2 module will send the data after each inspection, until the number of repeats is executed, or until another *Read* type command (including *Read Configuration*) is received; the newly received command will then be executed. Commands such as *Trigger*, or *Change Display Mode* received before the repeats are completed will not cancel the remainder of the command repeats, nor will going to setup mode then returning to run mode.

ASCII Protocol Conventions (continued)

- The **n** qualifier is included in some commands to designate a specific configuration.
- The **f** flag parameter is included in some commands to select specific blocks, bits, or images.

  For example:

  >RCn,CBf [CR]

  Replace **n** in the above command with a digit (0-2) to specify RAM, EEPROM 1 or EEPROM 2. Replace **f** in the above command with a digit to specify a single block. Replace **f** with "*" to specify all blocks. Replace **f** with a range (**f1** – **f2**) to specify a group of blocks as shown in the example below:

  > RC1,CB1-3 [CR]

ASCII Data Conversion

Some ASCII commands involve the exchange of data between the VIM2 module and the host. This data is defined in the command or response format descriptions for each command, where applicable.

Note that, when ASCII protocol is used, data from the VIM2 module is transmitted as hexadecimally-encoded ASCII characters; these ASCII characters must be interpreted as hexadecimal digits for decoding by the user.

For example, a VIM2 module response to this *Read Outputs* command:

>RO,1-4 [CR]

could be the following hexadecimal values:

30 30 20 30 30 20 30 30 20 30 31 13

These values represent the ASCII characters

0 0 *space* 0 0 *space* 0 0 *space* 0 1 *carriage return.*

The above ASCII response would be interpreted, then, as hexadecimal 00 00 00 01 and be used as the response byte data by the user.

Note: Examples of response data are included in the command descriptions in this chapter. These data examples depict hexadecimal values – the hexadecimal values obtained by converting the hexadecimally-encoded ASCII response to hexadecimal data bytes, as described above.

Using ASCII Protocol

After you have made the equipment connections and configured the VIM2 module for RS-485 communications with ASCII as the selected protocol, all ASCII strings issued by the host will be interpreted as commands.

The VIM2 module will validate the command. If the command is valid the VIM2 module will reply:

[CR][LF]

If the command is not valid the VIM2 module will respond:

? [CR] [LF]

The VIM2 module will process all valid commands and discard any invalid commands. Data may or may not be returned with a command depending upon the type of command that was sent. Refer to Appendix G for an ASCII conversion chart.

Note: A simple way to test the RS-485 link is to send the VIM2 module a **[CR]**. If you have the port properly connected and have the VIM2 module configured for RS-485 with ASCII as the selected protocol, you will receive this response: ? **[CR][LF]**. If you receive no response, check your connections and VIM2 module configuration.

The ASCII commands are discussed beginning on the next page.

Serial Line Initialization

Use this command to set two parameters for the serial line error tracking (listed as **n1** and **n2** for this command, and **d1** and **d2** in the response to the *Serial Line Diagnostics* command). Refer also to the *Serial Line Error Tracking* section in this chapter for more information.

This command has the following structure:

>X I,n1,n2 [CR]

n1 = *Decimal value for maximum number of erroneous messages (default = 100). Range is 0-65535.*

n2 = *Decimal value for maximum number of inspections between resets of consecutive erroneous error counter, or between reenabling of communications (default = 500). Range is 0-65535.*

Sending this command also resets the VIM2 module's counter for each parameter (refer to **d3** and **d4** in the response to the *Serial Line Diagnostics* command on the next page).

Set the **n1** value according to the selected baud rate, and the maximum allowable error delay. Set the **n2** value according to the average inspection rate and the required delay before trying to reenable communications.

Host/Mode Stipulations: Valid for either master host or non-master host. Valid in all modes.

Serial Line Diagnosis

Use this command to receive diagnostic data from the VIM2 module regarding the serial communications link. Refer also to the *Serial Line Error Tracking* section in this chapter.

This command has the following structure:

>X D [CR]

The response includes five words of data, d1 – d5:

> **d1 (H) _ d1 (L) _ d2 (H) _ d2 (L) _ d3 (H) _ d3 (L) _ d4 (H) _ d4 (L) _ d5 (H) _ d5 (L) [CR]**

where each data word consists of two bytes, a high byte (H) and a low byte (L), and each byte is represented by two hexadecimal characters (00-FF), followed by a space. Each hexadecimal character is represented by an ASCII character.

Here are descriptions of each data type:

d1: Maximum number of erroneous messages. Set by the *Serial Line Initialization* command. Defaults to 100 (decimal) upon power-up, or each time the VIM2 module configuration is changed.

d2: Maximum number of inspections between two communication resets. Set by the *Serial Line Initialization* command. Defaults to 500 (decimal) upon power-up, or if the VIM2 module configuration is changed.

d3: Current number of consecutive erroneous messages. The VIM2 module increments this counter each time it receives a message which is not recognized as valid.

d4: Current number of inspections since last reset. The VIM2 module increments this counter each time an inspection is conducted.

d5: Number of serial line failures. This number is incremented each time a serial line failure occurs (that is, when the **d3** value reaches the maximum set by **d1**).

An example response would be:

00 _ 64 _ 01 _ F4 _ 00 _ 02 _ 00 _ C8 _ 00 _ 01 [CR]

This response indicates, for example, that **d1** = 00 64 (100 decimal). That is, the maximum number of erroneous messages is set to 100 (this is the default setting).

Host/Mode Stipulations: Valid for either master host or non-master host. Valid in all modes.

Echo

Use the echo command to check the serial communications link. This command will return the same same string of characters that are sent out with the command. This command has the following format:

>Ex,data [CR]

x	=	*Number of times command is repeated (optional, range 0-255, default = 1)* **Note:** *A value of 0 (zero) is treated as a 1 (for this command only).*
data	=	*Character string to be echoed*

For example:

>E2,HELLO [CR]

This command causes the VIM2 module to return:

HELLO [CR] [LF]
HELLO [CR] [LF]

Host/Mode Stipulations: Valid for either master host or non-master host. Valid in all modes.

Read Configuration

Use this command to retrieve specified configuration blocks. You can request the configuration data from either of the two EEPROM configurations. This command has the following format:

>RCn,CBf [CR] or
>RCn,CB f1–f2 [CR]

n	=	*Selected configuration, where:* *1 = EEPROM #1* *2 = EEPROM #2*
f	=	*Selected block. Valid range of 1-4, or use " * " to indicate all blocks.*
f1-f2	=	*Span of blocks (for example, 2–4, or 1–2)*

Read Configuration (continued)

Response to the *Read Configuration* command is:

data . . .
. . . [CR][LF] *for first selected block*
data . . .
. . . [CR][LF] *for second selected block*
etc.

where data consists of a number of two-byte words. Each byte consists of two hexadecimal characters (00-FF), followed by a space, and each individual hexadecimal character is represented by an ASCII character. Refer to Appendix B for configuration block descriptions and sizes.

Host/Mode Stipulations: Valid in *standby* mode for master host or non-master host. Valid in *run* mode for master host only.

Write Configuration

Use this command to download a configuration to the VIM2 module. You download specified configuration blocks (see Appendix B for configuration block information). The VIM2 module stores the downloaded configuration block(s) in a "buffer," or temporary memory location.

Note: After you use the *Write Configuration* command, you must send the *Verify* command in order to have the VIM2 module validate the configuration block(s), and transfer them from the buffer to RAM (the working configuration). After this, you can copy the configuration block(s) to one of the two EEPROM configurations, if desired, using the *Save Configuration to EEPROM* command.

The *Write Configuration* command has the following format:

>W,C,f[CR]
data [CR]
. . .
data [CR]

f = *Selected block. Valid range of 1-4, or use " * " to indicate all blocks.*

or **>W,C,f1–f2 [CR]**
data . . .
. . . [CR] *for first selected block*
data . . .
. . . [CR] *for second selected block*
etc.

f1-f2 = *Span of blocks (for example, 2-4, or 1-2)*

Write Configuration (continued)

Data in the downloaded configuration blocks consists of a number of two-byte words. Each byte consists of two hexadecimal characters (00-FF), followed by a space, and each individual hexadecimal character is represented by an ASCII character.

Host/Mode Stipulations: Valid for master host only. Valid in *standby* mode only.

Verify

Use this command after you download the configuration block(s) using the *Write Configuration* command.

The *Verify* command has the following format:

>V[CR]

There is no VIM2 module response to this command.

When you send this command, the VIM2 module checks the validity of the configuration block(s) stored in the buffer.

Note: When you use the *Write Configuration* command, the VIM2 module stores the downloaded configuration block(s) in a "buffer," or temporary memory location.

If the configuration data in the buffer is *valid*, the VIM2 module copies the data to RAM (the working configuration).

Note: When the configuration is copied to RAM, the configuration previously stored in RAM is overwritten. Also, any statistics that have been stored by the VIM2 module are cleared.

If the data is found to be *not valid*, the VIM2 module's *Configuration Fault* discrete bit is set, and the VIM2 module does not copy the configuration block(s) to RAM.

Note: Since no response is returned for this command, the *Verify* operation may fail without any warning other than the *Configuration Fault* discrete bit. We recommend that you check the *Configuration Fault* discrete bit after sending the *Verify* command (refer to either the *Read Outputs* or the *Read Discrete Results* command for information).

Host/Mode Stipulations: Valid for master host only. Valid in *standby* mode only.

Save RAM Configuration to EEPROM

Use this command to copy the configuration stored in the VIM2 module's RAM (the working configuration) to the EEPROM configuration specified in the command.

The *Save Configuration to EEPROM* command has the following format:

>Sn[CR]

n = *Selected EEPROM, where:*
1 = EEPROM #1
2 = EEPROM #2

For example, to store the RAM configuration to EEPROM #2, send this command:

>S2 [CR]

Host/Mode Stipulations: Valid for master host only. Valid in *standby* mode only.

Load EEPROM Configuration to RAM

Use this command to load one of the two configurations stored in the VIM2 module's EEPROM to the VIM2 module's RAM, thus becoming the new working configuration. When sending this command, you select which EEPROM to load (#1 or #2), and you also specify whether you want the selected EEPROM configuration to become the new start-up configuration.

The *Load EEPROM Configuration to RAM* command has the following format:

>LO,n1,n2[CR]

n1 = *Selected EEPROM, where:*
1 = EEPROM #1
2 = EEPROM #2
n2 = *Save new start-up configuration, where:*
0 = No
Other = Yes

For example, to load EEPROM #2 to the RAM configuration, and then set EEPROM #2 as the starting configuration, send this command:

>LO,2,1 [CR]

Host/Mode Stipulations: Valid for master host only. Valid in *standby* mode only.

Read Outputs

Use this command to have the VIM2 module send the status of the specified discrete bits. The VIM2 module responds to the first request as soon as it is not busy. If a repeat count is included in the command, the VIM2 module provides subsequent responses upon completion of each inspection.

The *Read Outputs* command has the following format:

>Rx,Of [CR] *or* **Rx,Of1-f2 [CR]**	**x** =	*Number of times command is repeated (optional, default = 1)*
	f =	*Selected bit. Valid range of 1-8, or use " * " to indicate all bits.*
	f1-f2 =	*Span of bits (for example, 1–4)*

The discrete bits indicated by the **f** parameter are as follows:

Bit **1** = Module fault bit
Bit **2** = Configuration fault bit
Bit **3** = Module busy bit
Bit **4** = Decision bit
Bit **5** = Probe fault bit
Bit **6** = X/Y compensation fault bit
Bit **7** = 0 (Not Used)
Bit **8** = Configuration busy bit

The response to the command has the following format:

data [CR]

where **data** consists of one byte for each of the bits specified in the *Read Output* command. Each byte consists of two hexadecimal characters, followed by a space, and each hexadecimal character is represented by an ASCII character. The value for each byte is always either 00 (off) or 01 (on).

For example, to request the return of all the discrete bits, with no repeat count, you send this command:

>R,O* [CR]

The example response would include eight bytes (separated by spaces), one byte for each of the eight bits requested:

00_00_00_01_00_00_00_00 [CR]

↑ Byte 1 (Bit 1) **↑ Byte 8 (Bit 8)**

In this example, the fourth byte (the *Decision* bit) = **01**, indicating that in the last inspection the part was rejected.

Host/Mode Stipulations: Valid for either master host or non-master host. Valid in all modes.

Read Discrete Results

Use this command to have the VIM2 module send the on/off status of all the discrete bits, and the pass/fail result of the brightness probe, windows, lines gages, and math tools. The VIM2 module responds to the first request as soon as it is not busy. If a repeat count is included in the command, the VIM2 module provides subsequent responses upon completion of each inspection.

The *Read Discrete Results* command has this format:

>RRx,RD[CR]

x = *Number of times command is repeated (optional, default =1)*

The VIM2 module response has the following format:

byte 1_byte 2_byte 3_byte 4_byte 5_byte 6_byte 7_byte 8_byte 9_byte 10[CR]

where each byte consists of two hexadecimal characters (00-FF), followed by a space, and each individual hexadecimal character is represented by an ASCII character. Convert hex values to binary to obtain status (0 =pass, 1=fail) for each discrete result. Response bytes 1-10 are:

Byte	Description
1	Discrete bit results (see the bit assignments below). Bit **0** = Module fault bit Bit **1** = Configuration fault bit Bit **2** = Module busy bit Bit **3** = Decision bit Bit **4** = Probe fault bit Bit **5** = X/Y compensation fault bit Bit **6** = 0 (Not Used) Bit **7** = Configuration busy bit
2	Brightness probe and window results. Bit 0 = probe status, Bits 1-4 = window 1-4 status, respectively. Bits 5-8 = 0 (not used).
3	Line gages X, Y, 1, and 2 status: Bit **0** = Float gage X, upper function status Bit **1** = Float gage X, lower function status Bit **2** = Float gage Y, upper function status Bit **3** = Float gage Y, lower function status Bit **4** = Line gage 1, upper function status Bit **5** = Line gage 1, lower function status Bit **6** = Line gage 2, upper function status Bit **7** = Line gage 2, lower function status

Read Discrete Results (continued)

Byte	Description *(continued)*
4	Line gages 3-6 status: Bit 0 = line gage 3, upper function status, bit 1 = line gage 3, lower function status, bit 2 = line gage 4, upper function status, etc.
5	Line gages 7-10 status
6	Line gages 11-14 status
7	Line gages 15-18 status
8	Line gages 19-22 status
9	Math tools 1-8 status: Bit 0 = math tool 1 status, bit 1 = math tool 2 status, etc.
10	Math tools 9-12 status (bits 4-7 not used)

For example, suppose the VIM2 module's response to the *Read Discrete Results* command is this (**Note:** Recall that the response values below would be obtained by converting a hexadecimally-encoded ASCII response to hexadecimal values):

08_02_10_00_00_00_00_00_00_00 [CR]

↑ Byte 1 **Byte 10 ↑**

In this example, Byte 1 = 08. This represents the binary value:

0000 1000
↑ Bit 7 ↑ Bit 0

Byte 1 contains the discrete bit status; according to the binary code, the *Decision* bit (bit 3, the fourth bit from the right) is asserted, indicating that in the last inspection the part was rejected.

Byte 2 = **02**. This represents the value **0000 0010** binary. Byte 2 contains the probe and window results. According to the binary code, bit 1 is asserted, meaning window #1 failed.

Byte 3 = **10**. This represents the value **0001 0000** binary. Byte 3 contains the pass/fail result of the upper/lower functions of line gages X, Y, 1, and 2. According to the binary code, bit 4 is asserted, meaning the upper function of line gage 1 failed.

Bytes 4-10 = **00**. This represents the value **0000 0000** binary. According to the binary codes for these bytes, none of the remaining line gage functions or math tools failed.

Host/Mode Stipulations: Valid for either master host or non-master host. Valid in *run* mode only.

Read Results Blocks

Use this command to have the VIM2 module send the specified results block(s). The VIM2 module responds to the first request as soon as it is not busy. If a repeat count is included in the command, the VIM2 module provides subsequent responses upon completion of each inspection.

The *Read Results Blocks* command has this format:

>RRx,RBf[CR]

x = *Number of times command is repeated (optional, default =1)*
f = *Selected block. Valid range of 1-2, or use " * " to indicate both blocks.*

The VIM2 module response has the following format:

data[CR] *for first block specified*

. . .

data[CR] *for last block specified*

where **data** in the uploaded results blocks consists of a number of bytes, each separated by a space (see Appendix C for detailed results block information). Each byte consists of two hexadecimal characters (00-FF), and each individual hexadecimal character is represented by an ASCII character.

Host/Mode Stipulations: Valid for either master host or non-master host. Valid in *run* mode only.

Read Statistics Blocks

Use this command to have the VIM2 module send the specified statistics block(s). The VIM2 module responds to the first request as soon as it is not busy. If a repeat count is included in the command, the VIM2 module provides subsequent responses upon completion of each inspection.

The *Read Statistics Blocks* command has this format:

>RSx,SBf[CR] or
>RSx,SBf1-f2[CR]

x	=	*Number of times command is repeated (optional, default =1)*
f	=	*Selected block. Valid range of 1-4, or use " * " to indicate all blocks.*
f1-f2	=	*Span of blocks (for example 1-3)*

The VIM2 module response has the following format:

data[CR] *for first block specified*

...

data[CR] *for last block specified*

where **data** in the uploaded statistics blocks consists of a number of bytes, each separated by a space (see Appendix D for detailed statistics block information). Each byte consists of two hexadecimal characters (00-FF), and each individual hexadecimal character is represented by an ASCII character.

Host/Mode Stipulations: Valid for either master host or non-master host. Valid in *run* mode only (**Note:** In order for the VIM2 module to collect statistics, it must be running in the *learn* mode).

Lock

Use the *Lock* command to prevent the VIM2 module from entering the setup mode. The command has the following structure:

>L [CR]

There is no VIM2 module response to this command.

If you send the *Lock* command to the VIM2 module while it is in the setup mode, the VIM2 module will go to *run* mode as soon as the user exits the *setup* mode. If you send the *Lock* command to the VIM2 module while it is in the *standby* mode, the VIM2 module will exit the *standby* mode and go to *run* mode.

Host/Mode Stipulations: Valid for master host only. Valid in *setup* or *standby* mode only.

Unlock

Use the *Unlock* command to allow the VIM2 module to enter the *setup* mode. The command has the following structure:

>U [CR]

There is no VIM2 module response to this command.

If you send the *Unlock* command to the VIM2 module while it is in the *run* mode, the VIM2 module will exit the *run* mode and go to *standby* mode.

Note: To enter *setup* mode from *standby* mode, you must pick a bright area on the monitor screen with the light pen.

If you send the *Unlock* command to the VIM2 module while it is in the *standby* mode (it is already unlocked), there will be no effect.

Host/Mode Stipulations: Valid for master host only. Valid in *run* or *standby* mode only.

Trigger

Use the *Trigger* command to initiate a VIM2 module inspection cycle (if the VIM2 module is in run mode), or to have the VIM2 module acquire an image (if the VIM2 module is in setup mode). The command has the following structure:

>T [CR]

There is no VIM2 module response to this command.

Note: Sending the *Trigger* command to the VIM2 module does not activate the SWG TRIGGER LED on the VIM2 module.

Host/Mode Stipulations: Valid for master host only. Valid in *run* mode *if* the VIM2 module is set for host triggering. Valid in *setup* mode *if* the VIM2 module is set for host triggering, and if the *setup* trigger is set to *Run Mode Trigger*.

Change Display Mode

Use the *Change Display Mode* command to change the monitor image displayed during *run* mode to the image specified in this command. The command has the following structure:

>W,D,n [CR]

n = *Selected display, where:*
- *1 = window 1*
- *2 = window 2*
- *3 = window 3*
- *4 = window 4*
- *5 = windows 1-4*
- *6 = "live" image*

There is no VIM2 module response to this command.

Note: If the VIM2 module is displaying a halted image (as a result of running in either the *Halt on Reject* or *Halt on Inspection* arm mode), sending the *Change Display Mode* command will have the same effect as a light pen pick. That is, when the command is sent, the monitor image display will not change to the specified image, but will toggle to the next display in sequence.

Host/Mode Stipulations: Valid for master host or non-master host. Valid in *run* mode only.

Reset Statistics

Use this command to reset all statistics collected while the VIM2 module operated in the *learn* mode. The command has this format:

>ZS [CR]

There is no VIM2 module response to this command.

Host/Mode Stipulations: Valid for master host or non-master host. Valid in *run* mode or *standby* mode only.

ASCII Command Summary

After you have become familiar with the ASCII commands, you can use the following command summary as a quick reference guide.

Table 5.A
ASCII Command Summary

Command	Command Structure	Field Descriptions	Host/Mode Stipulations*
Serial Line Diagnosis	>XD [CR]		For Master / Non-Master Modes SU, SB, R
Serial Line Initialization	>X I,n1,n2 [CR]	n1 = max. erroneous messages (0-65535) n2 = max. inspections until reset (0-65535)	For Master / Non-Master Modes SU, SB, R
Echo	>Ex, data [CR]	x = times repeated (0-255) Data = ASCII string	For Master / Non-Master Modes SU, SB, R
Read Configuration	>RCn, CBf [CR]	n = configuration (1 = EEPROM #1, 2 = EEPROM #2) f = configuration block(s)	Mode SB for Master / Non-Master Modes SB, R for Master
Write Configuration	>W,C,f [CR] >data [CR] . . . >data [CR]	f = configuration block(s) data = configuration data for each specified block	For Master, Mode SB only
Verify	>V [CR]		For Master, Mode SB only
Save Configuration RAM to EEPROM	>Sn [CR]	n = EEPROM configuration (1 = EEPROM #1, 2 = EEPROM #2)	For Master, Mode SB only
Load Configuration From EEPROM to RAM	>LO,n1,n2 [CR]	n1 = EEPROM configuration (1 = EEPROM #1, 2 = EEPROM #2) n2 = new startup configuration (0 = No, Other = Yes)	For Master, Mode SB only
Read Outputs	>Rx, Of [CR]	x = times repeated (0-255) f = selected bit(s)	For Master / Non-Master Modes SU, SB, R
Read Discrete Bit Results	>RRx, RD [CR]	x = times repeated (0-255)	For Master / Non-Master Mode R
Read Results Blocks	>RRx, RBf [CR]	x = times repeated (0-255) f = selected block(s)	For Master / Non-Master Mode R
Read Statistics Blocks	>RSx, SBf [CR]	x = times repeated (0-255) f = selected block(s)	For Master / Non-Master Mode R
Lock	>L [CR]		For Master, Modes SU, SB
Unlock	>U [CR]		For Master, Modes SB, R
Trigger	>T [CR]		For Master, Modes SU, R

*Master and Non-Master indicate host type for which command is valid. SU, SB, and R indicate VIM2 module modes for which command is valid, where SU = setup mode, SB = standby mode, and R = run mode.

ASCII Command Summary
(Continued)

Table 5.A
ASCII Command Summary *(continued)*

Command	Command Structure	Field Descriptions	Host/Mode Stipulations*
Change Display Mode	>W,D,n [CR]	n = selected display (1-4 = windows 1-4, respectively, 5 = all windows, 6 = "live")	For Master / Non-Master Mode R
Reset Statistics	>ZS [CR]		For Master / Non-Master Modes SB, R

*Master and Non-Master indicate host type for which command is valid. SU, SB, and R indicate VIM2 module modes for which command is valid, where SU = setup mode, SB = standby mode, and R = run mode.

DF1 Protocol

The remainder of this chapter describes the use of DF1 protocol commands. After you have made the equipment connections and configured the VIM2 module for RS-485 communications and DF1 protocol, DF1 packets of data can be sent to the VIM2 module.

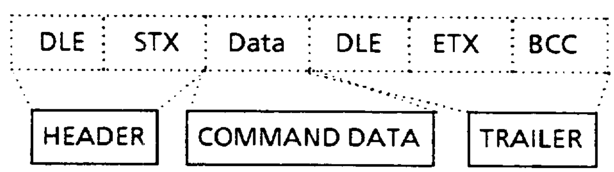
In this section, we only describe the DF1 command structure and the actual DF1 commands that are used with the VIM2 module. For more information about DF1 protocol, refer to Appendix E of this manual.

DF1 Character Set

In the DF1 protocol mode, all data is transferred between the VIM2 module and a host as bytes with values of 00-FF (hex). Refer to Appendix G to convert control codes like ACK and NAK to/from hexadecimal values.

Command Structure

Each command the host device sends to the VIM2 module is represented by a block of data beginning with DLE STX (Data Link Escape, Start of Transmission) and terminated with DLE ETX BCC (Data Link Escape, End Transmission, Block Check Character). The data between the header and trailer characters is the command data. The following shows the structure of a typical command:



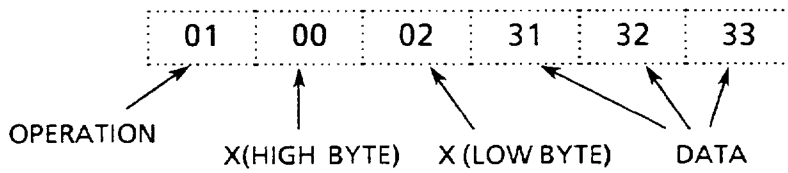
Where: DLE = 10(hex)
STX = 02(hex)
ETX = 03(hex)

Note: To avoid any confusion between DLE (10 hex) and data equal to 10 (hex), a value of 10 (hex) is transmitted as 10 (hex) 10 (hex). The DLE code is transmitted simply as 10 (hex). This is referred to as "DLE stuffing."

Command Structure (continued)

There are different types of fields that can occur in the command data; these fields are described below:

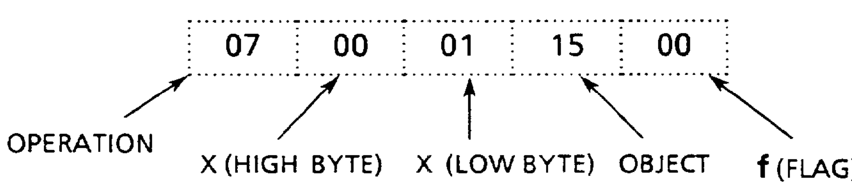
- Operation field- This field appears in every command, and contains the code for the command directed to the VIM2 module. For example, the structure of a command is shown below, where 01 indicates the *Echo* command:



- x (H) / x (L) – These two fields are included with certain commands to indicate the number of times the command is executed (in the example above, the x value is 0002 – the command is executed twice). The valid range is 0000 to FFFF (65,535). If you use a value of 0000, the command will be executed indefinitely (until another command is received). **Note:** A value of 0000 is treated as a 0001 for the *Echo* command.

  Note also that when a repeat count is used with any *Read* type command (not including *Read Configuration*), the VIM2 module will send the data after each inspection, until the number of repeats is executed, or until another *Read* type command (including *Read Configuration*) is received; the newly received command will then be executed. Commands such as *Trigger*, or *Change Display Mode* received before the repeats are completed will not cancel the remainder of the command repeats, nor will going to *setup* mode then returning to *run* mode.

- Data field- Contains data (described per command).
- Object field - This field appears in some commands, in addition to the operation field, to further define the operation. For example, in the diagram below, the operation field, 07, indicates a *Read* command, while the object field, 15, specifies that results blocks will be read.



- The **n** qualifier is included in some commands to indicate a specific configuration, or to indicate parameter values.
- The **f** flag parameter is included in some commands to select specific blocks, bits, or images.

Serial Line Initialization

Use this command to set two parameters for the serial line error tracking (listed as ***n1(H) / n1(L)*** and ***n2(H)/ n2(L)*** for this command, and **d1** and **d2** in the response to the *Serial Line Diagnosis* command). Refer also to the *Serial Line Error Tracking* section in this chapter for more information.

This command has the following structure:

11	n1 (H)	n1 (L)	n2 (H)	n2 (L)

Where:

11 = *Serial Line Initialization* command

n1(H) / n1(L) = *Maximum number of erroneous messages (default = 100). Range is 0000-FFFF.*

n2(H) / n2(L) = *Maximum number of inspections between resets of consecutive erroneous error counter, or between reenabling of communications (default = 500). Range is 0000-FFFF.*

Sending this command also resets the VIM2 module's counter for each parameter (refer to **d3** and **d4** in the response to the *Serial Line Diagnostics* command).

Set the ***n1(H) / n1(L)*** value according to the selected baud rate, and the maximum allowable error delay. Set the ***n2(H)/ n2(L)*** value according to the average inspection rate and the required delay before trying to reenable communications.

Host/Mode Stipulations: Valid for either master host or non-master host. Valid in all modes.

Serial Line Diagnosis

Use this command to receive diagnostic data from the VIM2 module regarding the serial communications link. Refer also to the *Serial Line Error Tracking* section in this chapter for more information.

This command has the following structure:

12

Where: **12** = *Serial Line Diagnosis* command

The VIM2 module response contains a data field which includes five words of data:

d1 (H)	d1 (L)	d2 (H)	d2 (L)	d3 (H)	d3 (L)	d4 (H)	d4 (L)	d5 (H)	d5 (L)

where each word consists of two bytes. Here are descriptions of each word:

d1: Maximum number of erroneous messages. Set by the *Serial Line Initialization* command. Defaults to 100 (decimal) upon power-up, or each time the VIM2 module configuration is changed.

d2: Maximum number of inspections between between two communication resets. Set by the *Serial Line Initialization* command. Defaults to 500 (decimal) upon power-up, or each time the VIM2 module configuration is changed.

d3: Current number of consecutive erroneous messages. The VIM2 module increments this counter each time it receives a message which is not recognized as valid.

d4: Current number of inspections since last reset. The VIM2 module increments this counter each time an inspection is conducted.

d5: Number of serial line failure. The VIM2 module increments this number each time a serial line failure occurs (that is, when the **d3** value reaches the maximum set by **d1**).

An example data field in the response would be:

00	64	01	F4	00	02	00	C8	00	01

This response indicates, for example, that **d1** = 0064 (100 decimal). That is, the maximum number of erroneous messages is set to 100 (this is the default setting).

Host/Mode Stipulations: Valid for either master host or non-master host. Valid in all modes.

Echo Use the *Echo* command to check the communications link. This command will return the same same string of characters that are sent out with the command. This command has the following structure:

01	x (H)	x (L)	Data

Where:

_01_ = *Echo* command

x (H) / x (L) = *Number of times repeated*
Note: *A value of 0000 (zero) is treated as 0001 (for this command only).*

data = *Character string to be echoed*

For example, this command:

DLE	STX	01	00	02	31	32	33	34	35	DLE	ETX	BCC

OPERATION (01) · COUNT (00 02) · DATA (31 32 33 34 35)

will cause the VIM2 module to return:

DLE	STX	31	32	33	34	35	DLE	ETX	BCC
DLE	STX	31	32	33	34	35	DLE	ETX	BCC

Host/Mode Stipulations: Valid for either master host or non-master host. Valid in all modes.

Read Configuration

Use this command to retrieve specified configuration blocks. You can request the configuration data from either of the two EEPROM configurations (see Appendix B for configuration block data). This command has the following format:

06	07	n	f

Where:

n = *Selected configuration, where:*
01 = EEPROM #1
02 = EEPROM #2

f = *Selected block(s), where you create a hex value which sets individual bits:*
Bit 0 = selects block 1
Bit 1 = selects block 2
Bit 2 = selects block 3
Bit 3 = selects block 4
Bits 4-7 not used (zero)

For example, to read configuration blocks 2 and 3 from EEPROM configuration #1, you would send this command:

n → 01, f → 06

DLE	STX	06	07	01	06	DLE	ETX	BCC

0	0	0	0	0	1	1	0

Bits 7 → 4   Bits 3 → 0

where **n** = 01 (selects EEPROM 1), and **f** = 06 (indicating bits 1 and 2 are set, which select blocks 2 and 3).

The VIM2 module responds by sending a message for each of the selected blocks; the data field of each returned message contains the binary content of the selected block:

DLE	STX	... data ...	DLE	ETX	BCC	First selected block ...

...

DLE	STX	... data ...	DLE	ETX	BCC	Last selected block.

Host/Mode Stipulations: Valid in *standby* mode for master host or non-master host. Valid in *run* mode for master host only.

Write Configuration

Use this command to download a configuration to the VIM2 module. You download specified configuration blocks (see Appendix B for configuration block information). The VIM2 module stores the downloaded configuration block(s) in a "buffer," or temporary memory location.

Note: After you use the *Write Configuration* command, you must send the *Verify* command in order to have the VIM2 module validate the configuration block(s), and transfer them from the buffer to RAM (the working configuration).

The *Write Configuration* command has the following format:

0B	07	f

Where: *f* = *Selected block(s), where you use a hex value which sets individual bits:*
Bit 0 = selects block 1
Bit 1 = selects block 2
Bit 2 = selects block 3
Bit 3 = selects block 4
Bits 4-7 not used (zero)

Immediately upon sending the Write configuration command, you then follow with additional DF1 messages which contain the data for each selected configuration block. For example, to write configuration blocks 1 and 2, you would send this command:

DLE	STX	0B	07	03	DLE	ETX	BCC

0	0	0	0	0	0	1	1	f=03

Bits 7 → 4 Bits 3 → 0

where **f**=03 (indicating bits 0 and 1 are set, which select blocks 1 and 2). You would then follow the *Write Configuration* command with two messages – one for each of the selected blocks. The data field of each message would contain the binary content of the respective block:

DLE	STX	... data ...	DLE	ETX	BCC	data for Block 1
DLE	STX	... data ...	DLE	ETX	BCC	data for Block 2

Host/Mode Stipulations: Valid for master host only. Valid in *standby* mode only.

Verify Use this command after you download the configuration block(s) using the *Write Configuration* command.

The *Verify* command has the following format:

0D

There is no VIM2 module response to this command.

When you send this command, the VIM2 module checks the validity of the configuration block(s) stored in the buffer (**Note:** When you use the *Write Configuration* command, the VIM2 module stores the downloaded configuration block(s) in a "buffer," or temporary memory location).

If the configuration data in the buffer is *valid*, the VIM2 module copies the data to RAM (the working configuration).

Note: When the configuration is copied to RAM, the configuration previously stored in RAM is overwritten. Also, any statistics that have been stored by the VIM2 module are cleared.

If the data is found to be *not valid*, the VIM2 module's Configuration Fault discrete bit is set, and the VIM2 module does not copy the configuration block(s) to RAM.

Note: Since no response is returned for this command, the *Verify* operation may fail without any warning other than the Configuration Fault discrete bit. We recommend that you check the Configuration Fault discrete bit after sending the *Verify* command (refer to either the *Read Outputs* command or the *Read Discrete Results* command for information).

Host/Mode Stipulations: Valid for master host only. Valid in *standby* mode only.

Save RAM Configuration to EEPROM

Use this command to copy the configuration stored in the VIM2 module's RAM (the working configuration) to the EEPROM configuration specified in the command.

The *Save Configuration to EEPROM* command has the following format:

08	00	n

Where: **n** = *Selected EEPROM:*
1 = EEPROM #1
2 = EEPROM #2

For example, to store the RAM configuration to EEPROM #2, send this command:

DLE	STX	08	00	02	DLE	ETX	BCC

Host/Mode Stipulations: Valid for master host only. Valid in *standby* mode only.

Load EEPROM Configuration to RAM

Use this command to download one of the two configurations stored in the VIM2 module's EEPROM to the VIM2 module's RAM, thus becoming the new working configuration. When sending this command, you select which EEPROM to load (#1 or #2), and you also specify whether the selected configuration will become the new start-up configuration.

This command has the following format:

03	00	n1	n2

Where: **n1** = *Selected EEPROM, where:*
1 = EEPROM #1
2 = EEPROM #2

n2 = *Save new start-up configuration, where:*
0 = No
Other = Yes

For example, to store EEPROM #2 to RAM, and then set EEPROM #2 as starting configuration, send this command:

				n1	n2			
DLE	STX	03	00	02	01	DLE	ETX	BCC

Host/Mode Stipulations: Valid for master host only. Valid in *standby* mode only.

Read Outputs

Use this command to have the VIM2 module send the status of the specified discrete bits. The VIM2 module responds to the first request as soon as it is not busy. If a repeat count is included in the command, the VIM2 module provides subsequent responses upon completion of each inspection.

The *Read Outputs* command has the following format:

05	x(H)	x(L)	06	f

Where: **x(H) / X (L)** = *Number of times repeated*
f = *Selected block(s), where you use a hex value which selects bits:*
Bit 0 = Module fault
Bit 1 = Configuration fault
Bit 2 = Module busy
Bit 3 = Decision
Bit 4 = Probe fault
Bit 5 = X/Y float gage fault
Bit 6 = 0 (Not Used)
Bit 7 = Configuration busy

For example, to read *bits 0-3* once, send this command:

DLE	STX	05	00	01	06	0F	DLE	ETX	BCC

f = 0F

0	0	0	0	1	1	1	1

Bits 7 → 4 Bits 3 → 0

The response to the command has the following format:

DLE	STX	. . . data . . .	DLE	ETX	BCC

where **data** consists of the bytes specified in the *Read Output* command. Each specified discrete bit is represented by one byte; the value is always either 00 (off) or 01 (on). For example, a response to the command shown above might be:

DLE	STX	00	00	00	01	DLE	ETX	BCC

Bits 0 – 3

in which bit 3 (the *Decision* bit) is asserted, indicating a reject decision.

Host/Mode Stipulations: Valid for either master host or non-master host. Valid in all modes.

Read Discrete Results

Use this command to have the VIM2 module send the on/off status of all the discrete bits, and the pass/fail result of the brightness probe, windows, lines gages, and math tools. The VIM2 module responds to the first request as soon as it is not busy. If the command is to be repeated, the VIM2 module provides subsequent responses after each inspection.

The *Read Discrete Results* command has this format:

07	x(H)	x(L)	06

Where: **x(H) / X (L)** = *Number of times repeated*

The VIM2 module response has the following format:

DLE	STX	byte1	byte2	byte3	byte4	byte5	byte6	byte7	byte8	byte9	byte10	DLE	ETX	BCC

Response bytes 1-10 are interpreted as follows (convert hex values to binary to obtain status (0 = pass, 1 = fail) for each discrete result):

Byte	Description
1	Discrete bit results (use the same bit assignments as listed in the *Read Output* command).
2	Brightness probe and window results: Bit 0 = probe status, bits 1-4 = window 1-4 status, respectively.
3	Line gages X, Y, 1, and 2 results, upper and lower functions: Bit 0 = gage X upper, bit 1 = gage X lower, bit 2 = gage Y upper, bit 3 = gage Y lower, bit 4 = gage 1 upper, bit 5 = gage 1 lower, bit 6 = gage 2 upper, bit 7 = gage 2 lower.
4	Line gages 3-6 results, upper and lower functions: Bit 0 = gage 3 upper, bit 1 = gage 3 lower, bit 2 = gage 4 upper, etc.
5	Line gages 7-10 results
6	Line gages 11-14 results
7	Line gages 15-18 results
8	Line gages 19-22 results
9	Math tools 1-8 results: Bit 0 = math tool 1 status, bit 1 = math tool 2, etc.
10	Math tools 9-12 results (bits 4-7 not used).

***Read Discrete Results** (continued)*

For example, suppose the VIM2 module's response to the *Read Discrete Results* command is this:

DLE	STX	08	02	20	00	00	00	00	00	00	00	DLE	ETX	BCC

In this example:

Byte 1 = **08**. This converts to the value **0000 1000** binary. Byte 1 contains the discrete bit status; according to the binary code (see the *Read Outputs* command), the *Decision* bit (bit 3) is asserted, indicating that in the last inspection the part was rejected.

Byte 2 = **02**. This converts to the value **0000 0010** binary. Byte 2 contains the probe and window results. According to the binary code, window #1 failed.

Byte 3 = **20**. This converts to the value **0010 0000** binary. Byte 3 contains the pass/fail result of the upper/lower functions of line gages X, Y, 1, and 2. According to the binary code, the lower function of line gage 1 failed.

Bytes 4-10 = **00**. This converts to the value **0000 0000** binary. According to the binary codes for these bytes, none of the remaining line gage functions or math tools failed.

Host/Mode Stipulations: Valid for either master host or non-master host. Valid in *run* mode only.

Read Results Blocks

Use this command to have the VIM2 module send the specified results block(s). The VIM2 module responds to the first request as soon as it is not busy. If a repeat count is included in the command, the VIM2 module provides subsequent responses upon completion of each inspection (see Appendix C for results block descriptions).

The *Read Results Blocks* command has this format:

07	x(H)	x(L)	15	f

x = *Number of times command is repeated (optional, default =1)*
f = *Selected block(s), where a hex value selects blocks:*
Bit 0 = Results block 1
Bit 1 = Math tool results
Bits 2-7 Not used (zero)

The response to the command has the following format:

DLE	STX	... data ...	DLE	ETX	BCC

where **data** consists of the binary content of specified results block. If two results blocks are specified in the command, then a separate response is issued for each block.

For example, to read both *blocks* once, send this command:

DLE	STX	07	00	01	15	03	DLE	ETX	BCC

f = 03

0	0	0	0	0	0	1	1

Bits 7 → 4 Bits 3 → 0

The response to the command shown above would be:

DLE	STX	... data ...	DLE	ETX	BCC	data = results block 1
DLE	STX	... data ...	DLE	ETX	BCC	data = results block 2

Host/Mode Stipulations: Valid for either master host or non-master host. Valid in *run* mode only.

Read Statistics Blocks

Use this command to have the VIM2 module send the specified statistics block(s). The VIM2 module responds to the first request as soon as it is not busy. If a repeat count is included in the command, the VIM2 module provides subsequent responses upon completion of each inspection (see Appendix D for statistics block descriptions).

The *Read Statistics Blocks* command has this format:

07	x(H)	x(L)	53	f

x (H) / x (L) = *Number of times command is repeated (optional, default = 1)*

f = *Selected block(s), where a hex value selects blocks:*
Bit 0 = Statistics block 1
Bit 1 = Statistics block 2
Bit 2 = Statistics block 3
Bit 3 = Statistics block 4
Bits 4-7 Not used (zero)

The response to the command has the following format:

DLE	STX	. . . data . . .	DLE	ETX	BCC

where **data** consists of the binary content of the specified statistics block. If multiple statistics blocks are specified, then a separate response is issued for each block.

For example, to read block 3 once, send this command:

DLE	STX	07	00	01	53	04	DLE	ETX	BCC

f = 04

0	0	0	0	0	1	0	0

Bits 7 → 4   Bits 3 → 0

The response to the command shown above would be:

DLE	STX	. . . data . . .	DLE	ETX	BCC

data = block 1

Host/Mode Stipulations: Valid for either master host or non-master host. Valid in *run* mode only (**Note:** In order for the VIM2 module to collect statistics, it must be running in the *learn* mode).

Lock

Use the *Lock* command to prevent the VIM2 module from entering the setup mode. The command has the following structure:

04

There is no VIM2 module response to this command.

If you send the *Lock* command to the VIM2 module while it is in the *setup* mode, the VIM2 module will go to *run* mode as soon as the user exits the *setup* mode. If you send the *Lock* command to the VIM2 module while it is in the *standby* mode, the VIM2 module will exit the *standby* mode and go to *run* mode.

Host/Mode Stipulations: Valid for master host only. Valid in *setup* or *standby* mode only.

Unlock

Use the *Unlock* command to allow the user to enter the VIM2 module setup mode. The command has the following structure:

0A

There is no VIM2 module response to this command.

If you send the *Unlock* command to the VIM2 module while it is in the run mode, the VIM2 module will exit the *run* mode and go to *standby* mode.

Note: To enter *setup* mode from *standby* mode, you must pick a bright area on the monitor screen with the light pen.

If you send the *Unlock* command to the VIM2 module while it is in the *standby* mode (it is already unlocked), there will be no effect.

Host/Mode Stipulations: Valid for master host only. Valid in *run* or *standby* mode only.

Trigger

Use the *Trigger* command to initiate a VIM2 module inspection cycle (if the VIM2 module is in *run* mode), or to have the VIM2 module acquire an image (if the VIM2 module is in setup mode). The command has the following structure:

09

There is no VIM2 module response to this command.

Note: Sending the *Trigger* command to the VIM2 module does not activate the SWG TRIGGER LED on the VIM2 module.

Host/Mode Stipulations: Valid for master host only. Valid in *run* mode *if* the VIM2 module is set for host triggering. Valid in *setup* mode *if* the VIM2 module is set for host triggering, and the *setup* mode trigger is set for *Run Mode Trigger*.

Change Display Mode

Use the *Change Display Mode* command to change the monitor image displayed during run mode to the image specified in this command. The command has the following structure:

0B	02	n

Where: **n** = *Selected display:*
01 = window 1
02 = window 2
03 = window 3
04 = window 4
05 = windows 1-4
06 = "live" image

Note: If the VIM2 module is displaying a halted image (as a result of running in either the *Halt on Reject* or *Halt on Inspection* arm mode), sending the *Change Display Mode* command will have the same effect as a light pen pick. That is, when the command is sent, the monitor image display will not change to the specified image, but will toggle to the next display in sequence.

Host/Mode Stipulations: Valid for master host or non-master host. Valid in *run* mode only.

Reset Statistics

Use this command to reset all statistics collected while the VIM2 module operated in the learn mode. The command has this format:

```
10
```

There is no VIM2 module response to this command.

Host/Mode Stipulations: Valid for master host or non-master host. Valid in *run* mode or *standby* mode only.

DF1 Command Summary

After you have become familiar with the DF1 commands, you can use the following command summary as a quick reference guide.

Table 5.B
DF1 Command Summary

Command	Command Structure	Field Descriptions	Host/Mode Stipulations*
Serial Line Diagnosis	12		For Master / Non-Master Modes SU, SB, R
Serial Line Initialization	11 / n1H / n1L / n2H / n2L	n1H/n1L = max. erroneous messages (0000-FFFF) n2H/n2L = max. inspections until reset (0000-FFFF)	For Master / Non-Master Modes SU, SB, R
Echo	01 / xH / xL / data	xH/xL = repeats (0000-FFFF) Data = hexadecimal string	For Master / Non-Master Modes SU, SB, R
Read Configuration	06 / 07 / n / f	n = configuration (01 = EEPROM 1, 02 = EEPROM 2) f = configuration block(s)	Mode SB for Master / Non-Master Modes SB, R for Master
Write Configuration	0B / 07 / f data . . . data	f = configuration block(s) data = configuration data for each specified block	For Master, Mode SB only
Verify	0D		For Master, Mode SB only
Save Configuration RAM to EEPROM	08 / 00 / n	n = EEPROM configuration (01 = #1, 02 = #2)	For Master, Mode SB only
Load Configuration From EEPROM to RAM	03/ 00/ n1/ n2	n1 = EEPROM configuration (01 = #1, 02 = #2) n2 = new startup configuration (00 = No, other = Yes)	For Master, Mode SB only
Read Outputs	05 / xH / xL / 06 / f	xH/xL = repeats (0000-FFFF) f = selected discrete bit(s)	For Master / Non-Master Modes SU, SB, R
Read Discrete Bit Results	07 / xH / xL / 06	xH/xL = repeats (0000-FFFF)	For Master / Non-Master Mode R
Read Results Blocks	07 / xH / xL / 15 / f	xH/xL = repeats (0000-FFFF) f = selected block(s)	For Master / Non-Master Mode R
Read Statistics Blocks	07 / xH / xL / 53 / f	xH/xL = repeats (0000-FFFF) f = selected block(s)	For Master / Non-Master Mode R
Lock	04		For Master, Modes SU, SB
Unlock	0A		For Master, Modes SB, R
Trigger	09		For Master, Modes SU, R

*Master and Non-Master indicate host type for which command is valid. SU, SB, and R indicate VIM2 module modes for which command is valid, where SU = setup mode, SB = standby mode, and R = run mode.

Command Summary
(Continued)

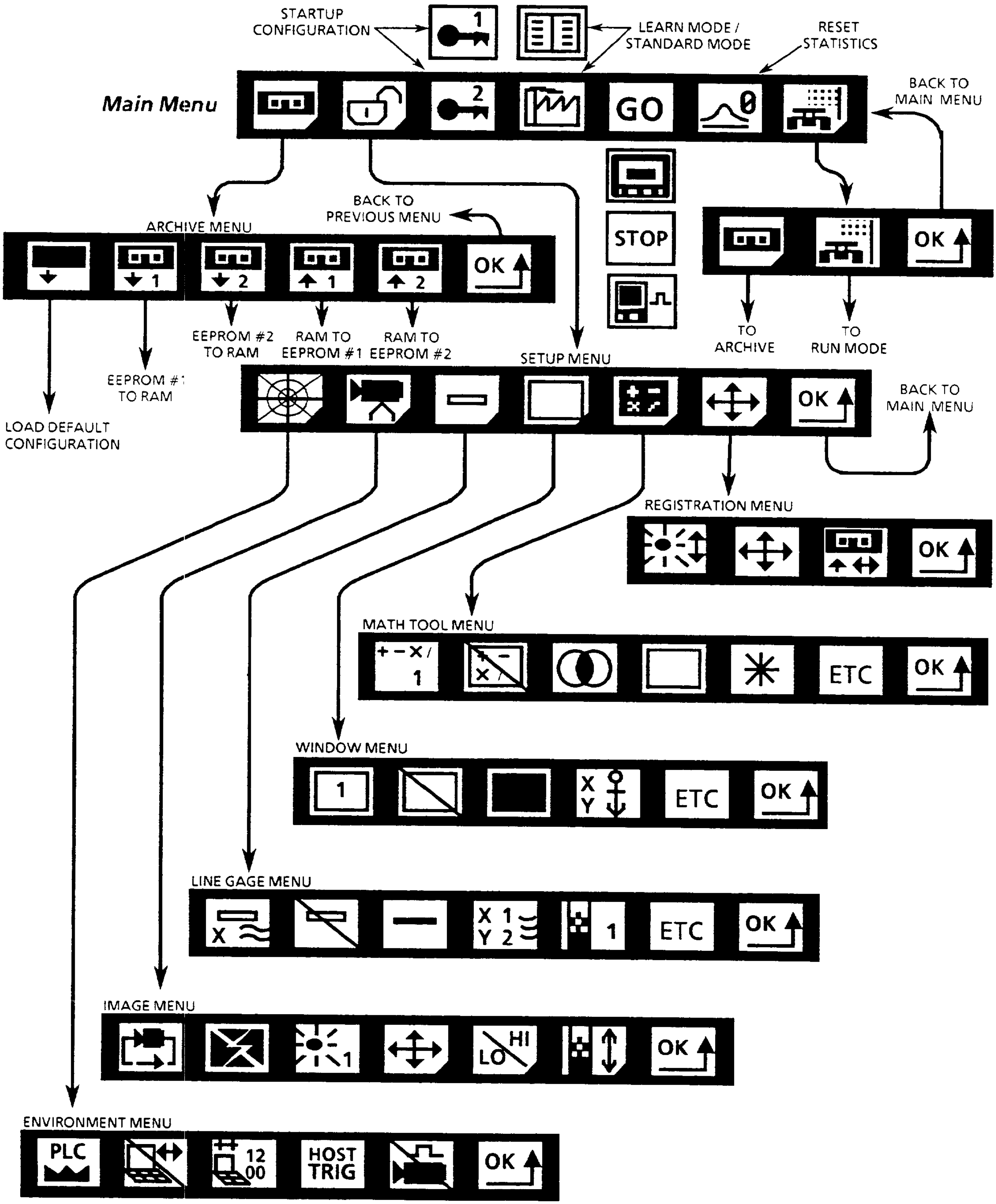
Table 5.B
DF1 Command Summary *(continued)*

Command	Command Structure	Field Descriptions	Host/Mode Stipulations*
Change Display Mode	0B / 02 / n	n = selected display (01-04 = windows 1-4, 05 = all windows, 06 = "live")	For Master / Non-Master Mode R
Reset Statistics	10		For Master / Non-Master Modes SB, R

*Master and Non-Master indicate host type for which command is valid. SU, SB, and R indicate VIM2 module modes for which command is valid, where SU = setup mode, SB = standby mode, and R = run mode.

(This page intentionally left blank)

Appendix A Menu Branching Diagrams



ENVIRONMENT MENU

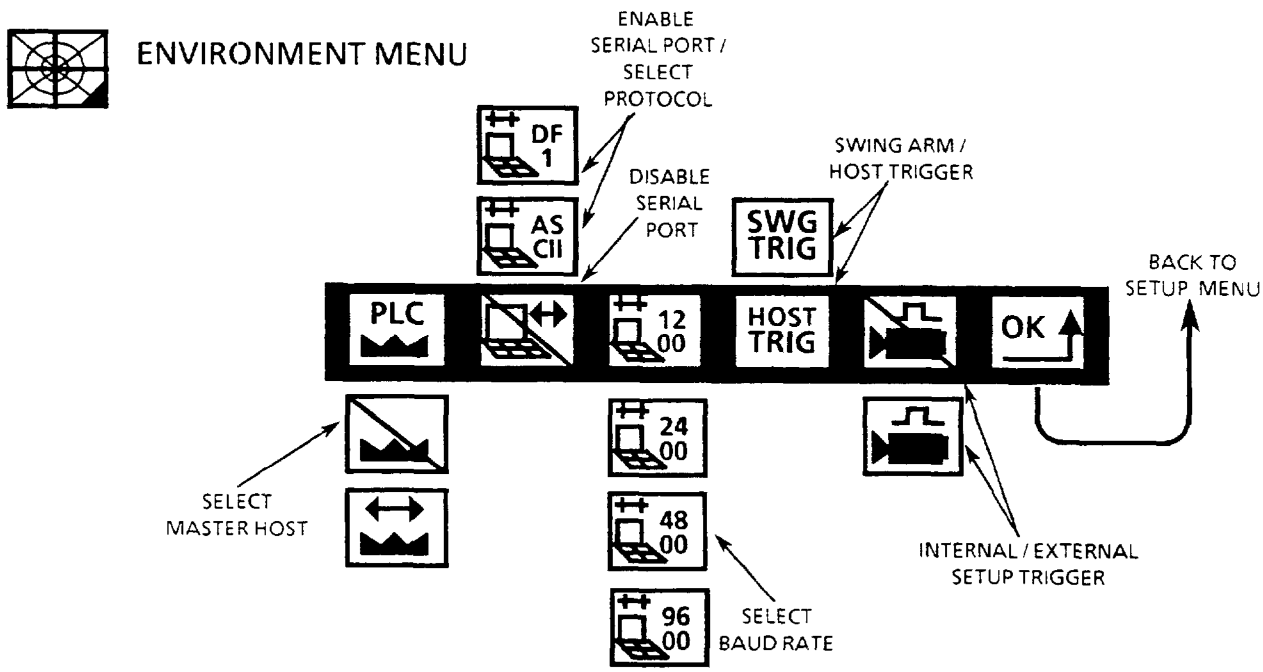
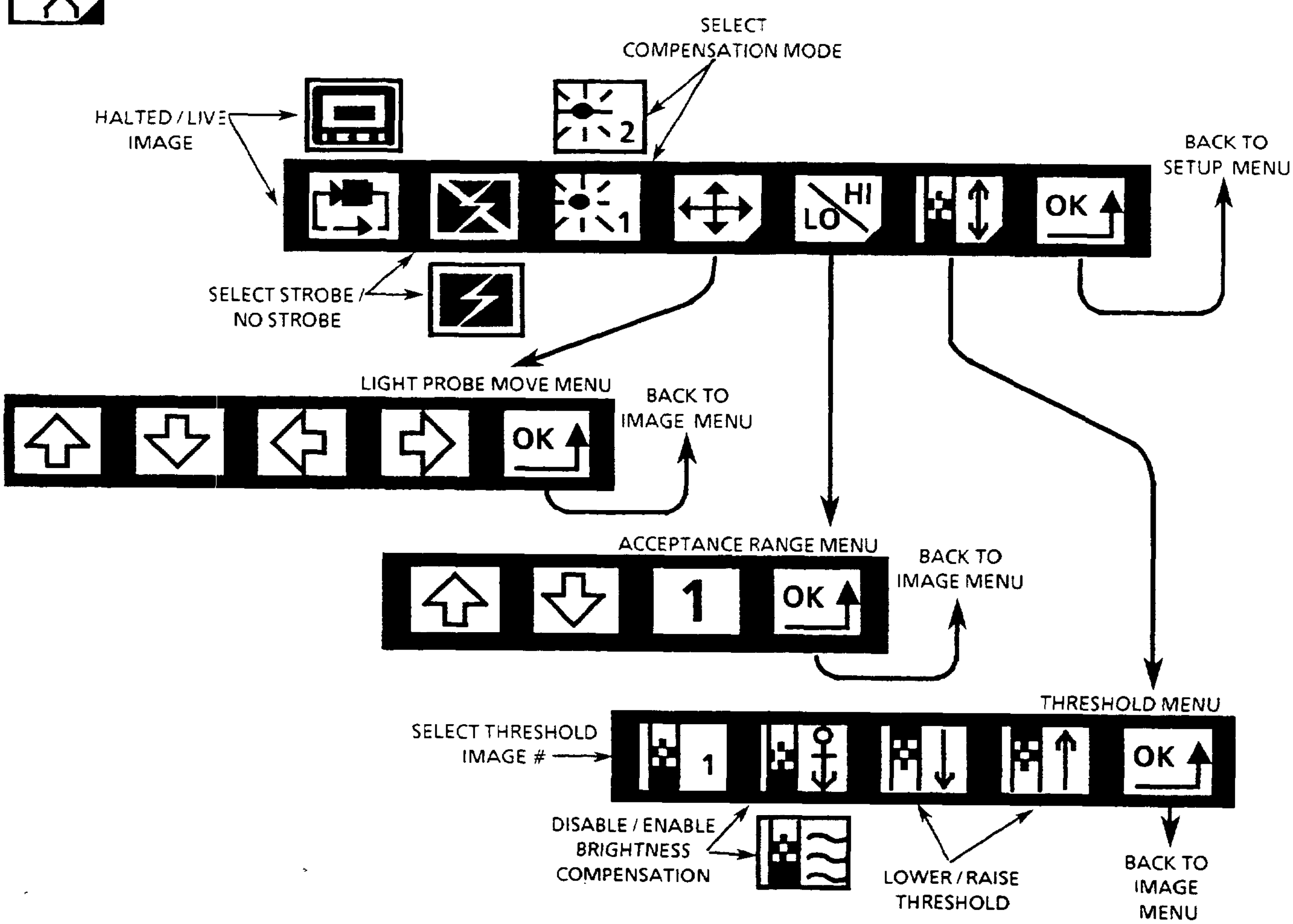
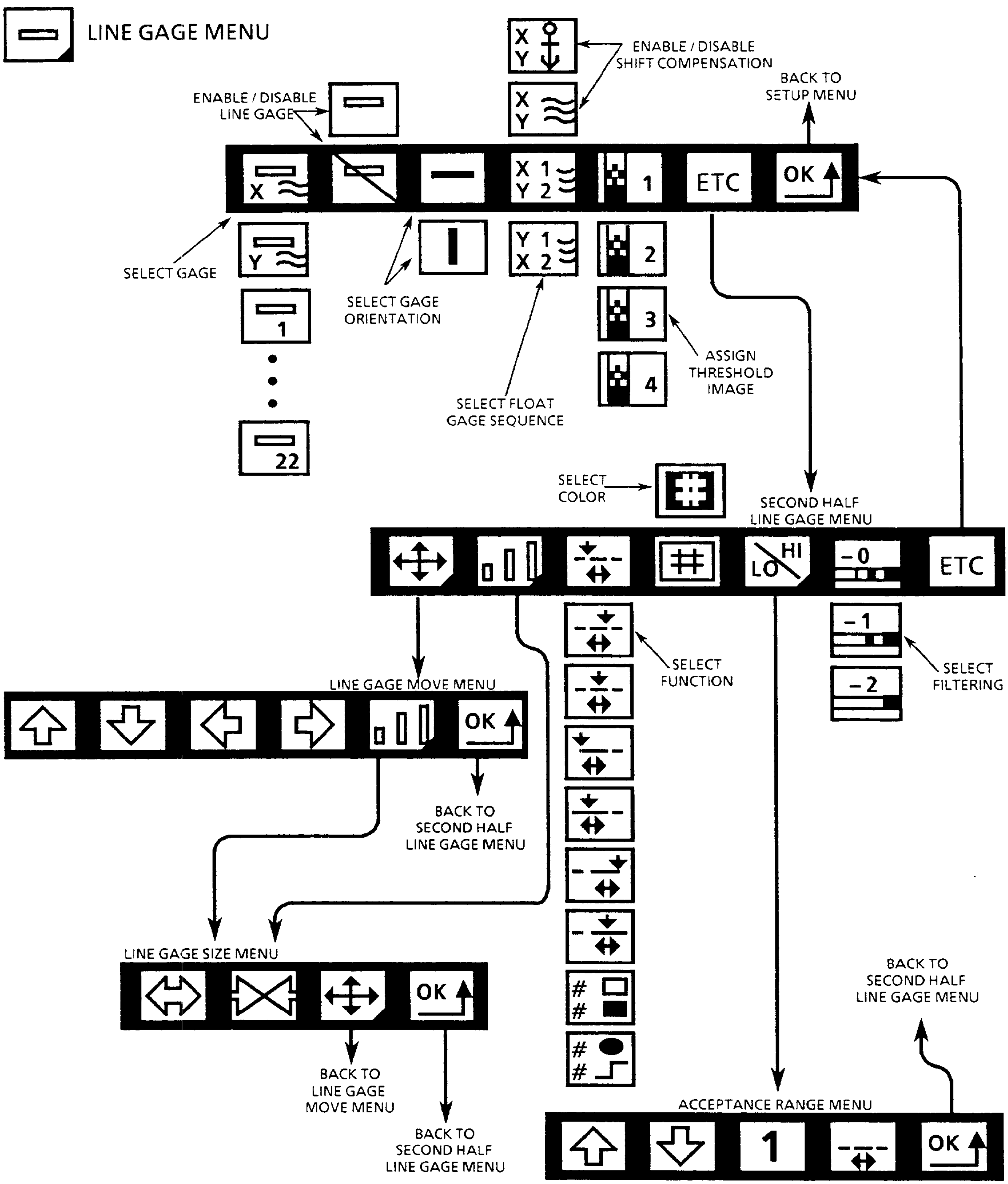
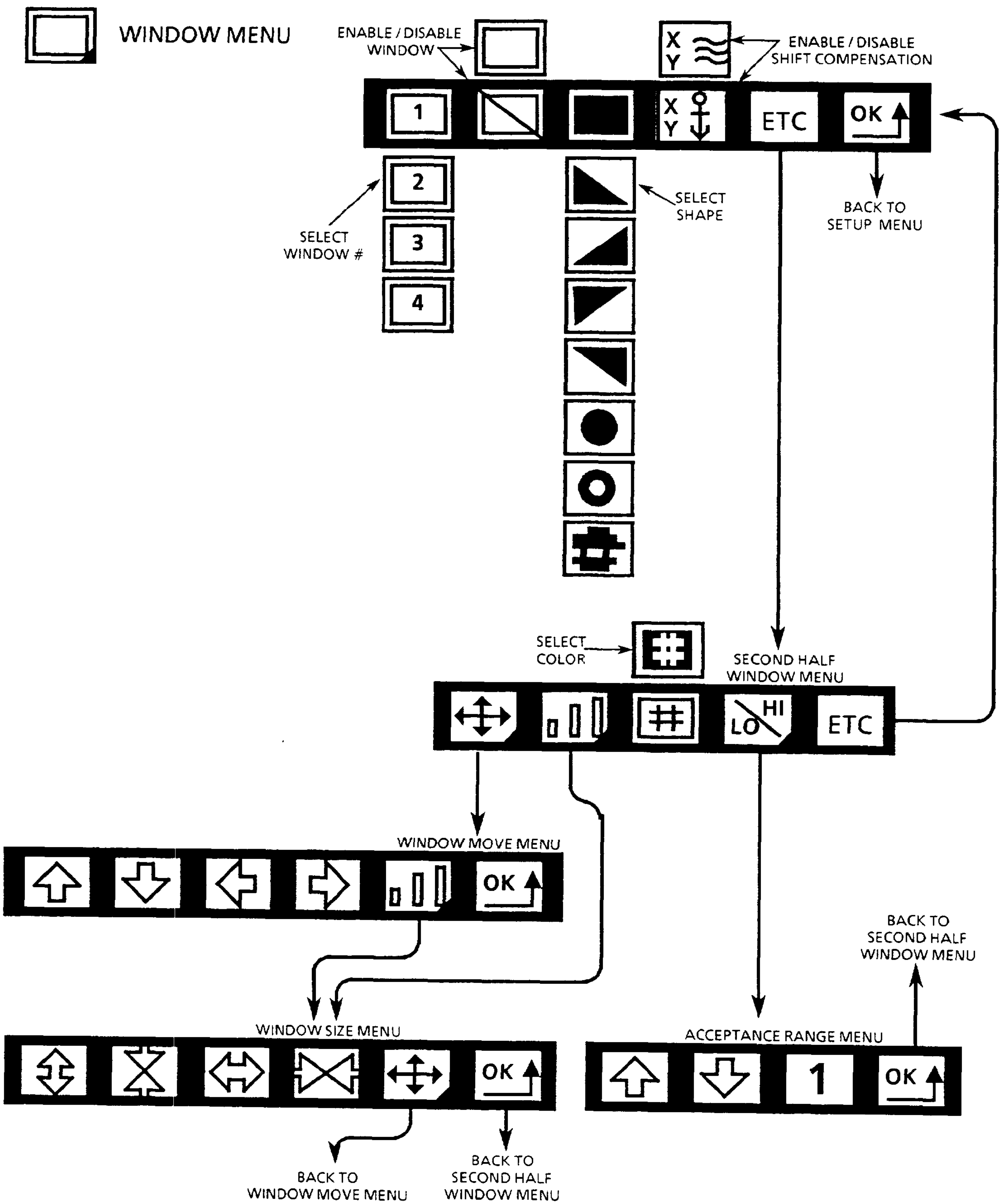
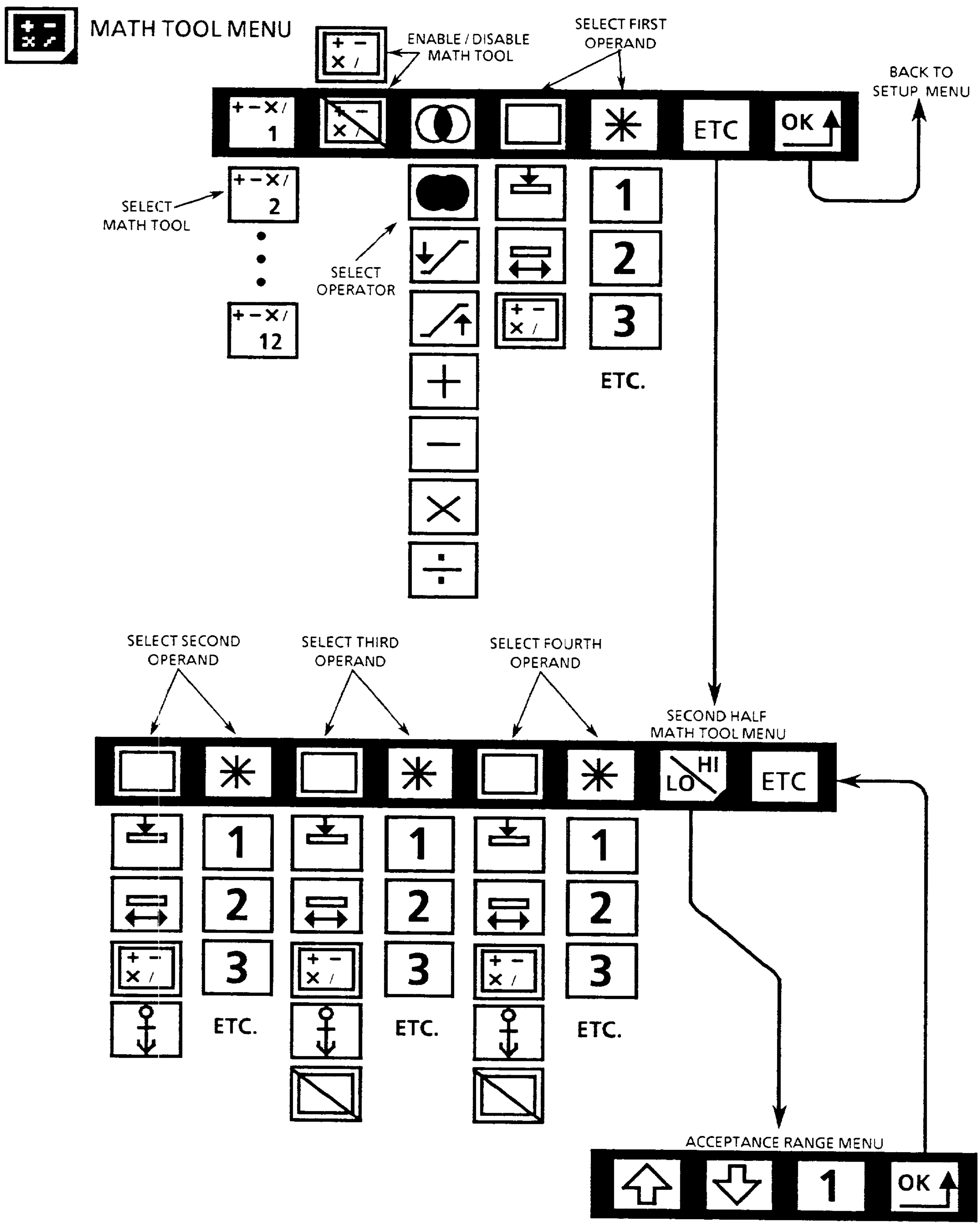


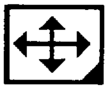
IMAGE MENU

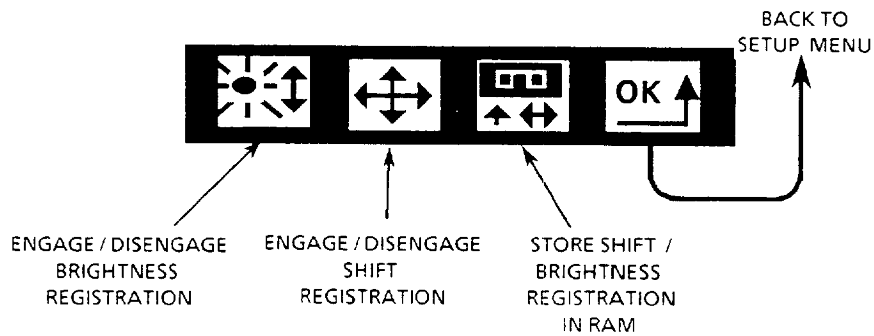



LINE GAGE MENU
ENABLE / DISABLE SHIFT COMPENSATION
BACK TO SETUP MENU
ENABLE / DISABLE LINE GAGE
ETC
OK
SELECT GAGE
SELECT GAGE ORIENTATION
ASSIGN THRESHOLD IMAGE
SELECT FLOAT GAGE SEQUENCE
SELECT COLOR
SECOND HALF LINE GAGE MENU
HI
LO
ETC
SELECT FUNCTION
SELECT FILTERING
LINE GAGE MOVE MENU
OK
BACK TO SECOND HALF LINE GAGE MENU
LINE GAGE SIZE MENU
OK
BACK TO LINE GAGE MOVE MENU
BACK TO SECOND HALF LINE GAGE MENU
BACK TO SECOND HALF LINE GAGE MENU
ACCEPTANCE RANGE MENU
OK

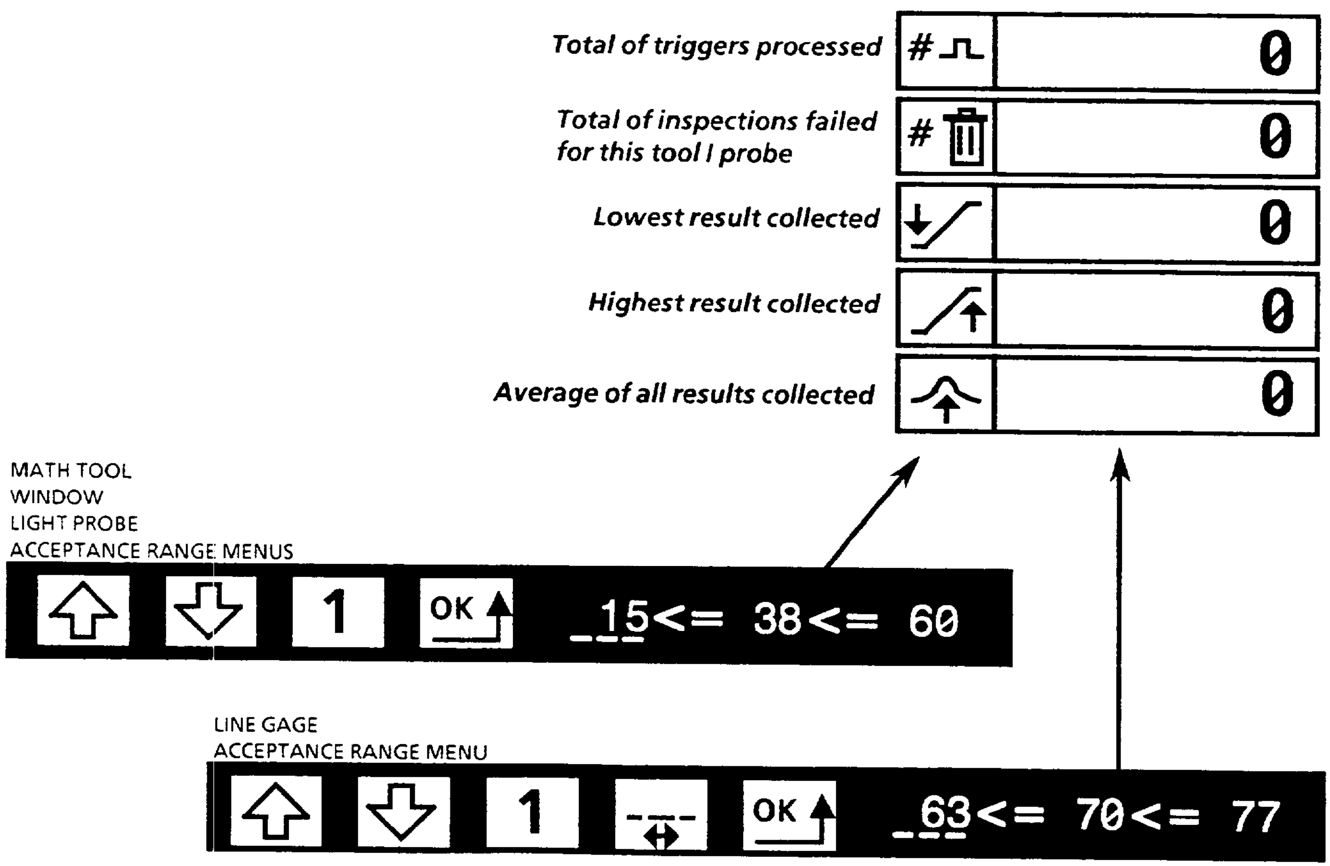

WINDOW MENU
ENABLE / DISABLE WINDOW
ENABLE / DISABLE SHIFT COMPENSATION
1
ETC
OK
2
3
4
SELECT WINDOW #
SELECT SHAPE
BACK TO SETUP MENU
SELECT COLOR
SECOND HALF WINDOW MENU
HI LO
ETC
WINDOW MOVE MENU
OK
BACK TO SECOND HALF WINDOW MENU
WINDOW SIZE MENU
OK
ACCEPTANCE RANGE MENU
1
OK
BACK TO WINDOW MOVE MENU
BACK TO SECOND HALF WINDOW MENU


MATH TOOL MENU
ENABLE / DISABLE MATH TOOL
SELECT FIRST OPERAND
BACK TO SETUP MENU
+ − × / 1
ETC
OK
SELECT MATH TOOL
+ − × / 2
+ − × / 12
SELECT OPERATOR
1
2
3
ETC.
SELECT SECOND OPERAND
SELECT THIRD OPERAND
SELECT FOURTH OPERAND
SECOND HALF MATH TOOL MENU
HI LO
ETC
ACCEPTANCE RANGE MENU
1
OK

 REGISTRATION MENU



STATISTICS FUNCTIONS



Appendix B Configuration Blocks

CONFIGURATION BLOCK 1 OF 4 (Block Length of 30 Words)			
WORD (16 Bits)	BIT (Decimal)	FUNCTION	VALUES
1	0	Source of Trigger	0 = Host, 1 = Swingarm
	1	Strobe Light Use	0 = Disabled, 1 = Enabled
	2	Setup Trigger	0 = Standalone 1 = Runtime
	3-4	Master Host	0 = PLC, 1 = Standalone, 2 = RS-485
	5-7	Reserved for Future Expansion	Zero
	8-9	Freeze Status	0 = Go, 1 = Freeze on reject, 2 = Halt on reject, 3 = Halt on next inspection
	10	Runtime mode	0 = Standard, 1 = Learn
	11-12	RS-485 Protocol	0 = Disable, 1 = DF1, 2 = ASCII
	13-15	RS-485 Baud Rate	0 = 001100, 1 = 2400, 2 = 4800, 3 = 9600
2	0-7	Probe X-Axis Position	Column Value of 0 to 248
	8-15	Probe Y-Axis Position	Row Value of 16 to 240
3	0-7	Probe Compensation Reference Level	Value of 0 to 255
	8	Probe Compensation Mode	0 = Immed. Brightness Comp., 1 = Next Field Brightness Comp.
	9-15	Reserved for Future Expansion	Zero
4	0-7	Probe Lo Acceptance Range Limit	Value of 0 to 255
	8-15	Probe Hi Acceptance Range Limit	Value of 0 to 255

CONFIGURATION BLOCK 1 OF 4 (Block Length of 30 Words)			
WORD (16 Bits)	BIT (Decimal)	FUNCTION	VALUES
5	0-7	Window 1 Base X-Axis Position	Column Value of 0 to 254
	8-15	Window 1 Base Y-Axis Position	Row Value of 11 to 252
6	0-15	Window Lo Acceptance Range Limit	Value of 0 to 62,464
7	0-15	Window Hi Acceptance Range Limit	Value of 0 to 62,464
8	0-3*	Window Shape	0 = Rectangle, 1 to 4 = Triangles, 5 = Circle, 6 = Doughnut 7 = Through The Lens
	4	Window Enable Flag	0 = Disabled, 1 = Enabled
	5	X/Y Anchored/ Float Value Flag	0 = X/Y Anchored, 1 = X/Y Floating
	6	Count White/Black Pixels Select	0 = Black, 1 = White
	7	Brightness Float Status	0 = Anchored, 1 = Floating
	8-15	Binary Threshold Value	Value of 0 to 255
9		Meaning of Word 9 Varies Depending on Window Shape (See Word 8 Bits 0-3)*	
		Rectangular Window (Bottom Rt. Cnr.)	
	0-7	Window Corner X-Axis Position	Column Value of 0 to 254
	8-15	Window Corner Y-Axis Position	Row Value of 11 to 252
		Triangular Window	
	0-7	X-Axis Position of Horizontal Leg	Column Value of 0 to 254
	8-15	Y-Axis Position of Vertical Leg	Row Value of 11 to 252
		Circular Window	
	0-7	Radius of Circle	Value of 0 to 127
	8-15	Reserved for Future Expansion	
		Doughnut Window	
	0-7	Radius of Outer Circle	Value of 5 to 127
	8-15	Radius of Inner Circle	Value of 1 to 123
		Through the Lens Window	
	0-15	Reserved for Future Expansion	
10	0-15	Window area in Square Pixels	Value of 0 to 62,464
11-16		Window 2 Data -- Same as Words 5-10	
17-22		Window 3 Data -- Same as Words 5-10	
23-28		Window 4 Data -- Same as Words 5-10	
29	0-15	Configuration Revision Level	
30	0-15	Reserved for Future Expansion	Zero

CONFIGURATION BLOCK 2 OF 4 (Block Length of 62 Words)			
WORD (16 Bits)	**BIT** (Decimal)	**FUNCTION**	**VALUES**
1	0-7	X Gauge - X-Axis Start of Line	Column Value of 0 to 254
	8-15	Y Gauge - Y-Axis Start of Line	Row Value of 11 to 252
2	0-7	Lo Accept. Range Limit - Upper Function	Value of 0 to 255
	8-15	Hi Accept. Range Limit - Upper Function	Value of 0 to 255
3	0-7	Lo Accept. Range Limit - Lower Function	Value of 0 to 255
	8-15	Hi Accept. Range Limit - Lower Function	Value of 0 to 255
4	0-1	Selected Threshold Number for Application of Line Gauge	0 = Threshold 1, 1 = Threshold 2 2 = Threshold 3, 3 = Threshold 4
	2-3	X/Y Float Status	0 = Anchored, 1 = Floating 2 = X,Y Sequence, 3 = Y,X Sequence
	4	Line Status Flag	0 = Disabled, 1 = Enabled
	5	Line Direction Flag	0 = Horizontal, 1 = Vertical
	6	White/Black Count Selection	0 = Black, 1 = White
	7	Reserved for Future Expansion	Zero
	8-11	Line Function Code	Value of 0 to 8 0 = Left Edge & Width of Largest Blob 1 = Right Edge & Width of Largest Blob 2 = Center & Width of Largest Blob 3 = Left Edge & Width of Leftmost Blob 4 = Center & Width of Leftmost Blob 5 = Right Edge & Width of Rightmost Blob 6 = Center & Width of Rightmost Blob 7 = Count White & Black Pixels 8 = Count Number of Blobs & Number of Edges
	12-13	Line Filter Code	0 = No Filter, 1 = Filter 1, 2 = Filter 2
	14-15	Reserved for Future Expansion	Zero
5	0-7	Line End Point Location	Value of 0 to 254
	8-15	Floating Reference Zero Ordinate	Value of 0 to 254 (Zero for Line Gauges 1 to 22)
6-10		Config. Data For Y Gauge	Same Assignments as Words 1-5
11-15		Configuration Data For Line Gauge 1	Same Assignments as Words 1-5
16-20		Configuration Data For Line Gauge 2	Same Assignments as Words 1-5
21-25		Configuration Data For Line Gauge 3	Same Assignments as Words 1-5
26-30		Configuration Data For Line Gauge 4	Same Assignments as Words 1-5
31-35		Configuration Data For Line Gauge 5	Same Assignments as Words 1-5
36-40		Configuration Data For Line Gauge 6	Same Assignments as Words 1-5
41-45		Configuration Data For Line Gauge 7	Same Assignments as Words 1-5
46-50		Configuration Data For Line Gauge 8	Same Assignments as Words 1-5

CONFIGURATION BLOCK 2 OF 4 (Block Length of 62 Words)			
WORD (16 Bits)	**BIT** (Decimal)	**FUNCTION**	**VALUES**
51-55		Configuration Data For Line Gauge 9	Same Assignments as Words 1-5
56-60		Configuration Data For Line Gauge 10	Same Assignments as Words 1-5
61	0-15	Configuration Revision Level	
62	0-15	Reserved for Future Expansion	Zero

CONFIGURATION BLOCK 3 OF 4 (Block Length of 63 Words)			
WORD (16 Bits)	**BIT** (Decimal)	**FUNCTION**	**VALUES**
1	0-7	Line Gauge 11 - X-Axis Start of Line	Column Value of 0 to 254
	8-15	Line Gauge 11 - Y-Axis Start of Line	Row Value of 11 to 252
2	0-7	Lo Accept. Range Limit - Upper Function	Value of 0 to 255
	8-15	Hi Accept. Range Limit - Upper Function	Value of 0 to 255
3	0-7	Lo Accept. Range Limit - Lower Function	Value of 0 to 255
	8-15	Hi Accept. Range Limit - Lower Function	Value of 0 to 255
4	0-1	Selected Threshold Number for Application of Line Gauge	0 = Threshold 1, 1 = Threshold 2 2 = Threshold 3, 3 = Threshold 4
	2-3	X/Y Float Status	0 = Disabled, 1 = Enabled
	4	Line Status Flag	0 = Disabled, 1 = Enabled
	5	Line Direction Flag	0 = Horizontal, 1 = Vertical
	6	White/Black Count Selection	0 = Black, 1 = White
	7	Reserved for Future Expansion	Zero
	8-11	Line Function Code	Value of 0 to 8 0 = Left Edge & Width of Largest Blob 1 = Right Edge & Width of Largest Blob 2 = Center & Width of Largest Blob 3 = Left Edge & Width of Leftmost Blob 4 = Center & Width of Leftmost Blob 5 = Right Edge & Width of Rightmost Blob 6 = Center & Width of Rightmost Blob 7 = Count White & Black Pixels 8 = Count Number of Blobs & Number of Edges
	12-13	Line Filter Code	0 = No Filter, 1 = Filter 1, 2 = Filter 2
	14-15	Reserved for Future Expansion	
5	0-7	Line End Point Location	Value of 0 to 254
	8-15	Reserved for Future Expansion	Zero
6-10		Configuration Data For Line Gauge 12	Same Assignments as Words 1-5
11-15		Configuration Data For Line Gauge 13	Same Assignments as Words 1-5
16-20		Configuration Data For Line Gauge 14	Same Assignments as Words 1-5
21-25		Configuration Data For Line Gauge 15	Same Assignments as Words 1-5
26-30		Configuration Data For Line Gauge 16	Same Assignments as Words 1-5
31-35		Configuration Data For Line Gauge 17	Same Assignments as Words 1-5
36-40		Configuration Data For Line Gauge 18	Same Assignments as Words 1-5
41-45		Configuration Data For Line Gauge 19	Same Assignments as Words 1-5
46-50		Configuration Data For Line Gauge 20	Same Assignments as Words 1-5
51-55		Configuration Data For Line Gauge 21	Same Assignments as Words 1-5

CONFIGURATION BLOCK 3 OF 4 (Block Length of 63 Words)			
WORD (16 Bits)	**BIT** (Decimal)	**FUNCTION**	**VALUES**
56-60		Configuration Data For Line Gauge 22	Same Assignments as Words 1-5
61	0-15	Configuration Revision Level	
62	0-15	Reserved for Future Expansion	Zero
63	0-15	Reserved for Future Expansion	Zero

CONFIGURATION BLOCK 4 OF 4 (Block Length of 64 Words)			
WORD (16 Bits)	**BIT** (Decimal)	**FUNCTION**	**VALUES**
1-5		Configuration Data For Math Tool 1	
1	0	Math Tool Enable	0 = Disabled, 1 = Enabled
	1-4	Operation	0 = AND, 1 = OR, 2 = Minimum, 3 = Maximum, 4 = Add, 5 = Subtract, 6 = Multiply, 7 = Divide
	5-15	Reserved for Future Expansion	Zero
2	0-2	First operand type (**Note**: Refer to Chapter 12 for operand type restrictions)	0 = Not used, 1 = Window, 2 = Gage Upper, 3 = Gage Lower 4 = Math tool, 5 = Constant
	3-7	First operand number(**Note**: Refer to Chapter 12 for operand number restrictions)	Value of 0-31 Window: 0 = All, 1-4 Gage : 0 = All, 1-22 Math tool: 1-12 Constant: 0-29 = 1-30 30 = 50 31 = 100
	8-10	Second operand type	See first operand type
	11-15	Second operand number	See first operand number
3	0-2	Third operand type	See first operand type
	3-7	Third operand number	See first operand number
	8-10	Fourth operand type	See first operand type
	11-15	Fourth operand number	See first operand number
4	0-15	Low Accept Range Limit	Value of 0 to 65534
5	0-15	High Accept Range Limit	Value of 0 to 65534
6-10		Configuration Data For Math Tool 2	Same as Math Tool 1, Words 1-5
11-15		Configuration Data For Math Tool 3	Same as Math Tool 1, Words 1-5
16-20		Configuration Data For Math Tool 4	Same as Math Tool 1, Words 1-5
21-25		Configuration Data For Math Tool 5	Same as Math Tool 1, Words 1-5
26-30		Configuration Data For Math Tool 6	Same as Math Tool 1, Words 1-5
31-35		Configuration Data For Math Tool 7	Same as Math Tool 1, Words 1-5
36-40		Configuration Data For Math Tool 8	Same as Math Tool 1, Words 1-5
41-45		Configuration Data For Math Tool 9	Same as Math Tool 1, Words 1-5
46-50		Configuration Data For Math Tool 10	Same as Math Tool 1, Words 1-5
51-55		Configuration Data For Math Tool 11	Same as Math Tool 1, Words 1-5
56-60		Configuration Data For Math Tool 12	Same as Math Tool 1, Words 1-5

CONFIGURATION BLOCK 4 OF 4 (Block Length of 64 Words)			
WORD (16 Bits)	**BIT** (Decimal)	**FUNCTION**	**VALUES**
61	0-15	Configuration Revision Level	
62	0-15	Reserved for Future Expansion	Zero
63	0-15	Reserved for Future Expansion	Zero
64	0-15	Reserved for Future Expansion	Zero

Appendix C Results Blocks

RESULTS BLOCK 1 OF 2 (Block Length of 59 Words)			
WORD (16 Bits)	**BIT** (Decimal)	**FUNCTION**	**VALUES** (The "*" Indicates Values Affected by the Binary/BCD Discrete Bit Setting)
1	0	Brightness Probe Accept. Range Status	0 = Accept, 1 = Reject or Error
	1	Window 1 Acceptance Range Status	0 = Accept, 1 = Reject or Error
	2	Window 2 Acceptance Range Status	0 = Accept, 1 = Reject or Error
	3	Window 3 Acceptance Range Status	0 = Accept, 1 = Reject or Error
	4	Window 4 Acceptance Range Status	0 = Accept, 1 = Reject or Error
	5-15	Reserved for Future Expansion	Zero
2	0	Upper Function Status of X-Float Gauge	0 = Accept, 1 = Reject or Error
	1	Lower Function Status of X Float Gauge	0 = Accept, 1 = Reject or Error
	2	Upper Function Status of Y-Float Gauge	0 = Accept, 1 = Reject or Error
	3	Lower Function Status of Y Float Gauge	0 = Accept, 1 = Reject or Error
	4	Upper Function Status of Line Gauge 1	0 = Accept, 1 = Reject or Error
	5	Lower Function Status of Line Gauge 1	0 = Accept, 1 = Reject or Error
	6	Upper Function Status of Line Gauge 2	0 = Accept, 1 = Reject or Error
	7	Lower Function Status of Line Gauge 2	0 = Accept, 1 = Reject or Error
	8	Upper Function Status of Line Gauge 3	0 = Accept, 1 = Reject or Error
	9	Lower Function Status of Line Gauge 3	0 = Accept, 1 = Reject or Error
	10	Upper Function Status of Line Gauge 4	0 = Accept, 1 = Reject or Error
	11	Lower Function Status of Line Gauge 4	0 = Accept, 1 = Reject or Error
	12	Upper Function Status of Line Gauge 5	0 = Accept, 1 = Reject or Error
	13	Lower Function Status of Line Gauge 5	0 = Accept, 1 = Reject or Error
	14	Upper Function Status of Line Gauge 6	0 = Accept, 1 = Reject or Error
	15	Lower Function Status of Line Gauge 6	0 = Accept, 1 = Reject or Error
3	0	Upper Function Status of Line Gauge 7	0 = Accept, 1 = Reject or Error
	1	Lower Function Status of Line Gauge 7	0 = Accept, 1 = Reject or Error
	2	Upper Function Status of Line Gauge 8	0 = Accept, 1 = Reject or Error
	3	Lower Function Status of Line Gauge 8	0 = Accept, 1 = Reject or Error
	4	Upper Function Status of Line Gauge 9	0 = Accept, 1 = Reject or Error
	5	Lower Function Status of Line Gauge 9	0 = Accept, 1 = Reject or Error
	6	Upper Function Status of Line Gauge 10	0 = Accept, 1 = Reject or Error
	7	Lower Function Status of Line Gauge 10	0 = Accept, 1 = Reject or Error
	8	Upper Function Status of Line Gauge 11	0 = Accept, 1 = Reject or Error
	9	Lower Function Status of Line Gauge 11	0 = Accept, 1 = Reject or Error
	10	Upper Function Status of Line Gauge 12	0 = Accept, 1 = Reject or Error
	11	Lower Function Status of Line Gauge 12	0 = Accept, 1 = Reject or Error
	12	Upper Function Status of Line Gauge 13	0 = Accept, 1 = Reject or Error
	13	Lower Function Status of Line Gauge 13	0 = Accept, 1 = Reject or Error
	14	Upper Function Status of Line Gauge 14	0 = Accept, 1 = Reject or Error
	15	Lower Function Status of Line Gauge 14	0 = Accept, 1 = Reject or Error

RESULTS BLOCK 1 OF 2 (Block Length of 59 Words)			
WORD (16 Bits)	BIT (Decimal)	FUNCTION	VALUES (The "*" Indicates Values Affected by the Binary/BCD Discrete Bit Setting)
4	0	Upper Function Status of Line Gauge 15	0 = Accept, 1 = Reject or Error
	1	Lower Function Status of Line Gauge 15	0 = Accept, 1 = Reject or Error
	2	Upper Function Status of Line Gauge 16	0 = Accept, 1 = Reject or Error
	3	Lower Function Status of Line Gauge 16	0 = Accept, 1 = Reject or Error
	4	Upper Function Status of Line Gauge 17	0 = Accept, 1 = Reject or Error
	5	Lower Function Status of Line Gauge 17	0 = Accept, 1 = Reject or Error
	6	Upper Function Status of Line Gauge 18	0 = Accept, 1 = Reject or Error
	7	Lower Function Status of Line Gauge 18	0 = Accept, 1 = Reject or Error
	8	Upper Function Status of Line Gauge 19	0 = Accept, 1 = Reject or Error
	9	Lower Function Status of Line Gauge 19	0 = Accept, 1 = Reject or Error
	10	Upper Function Status of Line Gauge 20	0 = Accept, 1 = Reject or Error
	11	Lower Function Status of Line Gauge 20	0 = Accept, 1 = Reject or Error
	12	Upper Function Status of Line Gauge 21	0 = Accept, 1 = Reject or Error
	13	Lower Function Status of Line Gauge 21	0 = Accept, 1 = Reject or Error
	14	Upper Function Status of Line Gauge 22	0 = Accept, 1 = Reject or Error
	15	Lower Function Status of Line Gauge 22	0 = Accept, 1 = Reject or Error
5	0-15	Reserved for Future Expansion	Zero
6	0-15	Brightness Probe Luminance Level	Gray Scale Value of 0 to 255*
7	0-15	Window 1 Pixel Count	Value of 0 to 61,696*
8	0-15	Window 2 Pixel Count	Value of 0 to 61,696*
9	0-15	Window 3 Pixel Count	Value of 0 to 61,696*
10	0-15	Window 4 Pixel Count	Value of 0 to 61,696*
11	0-15	Upper Function Result of X-Float Gauge	Value of 0 to 255*
12	0-15	Lower Function Result of X-Float Gauge	Value of 0 to 255*
13	0-15	Upper Function Result of Y-Float Gauge	Value of 11 to 255*
14	0-15	Lower Function Result of Y-Float Gauge	Value of 11 to 255*
15	0-15	Upper Function Result of Line Gauge 1	Value of 0 to 255*
16	0-15	Lower Function Result of Line Gauge 1	Value of 0 to 255*
17	0-15	Upper Function Result of Line Gauge 2	Value of 0 to 255*
18	0-15	Lower Function Result of Line Gauge 2	Value of 0 to 255*
19	0-15	Upper Function Result of Line Gauge 3	Value of 0 to 255*
20	0-15	Lower Function Result of Line Gauge 3	Value of 0 to 255*
21	0-15	Upper Function Result of Line Gauge 4	Value of 0 to 255*
22	0-15	Lower Function Result of Line Gauge 4	Value of 0 to 255*
23	0-15	Upper Function Result of Line Gauge 5	Value of 0 to 255*

RESULTS BLOCK 1 OF 2 (Block Length of 59 Words)			
WORD (16 Bits)	BIT (Decimal)	FUNCTION	VALUES (The "*" Indicates Values Affected by the Binary/BCD Discrete Bit Setting)
24	0-15	Lower Function Result of Line Gauge 5	Value of 0 to 255*
25	0-15	Upper Function Result of Line Gauge 6	Value of 0 to 255*
26	0-15	Lower Function Result of Line Gauge 6	Value of 0 to 255*
27	0-15	Upper Function Result of Line Gauge 7	Value of 0 to 255*
28	0-15	Lower Function Result of Line Gauge 7	Value of 0 to 255*
29	0-15	Upper Function Result of Line Gauge 8	Value of 0 to 255*
30	0-15	Lower Function Result of Line Gauge 8	Value of 0 to 255*
31	0-15	Upper Function Result of Line Gauge 9	Value of 0 to 255*
32	0-15	Lower Function Result of Line Gauge 9	Value of 0 to 255*
33	0-15	Upper Function Result of Line Gauge 10	Value of 0 to 255*
34	0-15	Lower Function Result of Line Gauge 10	Value of 0 to 255*
35	0-15	Upper Function Result of Line Gauge 11	Value of 0 to 255*
36	0-15	Lower Function Result of Line Gauge 11	Value of 0 to 255*
37	0-15	Upper Function Result of Line Gauge 12	Value of 0 to 255*
38	0-15	Lower Function Result of Line Gauge 12	Value of 0 to 255*
39	0-15	Upper Function Result of Line Gauge 13	Value of 0 to 255*
40	0-15	Lower Function Result of Line Gauge 13	Value of 0 to 255*
41	0-15	Upper Function Result of Line Gauge 14	Value of 0 to 255*
42	0-15	Lower Function Result of Line Gauge 14	Value of 0 to 255*
43	0-15	Upper Function Result of Line Gauge 15	Value of 0 to 255*
44	0-15	Lower Function Result of Line Gauge 15	Value of 0 to 255*
45	0-15	Upper Function Result of Line Gauge 16	Value of 0 to 255*
46	0-15	Lower Function Result of Line Gauge 16	Value of 0 to 255*
47	0-15	Upper Function Result of Line Gauge 17	Value of 0 to 255*
48	0-15	Lower Function Result of Line Gauge 17	Value of 0 to 255*
49	0-15	Upper Function Result of Line Gauge 18	Value of 0 to 255*
50	0-15	Lower Function Result of Line Gauge 18	Value of 0 to 255*
51	0-15	Upper Function Result of Line Gauge 19	Value of 0 to 255*
52	0-15	Lower Function Result of Line Gauge 19	Value of 0 to 255*

RESULTS BLOCK 1 OF 2 (Block Length of 59 Words)			
WORD (16 Bits)	**BIT** (Decimal)	**FUNCTION**	**VALUES** (The "*" Indicates Values Affected by the Binary/BCD Discrete Bit Setting)
53	0-15	Upper Function Result of Line Gauge 20	Value of 0 to 255*
54	0-15	Lower Function Result of Line Gauge 20	Value of 0 to 255*
55	0-15	Upper Function Result of Line Gauge 21	Value of 0 to 255*
56	0-15	Lower Function Result of Line Gauge 21	Value of 0 to 255*
57	0-15	Upper Function Result of Line Gauge 22	Value of 0 to 255*
58	0-15	Lower Function Result of Line Gauge 22	Value of 0 to 255*
59	0-15	Reserved for Future Expansion	Zero

RESULTS BLOCK 2 OF 2 (Block Length of 13 Words)			
WORD (16 Bits)	**BIT** (Decimal)	**FUNCTION**	**VALUES** (The "*" Indicates Values Affected by the Binary/BCD Discrete Bit Setting)
1	0	Math Tool 1 Status	0 = Accept, 1 = Reject or Error
	1	Math Tool 2 Status	0 = Accept, 1 = Reject or Error
	2	Math Tool 3 Status	0 = Accept, 1 = Reject or Error
	3	Math Tool 4 Status	0 = Accept, 1 = Reject or Error
	4	Math Tool 5 Status	0 = Accept, 1 = Reject or Error
	5	Math Tool 6 Status	0 = Accept, 1 = Reject or Error
	6	Math Tool 7 Status	0 = Accept, 1 = Reject or Error
	7	Math Tool 8 Status	0 = Accept, 1 = Reject or Error
	8	Math Tool 9 Status	0 = Accept, 1 = Reject or Error
	9	Math Tool 10 Status	0 = Accept, 1 = Reject or Error
	10	Math Tool 11 Status	0 = Accept, 1 = Reject or Error
	11	Math Tool 12 Status	0 = Accept, 1 = Reject or Error
	12-15	Reserved for Future Expansion	Zero
2	0-15	Math Tool 1 Result	Value of 0 to 65, 535*
3	0-15	Math Tool 2 Result	Value of 0 to 65, 535*
4	0-15	Math Tool 3 Result	Value of 0 to 65, 535*
5	0-15	Math Tool 4 Result	Value of 0 to 65, 535*
6	0-15	Math Tool 5 Result	Value of 0 to 65, 535*
7	0-15	Math Tool 6 Result	Value of 0 to 65, 535*
8	0-15	Math Tool 7 Result	Value of 0 to 65, 535*
9	0-15	Math Tool 8 Result	Value of 0 to 65, 535*
10	0-15	Math Tool 9 Result	Value of 0 to 65, 535*
11	0-15	Math Tool 10 Result	Value of 0 to 65, 535*
12	0-15	Math Tool 11 Result	Value of 0 to 65, 535*
13	0-15	Math Tool 12 Result	Value of 0 to 65, 535*

(This page intentionally left blank)

Appendix D Statistics Blocks

STATISTICS BLOCK 1 OF 4 (Block Length of 55 Words)			
WORD (16 Bits)	BIT (Decimal)	FUNCTION	VALUES
1	0-15	Processed Triggers	Value of 0 to 65,535
2	0-15	Failed Inspections	Value of 0 to 65,535
3	0-15	Failed Probes	Value of 0 to 65,535
4	0-15	Window 1 Failures	Value of 0 to 65,535
5	0-15	Window 2 Failures	Value of 0 to 65,535
6	0-15	Window 3 Failures	Value of 0 to 65,535
7	0-15	Window 4 Failures	Value of 0 to 65,535
8	0-15	Float gage X (upper function) Failures	Value of 0 to 65,535
9	0-15	Float gage X (lower function) Failures	Value of 0 to 65,535
10	0-15	Float gage Y (upper function) Failures	Value of 0 to 65,535
11	0-15	Float gage Y (lower function) Failures	Value of 0 to 65,535
12	0-15	Line gage 1 (upper function) Failures	Value of 0 to 65,535
13	0-15	Line gage 1 (lower function) Failures	Value of 0 to 65,535
14	0-15	Line gage 2 (upper function) Failures	Value of 0 to 65,535
15	0-15	Line gage 2 (lower function) Failures	Value of 0 to 65,535
16	0-15	Line gage 3 (upper function) Failures	Value of 0 to 65,535
17	0-15	Line gage 3 (lower function) Failures	Value of 0 to 65,535
18	0-15	Line gage 4 (upper function) Failures	Value of 0 to 65,535
19	0-15	Line gage 4 (lower function) Failures	Value of 0 to 65,535
20	0-15	Line gage 5 (upper function) Failures	Value of 0 to 65,535
21	0-15	Line gage 5 (lower function) Failures	Value of 0 to 65,535
22	0-15	Line gage 6 (upper function) Failures	Value of 0 to 65,535
23	0-15	Line gage 6 (lower function) Failures	Value of 0 to 65,535
24	0-15	Line gage 7 (upper function) Failures	Value of 0 to 65,535
25	0-15	Line gage 7 (lower function) Failures	Value of 0 to 65,535
26	0-15	Line gage 8 (upper function) Failures	Value of 0 to 65,535
27	0-15	Line gage 8 (lower function) Failures	Value of 0 to 65,535

STATISTICS BLOCK 1 OF 4 (Block Length of 55 Words)			
WORD (16 Bits)	BIT (Decimal)	FUNCTION	VALUES
28	0-15	Line gage 9 (upper function) Failures	Value of 0 to 65,535
29	0-15	Line gage 9 (lower function) Failures	Value of 0 to 65,535
30	0-15	Line gage 10 (upper function) Failures	Value of 0 to 65,535
31	0-15	Line gage 10 (lower function) Failures	Value of 0 to 65,535
32	0-15	Line gage 11 (upper function) Failures	Value of 0 to 65,535
33	0-15	Line gage 11 (lower function) Failures	Value of 0 to 65,535
34	0-15	Line gage 12 (upper function) Failures	Value of 0 to 65,535
35	0-15	Line gage 12 (lower function) Failures	Value of 0 to 65,535
36	0-15	Line gage 13 (upper function) Failures	Value of 0 to 65,535
37	0-15	Line gage 13 (lower function) Failures	Value of 0 to 65,535
38	0-15	Line gage 14 (upper function) Failures	Value of 0 to 65,535
39	0-15	Line gage 14 (lower function) Failures	Value of 0 to 65,535
40	0-15	Line gage 15 (upper function) Failures	Value of 0 to 65,535
41	0-15	Line gage 15 (lower function) Failures	Value of 0 to 65,535
42	0-15	Line gage 16 (upper function) Failures	Value of 0 to 65,535
43	0-15	Line gage 16 (lower function) Failures	Value of 0 to 65,535
44	0-15	Line gage 17 (upper function) Failures	Value of 0 to 65,535
45	0-15	Line gage 17 (lower function) Failures	Value of 0 to 65,535
46	0-15	Line gage 18 (upper function) Failures	Value of 0 to 65,535
47	0-15	Line gage 18 (lower function) Failures	Value of 0 to 65,535
48	0-15	Line gage 19 (upper function) Failures	Value of 0 to 65,535
49	0-15	Line gage 19 (lower function) Failures	Value of 0 to 65,535
50	0-15	Line gage 20 (upper function) Failures	Value of 0 to 65,535
51	0-15	Line gage 20 (lower function) Failures	Value of 0 to 65,535
52	0-15	Line gage 21 (upper function) Failures	Value of 0 to 65,535
53	0-15	Line gage 21 (lower function) Failures	Value of 0 to 65,535
54	0-15	Line gage 22 (upper function) Failures	Value of 0 to 65,535
55	0-15	Line gage 22 (lower function) Failures	Value of 0 to 65,535

STATISTICS BLOCK 2 OF 4 (Block Length of 12 Words)			
WORD (16 Bits)	**BIT** (Decimal)	**FUNCTION**	**VALUES**
1	0-15	Math Tool 1 Failures	Value of 0 to 65, 535
2	0-15	Math Tool 2 Failures	Value of 0 to 65, 535
3	0-15	Math Tool 3 Failures	Value of 0 to 65, 535
4	0-15	Math Tool 4 Failures	Value of 0 to 65, 535
5	0-15	Math Tool 5 Failures	Value of 0 to 65, 535
6	0-15	Math Tool 6 Failures	Value of 0 to 65, 535
7	0-15	Math Tool 7 Failures	Value of 0 to 65, 535
8	0-15	Math Tool 8 Failures	Value of 0 to 65, 535
9	0-15	Math Tool 9 Failures	Value of 0 to 65, 535
10	0-15	Math Tool 10 Failures	Value of 0 to 65, 535
11	0-15	Math Tool 11 Failures	Value of 0 to 65, 535
12	0-15	Math Tool 12 Failures	Value of 0 to 65, 535

STATISTICS BLOCK 3 OF 4 (Block Length of 62 Words)			
WORD (16 Bits)	BIT (Decimal)	FUNCTION	VALUES
1	0-7	Probe Minimum Value	Value of 0 to 255
	8-15	Probe Maximum Value	Value of 0 to 255
2	0-7	Probe Average Value	Value of 0 to 255
	8-15	Reserved for Future Expansion	Zero
3	0-15	Window 1 Minimum Value	Value of 0 to 65,535
4	0-15	Window 1 Maximum Value	Value of 0 to 65,535
5	0-15	Window 1 Average Value	Value of 0 to 65,535
6-8		Window 2 Statistics (see Words 3-5)	
9-11		Window 3 Statistics (see Words 3-5)	
12-14		Window 4 Statistics (see Words 3-5)	
15	0-7	Float gage X (upper) Minimum Value	Value of 0 to 255
	8-15	Float gage X (lower) Minimum Value	Value of 0 to 255
16	0-7	Float gage X (upper) Maximum Value	Value of 0 to 255
	8-15	Float gage X (lower) Maximum Value	Value of 0 to 255
17	0-7	Float gage X (upper) Average Value	Value of 0 to 255
	8-15	Float gage X (lower) Average Value	Value of 0 to 255
18-20		Float gage Y Statistics (see Words 15-17)	
21-23		Line gage 1 Statistics (see Words 15-17)	
24-26		Line gage 2 Statistics (see Words 15-17)	
27-29		Line gage 3 Statistics (see Words 15-17)	
30-32		Line gage 4 Statistics (see Words 15-17)	
33-35		Line gage 5 Statistics (see Words 15-17)	
36-38		Line gage 6 Statistics (see Words 15-17)	
39-41		Line gage 7 Statistics (see Words 15-17)	
42-44		Line gage 8 Statistics (see Words 15-17)	
45-47		Line gage 9 Statistics (see Words 15-17)	
48-50		Line gage 10 Statistics (see Words 15-17)	
51-53		Line gage 11 Statistics (see Words 15-17)	
54-56		Line gage 12 Statistics (see Words 15-17)	
57-59		Line gage 13 Statistics (see Words 15-17)	
60-62		Line gage 14 Statistics (see Words 15-17)	

STATISTICS BLOCK 4 OF 4 (Block Length of 60 Words)			
WORD (16 Bits)	BIT (Decimal)	FUNCTION	VALUES
1	0-7	Line gage 15 (upper) Minimum Value	Value of 0 to 255
	8-15	Line gage 15 (lower) Minimum Value	Value of 0 to 255
2	0-7	Line gage 15 (upper) Maximum Value	Value of 0 to 255
	8-15	Line gage 15 (lower) Maximum Value	Value of 0 to 255
3	0-7	Line gage 15 (upper) Average Value	Value of 0 to 255
	8-15	Line gage 15 (lower) Average Value	Value of 0 to 255
4-6		Line gage 16 Statistics (see Words 1-3)	
7-9		Line gage 17 Statistics (see Words 1-3)	
10-12		Line gage 18 Statistics (see Words 1-3)	
13-15		Line gage 19 Statistics (see Words 1-3)	
16-18		Line gage 20 Statistics (see Words 1-3)	
19-21		Line gage 21 Statistics (see Words 1-3)	
22-24		Line gage 22 Statistics (see Words 1-3)	
25	0-15	Math Tool 1 Minimum Value	Value of 0 to 65,535
26	0-15	Math Tool 1 Maximum Value	Value of 0 to 65,535
27	0-15	Math Tool 1 Average Value	Value of 0 to 65,535
28-30		Math Tool 2 Statistics (see Words 25-27)	
31-33		Math Tool 3 Statistics (see Words 25-27)	
34-36		Math Tool 4 Statistics (see Words 25-27)	
37-39		Math Tool 5 Statistics (see Words 25-27)	
40-42		Math Tool 6 Statistics (see Words 25-27)	
43-45		Math Tool 7 Statistics (see Words 25-27)	
46-48		Math Tool 8 Statistics (see Words 25-27)	
49-51		Math Tool 9 Statistics (see Words 25-27)	
52-54		Math Tool 10 Statistics (see Words 25-27)	
55-57		Math Tool 11 Statistics (see Words 25-27)	
58-60		Math Tool 12 Statistics (see Words 25-27)	

(This page intentionally left blank)

Appendix E DF1 Protocol Description

Objectives

This appendix describes the Allen-Bradley DF1 communication protocol, and the programming you need to do at the host end to communicate with the VIM2 module.

What is DF1 Protocol?

DF1 protocol is an Allen-Bradley developed software convention used for serial communications. DF1 protocol provides some handshaking and data-packing formats which allow for fast communications with integrity of the data.

DF1 protocol combines some features of subcategories D1 and F1 as described in the ANSI X3.28-1976 specification. The DF1 for the VIM2 module is half-duplex, point-to-point communications between the VIM2 module and a computer host.

Transmission Codes

DF1 protocol is a character-oriented protocol that uses the following ASCII control characters extended to eight bits by adding a zero for bit 7 (see ANSI X3.4, CCITT V.3, or ISO 646 for the standard definition of these characters).

Control Character	Hexadecimal Code
STX (Start of Text)	02
ETX (End of Text)	03
ENQ (Enquiry)	05
ACK (Acknowledge)	06
DLE (Data Link Escape)	10
NAK (Negative Acknowledge)	15

Additionally, a block check character (BCC) field is used at the end of each packet for error checking. These bytes can be any value from 00 to FF hex.

As used in the following paragraphs, a code is an indivisible sequence of one or more bytes having a specific meaning to the protocol. Indivisible means that the component bytes of a code must be sent one after another with no other bytes between them. It does not refer to the timing of the bytes.

Transmission Codes
(continued)

The protocol uses these codes:

- Control codes:
  - DLE STX
  - DLE ETX BCC
  - DLE ACK
  - DLE NAK
  - DLE ENQ
- Data codes:
  - Data (single bytes having values 00-0F and 11-FF hex)
  - DLE DLE (or 10 10, to represent the value 10 hex)

Using two DLEs in sequence to represent the value 10 hex is called "DLE stuffing." The system reads a single 10 hex as a Date Link Escape, and expects the next byte to be a control code. A valid 10 hex intended to be data would be misinterpreted as a DLE. To send 10 hex as a data value, send it twice in sequence.

We can also group codes into two classes according to their use – codes issued from a station transmitting a message, and response codes issued from a station receiving a message. By this classification, the codes are:

- Codes from station transmitting a message:
  - DLE STX - indicates the start of a message.
  - Data (00-0F and 11-FF hex) - encodes the bytes of the command.
  - DLE DLE (or 10 10) - encodes the value 10 hex in the text code.
  - DLE ETX BCC - terminates a message.
  - DLE ENQ - requests the retransmission of the last received ACK or NAK.
- Response code from station receiving a message:
  - DLE ACK - signals that the receiver has successfully received the last message sent.
  - DLE NAK - signals that the receiver did not successfully receive the last message sent.

Message Packets

A message packet starts with a DLE STX, ends with a DLE ETX BCC, and includes all data codes in between. Data codes can occur only inside a message packet.

Figure A.1 shows the format of a message packet. At the end of each message is a one byte BCC.

Figure A.1
Message Packet Format

DLE	STX	Data . . .	DLE	ETX	BCC

Block Check Character

The block check character (BCC) is a means of checking the accuracy of each message packet transmission. It is the 2's complement of the 8-bit sum (modulo-256 arithmetic sum) of all data bytes between the DLE STX and the DLE ETX BCC. It does not include any other message packet codes or response codes.

For example, if a message packet contained the data codes 8, 9, 6, 0, 2, 4, and 3, the message packet codes would be (in hex):

10	02	08 09 06 00 02 04 03	10	03	E0
DLE	STX	Data	DLE	ETX	BCC

The sum of the data bytes in this message packet is 20 hex. The BCC is the 2's complement of this sum, or EO hex. This is shown in the following binary calculation:

```
0010 0000    20 hex
1101 1111    1's complement
       +1
---------
1110 0000    2's complement (E0 hex)
```

Block Check Character *(continued)*

To transmit the data value 10 hex, you must use the data code DLE DLE. However, only one of these DLE data bytes is included in the BCC sum. For example, to transmit the values 8, 9, 6, 0, 10, 4, and 3 hex, you would use the following message codes:

Represents single data byte value of 10

10	02	08 09 06 00 10 10 04 03	10	03	D2
DLE	STX	Data	DLE	ETX	BCC

In this case, the sum of the data bytes is 2E hex because only one DLE text code is included in the BCC. So the BCC is D2 hex.

The BCC algorithm provides a medium level of data security. It cannot detect transposition of bytes during transmission of a packet. It also cannot detect the insertion or deletion of data values of zero within a packet.

Message Size

The maximum size of a message packet is 250 bytes. This does not include the "DLE STX," "DLE ETX BCC," or the first "DLE" of data "DLE DLE."

Transmitter Actions

Whenever the message source can supply a packet and the transmitter is not busy, transmitter A sends a message packet (see Figure A.2). It then starts a timeout, and waits for a response.

When transmitter A gets a DLE ACK, the message transfer is complete. After signaling the message source that the message has been sent successfully, transmitter A proceeds with the next message.

If transmitter A gets a DLE NAK, it retransmits the same message. The transmitter restarts the timeout and waits again for a response. The default retry setting is 3. Once the number of retransmissions exceeds this limit, the transmitter should notify the message source that the transmission has failed. The transmitter can then proceed with the next message.

If the timeout expires before transmitter A gets a response, it sends a DLE ENQ to request a retransmission of the last response sent. Transmitter A restarts the timeout and waits for a response. The default retry setting is 3 for the VIM2 module. If this ENQ limit is exceeded, the transmitter should notify the message source that the transmission has failed. The transmitter can then proceed with the next message.

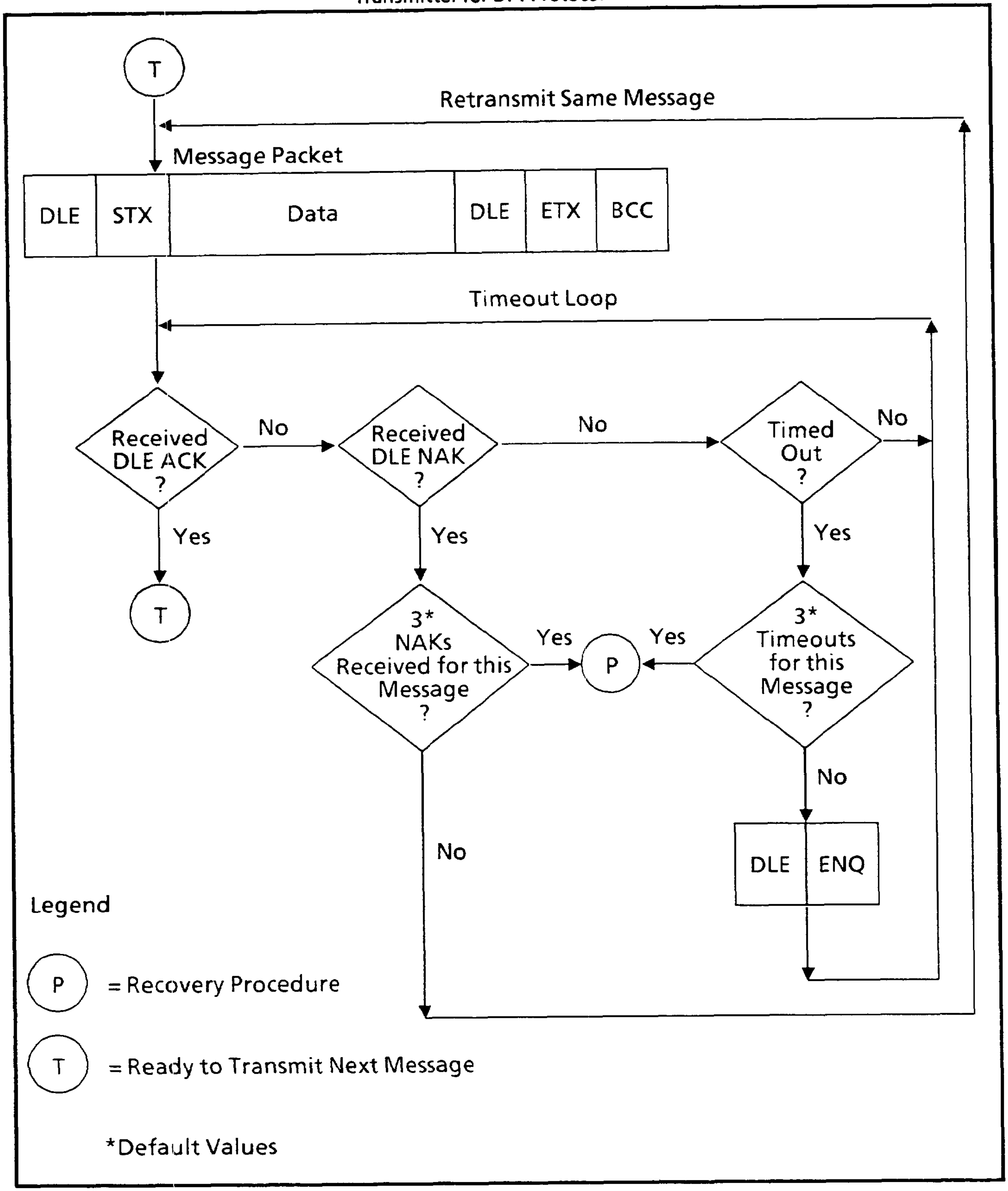
DLE ACK and DLE NAK are the only response codes defined. If the receiver gets an invalid response code, it should ignore it.

Note that the transmitter must encode a text value of 10 hex as two consecutive (indivisible) bytes, each of value 10 hex. This is necessary to distinguish the text value of 10 hex from the DLE control code of 10 hex. This technique is known as DLE stuffing. The receiver must be able to reverse this process and extract the original text value of 10 hex.

Transmitter Actions
(continued)

Figure A.2 is a flowchart which gives a simplified view of an example of software logic for implementing the transmitter.

Figure A.2
Transmitter for DF1 Protocol



Receiver Actions

Since the receiver gets input structured in a variety of formats from the physical world, it is more complex and must be capable of responding to many adverse situations. Some of the things that can conceivably happen are listed here:

- Receiver is busy and cannot process the message.
- A message can contain a parity error.
- The BCC can be invalid.
- The DLE STX or DLE ETX BCC may be missing.
- The message can be too long.
- A spurious control or text code can occur outside a message.
- A spurious control code can occur inside a message.
- Any combination of the above can occur.
- The DLE ACK response can be lost, causing the transmitter to send a duplicate copy of a message that has already passed to the message sink.

The receiver must keep a record of the last response code (DLE ACK or DLE NAK) sent. If it receives a DLE ENQ, the receiver sends this recorded response code again.

Until it receives a DLE STX or a DLE ENQ, the receiver ignores all input. On initialization, receipt of spurious control or text code outside a message, or receipt of DLE STX, the last response code is set to NAK. Receipt of ACK from last message sent will set the response to ACK. The receiver responds to a DLE ENQ input by sending its last response (ACK or NAK) and continues waiting for input. If the receiver gets a DLE STX, it resets its BCC accumulator and data buffer to zero and starts storing the data in the data buffer.

Receiver Actions
(continued)

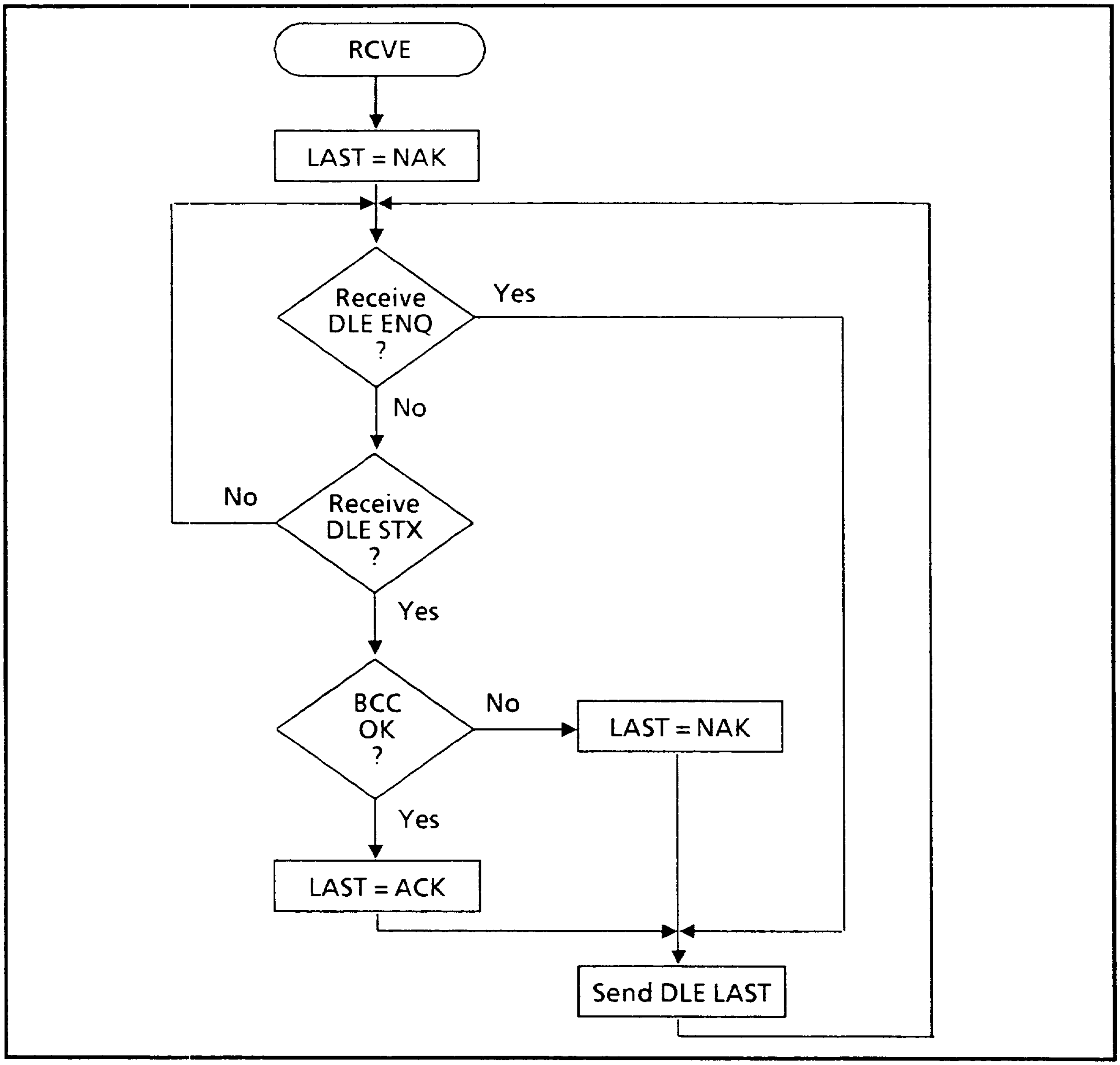
While the receiver stores all link-level data codes in the data buffer, it adds the link-level data code values to the BCC. If the data buffer overflows, or there is no buffer available, the receiver continues summing the BCC, but it discards the data.

The receiver also sets an error flag to indicate the occurrence of a parity, buffer overrun, message framing, or modem handshaking error. When the receiver gets a DLE ETX BCC, it checks the error flag, the BCC, and the message size. If any of the tests fail, the receiver sends a DLE NAK.

Figure A.3 is a flowchart which gives a simplified view of an example of software logic for implementing the receiver.

Receiver Actions
(continued)

Figure A.3
Receiver DF1 Protocol



Example If you were to connect a line monitor to the wires between stations you could observe the activates shown in Figure A.4:

Figure A.4
Message Error Recovery

Normal message

Path 1:	DLE STX (Data) DLE ETX BCC→		DLE STX (Data) DLE ETX BCC→	
Path 2:		←DLE ACK		←DLE ACK

Message with parity or BCC error and recovery

Path 1:	DLE STX (Noise) DLE ETX BCC→		DLE STX (Data) DLE ETX BCC→	
Path 2:		←DLE NAK		←DLE ACK

Message with ETX destroyed

Path 1:	DLE STX (Data) (Noise) [timeout] DLE ENQ→		DLE STX (Data) DLE ETX BCC→	
Path 2:		←DLE NAK		←DLE ACK

Good message but ACK destroyed

Path 1:	DLE STX (Data) DLE ETX BCC→		[timeout] DLE ENQ→	
Path 2:		←DL (Noise) CK		←DLE ACK

Note: A simple way to test the RS-485 link is to send the VIM2 module a DLE ENQ (enquiry). If you have the port properly connected and the VIM2 module is configured for DF1 protocol, the VIM2 module should send a DLE ACK or DLE NAK in response. If no response is provided, check your connections and VIM2 module configuration.

Appendix F Discrete Bit Addressing

Objective

This appendix describes the format for addressing the discrete bits when using a PLC with the VIM2 module. The addressing is discussed for these cases:

- Two-slot addressing
- One-slot addressing
- Half-slot addressing

The block transfer addressing is also discussed for each of the three cases listed above.

Two-slot Addressing

When two-slot addressing is used, the discrete bits are addressed as follows:

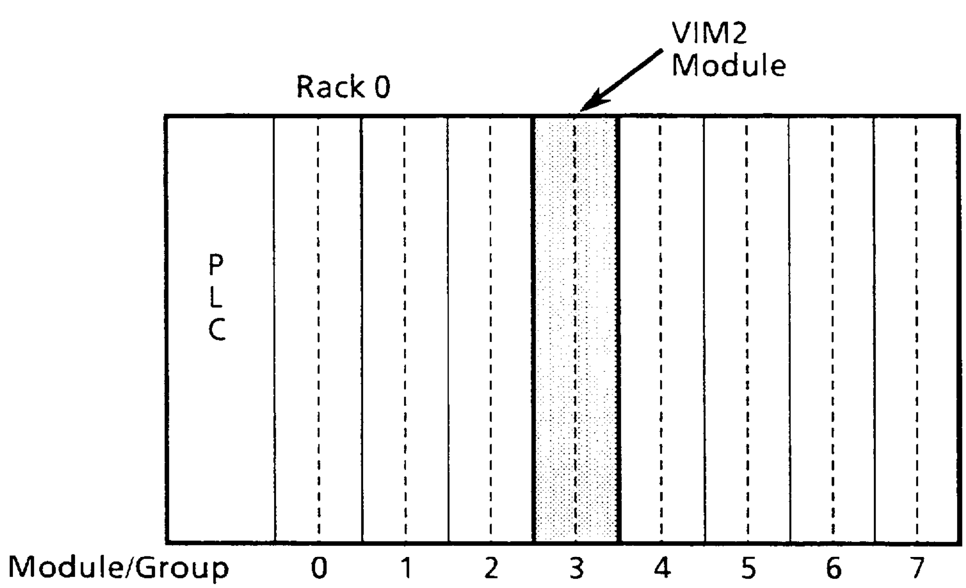
- Output (PLC to VIM2 module) – ORG / 10 to ORG /17
- Input (VIM2 module to PLC) – IRG / 10 to IRG /17

where R = rack number, and G = module/group number.

Block transfers are addressed to RG0.

For example, in the diagram below, the VIM2 module is located in rack 0, module group 3. In this case, discrete bit outputs are addressed O03 / 10 to O03 /17. Discrete bit inputs are addressed I03 / 10 to I03 /17.

Block transfers would be addressed to 030.



One-slot Addressing

When one-slot addressing is used, the discrete bits are addressed as follows:

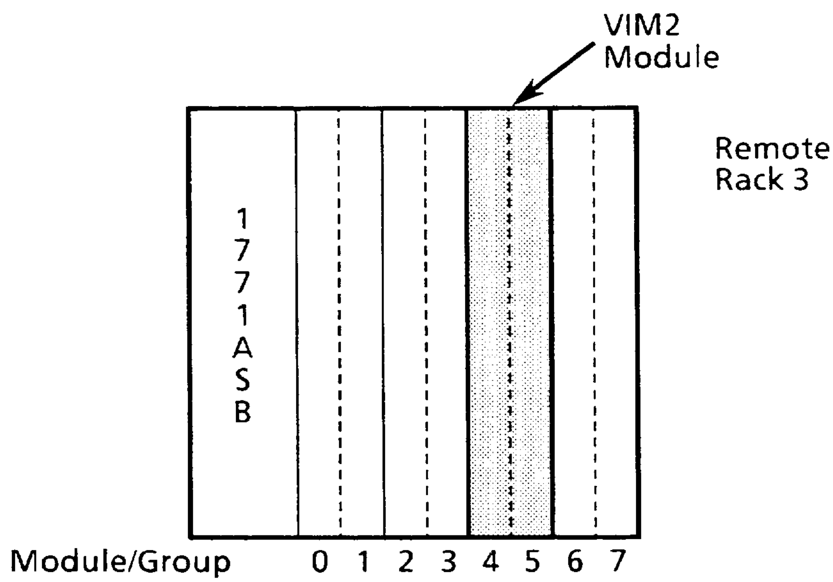
- Output (PLC to VIM2 module) – ORG / 0 to ORG /7
- Input (VIM2 module to PLC) – IRG / 0 to IRG /7

where R= rack number, and G=module group number of the *right* slot of the two occupied by the VIM2 module.

Block transfers are addressed to RG0 where R= rack number, and G=module group number of the *left* slot of the two occupied by the VIM2 module.

For example, in the diagram below, the VIM2 module is located in rack 3, in module groups 4 and 5. In this case, the discrete bit outputs are addressed O35 / 0 to O35 /7. Discrete bit inputs are addressed I35 / 0 to I35 /7.

Block transfers would be addressed to 340.



Half-slot Addressing

When half-slot addressing is used, the discrete bits are addressed as follows:

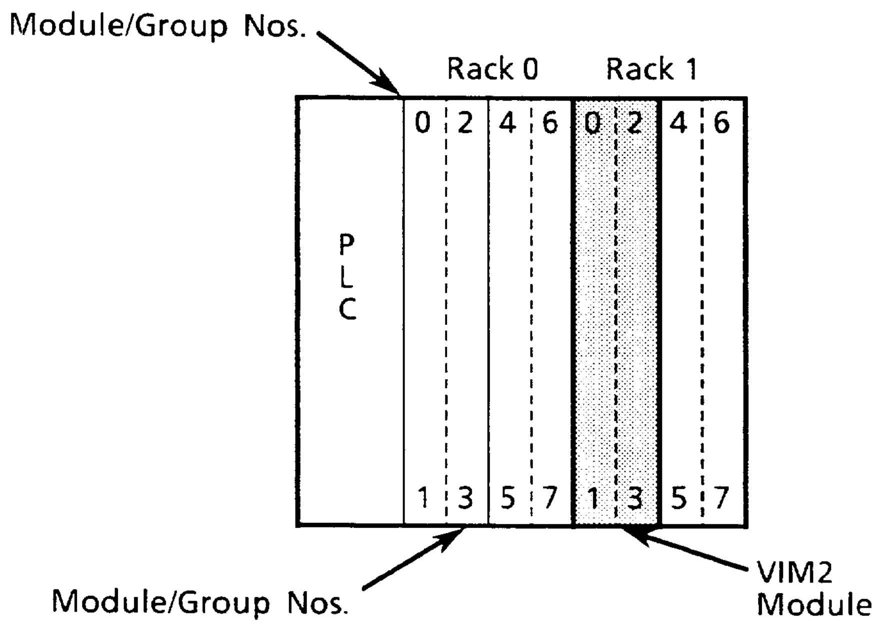
- Output (PLC to VIM2 module) – ORG / 0 to ORG /7
- Input (VIM2 module to PLC) – IRG / 0 to IRG /7

where R = rack number, and G = module group number of the *last module* of the two occupied by the VIM2 module.

Block transfers are addressed to RG0 where R = rack number, and G = module group number of the *second module* of the two occupied by the VIM2 module.

For example, in the diagram below, the VIM2 module is located in rack 1, in module groups 0 to 3. In this case, the discrete bit outputs are addressed O13 / 0 to O13 / 7. Discrete bit inputs are addressed I13 / 0 to I13 / 7.

Block transfers would be addressed to 110.



(This page intentionally left blank)

Appendix G ASCII Conversion Table

ASCII or Control Char.	Decimal Value	Hex Value	ASCII or Control Char.	Decimal Value	Hex Value	ASCII or Control Char.	Decimal Value	Hex Value	ASCII or Control Char.	Decimal Value	Hex Value
NUL	0	0	[Space]	32	20	@	64	40	‘	96	60
SOH	1	1	!	33	21	A	65	41	a	97	61
STX	2	2	"	34	22	B	66	42	b	98	62
ETX	3	3	#	35	23	C	67	43	c	99	63
EOT	4	4	$	36	24	D	68	44	d	100	64
ENQ	5	5	%	37	25	E	69	45	e	101	65
ACK	6	6	&	38	26	F	70	46	f	102	66
BEL	7	7	'	39	27	G	71	47	g	103	67
BS	8	8	(	40	28	H	72	48	h	104	68
HT	9	9	)	41	29	I	73	49	i	105	69
LF	10	A	*	42	2A	J	74	4A	j	106	6A
VT	11	B	+	43	2B	K	75	4B	k	107	6B
FF	12	C	,	44	2C	L	76	4C	l	108	6C
CR	13	D	-	45	2D	M	77	4D	m	109	6D
SO	14	E	.	46	2E	N	78	4E	n	110	6E
SI	15	F	/	47	2F	O	79	4F	o	111	6F
DLE	16	10	0	48	30	P	80	50	p	112	70
DC1	17	11	1	49	31	Q	81	51	q	113	71
DC2	18	12	2	50	32	R	82	52	r	114	72
DC3	19	13	3	51	33	S	83	53	s	115	73
DC4	20	14	4	52	34	T	84	54	t	116	74
NAK	21	15	5	53	35	U	85	55	u	117	75
SYN	22	16	6	54	36	V	86	56	v	118	76
ETB	23	17	7	55	37	W	87	57	w	119	77
CAN	24	18	8	56	38	X	88	58	x	120	78
EM	25	19	9	57	39	Y	89	59	y	121	79
SUB	26	1A	:	58	3A	Z	90	5A	z	122	7A
ESC	27	1B	;	59	3B	[	91	5B	{	123	7B
FS	28	1C	<	60	3C	\	92	5C	¦	124	7C
GS	29	1D	=	61	3D	]	93	5D	}	125	7D
RS	30	1E	>	62	3E	∧	94	5E	~	126	7E
US	31	1F	?	63	3F	_	95	5F			

Appendix H Definition of Terms

Objectives

This appendix supplies definitions to terms used in this manual which relate to vision technology in general, and to terms which relate specifically to the VIM2 module.

Definition of Terms

There are terms in this manual which are used in the machine vision industry, and others which are specific to the VIM2 system. These and other key terms are defined below:

- **Acceptance Range** – The range of values that are accepted for vision tool range tests. The acceptance range is defined by high and low range limits. The high and low range limits define the range of variation that can be tolerated above and below the nominal value. Range limits are defined by the user.

- **Archiving Configurations**– When you are configuring the VIM2 module for an inspection application, the configuration is stored in the VIM2's volatile RAM. If you wish to save the configuration, store the configuration in the VIM2's EEPROM memory. You can store, or archive, two different configurations in EEPROM.

- **Blob** – A group of contiguous (adjacent) white or black pixels along a line of pixels in an image. The line gages of the VIM2 module can make edge, center, and width measurements for blobs.

- **Block Transfer** – A block transfer is a method of communicating a "block" of data between a PLC and an I/O module. In this case, the I/O module is the VIM2 module and the block of data can be results data, configuration data, or statistics data. All block transfers are invoked by an instruction from the PLC controller.

- **Brightness Probe** – A sample area of the image used to measure light intensity or "brightness." This probe can be used to:

  - Measure the brightness of a small section of the image.

  - Detect lighting brightness changes from inspection to inspection in order to compensate for variations in brightness.

Definition of Terms
(continued)

- **Column** – A row of pixels in the vertical (Y) direction in the image or on the display screen.
- **Brightness Registration** – This feature allows you to adjust all the threshold image settings based on a permanent change in lighting brightness level in the workstage area.
- **Configuration Blocks** – Blocks of data that may be uploaded to, or downloaded from, a PLC controller, or to a computer connected through the VIM2 module's RS-485 port. The configuration blocks contains configuration information about measurement windows, line gages, the brightness probe, math tools, and other setup information.

  One of the more important aspects of the VIM2 module is that configuration data can be transferred to and from the PLC controller or other computer. As a result, configuration data can be sent to the PLC controller or other host, the VIM2 module removed and replaced, and the replacement module easily reconfigured by downloading a configuration from the host to the VIM2 module. Note that two different configurations can also be stored on board the VIM2 module in non-volatile EEPROM.
- **Constant** – A selectable numerical value used as a parameter when using the math tools.
- **Contrast** – The brightness difference between the workpiece and the background as seen in the image. Good contrast is important for reliable operation of the vision tools used in the VIM2 module.
- **Decision output** – The *Decision* output is a discrete output which indicates the accept/reject status of an inspection. This status is available through both the PLC controller (the *Decision* bit) and through the swingarm (the *Decision* output).
- **Depth of Field** – The range in which objects focus clearly. It is measured from the distance beyond the ideal focal point to the distance in front of it in which objects remain in focus.
- **EEPROM Configuration** – One of two configurations which are stored in the VIM2 module's non-volatile electrically eraseable programmable memory (EEPROM).
- **Field of View** – The inspected area that is seen through a lens or optical instrument by the sensor grid of the video camera.

Definition of Terms
(continued)

- **Icon** – A symbolic, pictorial representation of a feature or a menu (group of icons). "Picking" an icon with the light pen enables/disables or selects a feature or parameter, or selects the depicted icon menu. The icons and icon menus are introduced in Chapter 6. The complete icon menu branching map is illustrated in Appendix A.

- **Host** – A PLC, or a computer connected to the VIM2 module through the RS-485 port, which communicates with the VIM2 module. A host device can collect fault, results, statistics, or configuration information from the VIM2 module (see also *Master Host*).

- **Field (Video)** – A single scan of the video camera image. The camera produces a steady stream of video fields, each consisting of a series of scan lines (rasters).

- **Gray Level** – A measure of relative brightness from black, through many increments of gray, to white.

- **Light Pen** – The input device used to interact with the VIM2 module. The light pen has a retractable tip, and is used to select or "pick" icons and menus on the video monitor.

- **Line Gage** – Line gages represent one type of inspection tool provided by the VIM2 module. A line gage inspects a set of horizontally or vertically aligned pixels (found in a row or column).

- **Math Tools** – User-defined formulas which combine or adjust the results of inspection tools.

- **Master Host** – A host device (PLC, or a computer connected to the VIM2 module through the RS-485 port), which is designated as master host in the VIM2 module's configuration. A *master host*, as distinguished from a *host*, can configure the VIM2 module, and trigger inspections, as well as collect fault, results, statistics, or configuration information (see also *Host*).

- **Operator** – A symbol (such as +) or term (such as *AND*) that represents the operation performed by a math tool upon one or more operands.

- **Operand** – A quantity or value (such as a constant, or the result of a line gage function or window), upon which the operator in a math tool acts.

Definition of Terms
(continued)

- **Pick** – The action of depressing the tip of the light pen against a displayed icon or value on the monitor screen to "select" the function or menu represented by that icon.
- **Pixel** – One picture element (or dot) in a video image. The video image is a matrix of pixels.
- **RAM Configuration** – The "working" configuration (the configuration by which the VIM2 module operates), which is stored in the VIM2 module's volatile random access memory (RAM).
- **Results** – The value generated by a vision tool as a consequence of the tool inspecting a threshold image, or in the case of a math tool, the value generated as a consequence of the operator acting upon the operands.
- **Results Blocks** – Data tables stored by the VIM2 module which can be transferred to the PLC or other host. The results blocks contain information indicating the accept/reject status of acceptance range tests for the brightness probe, line gages, windows, and math tools. The actual probe luminance gray value, pixel counts for each window, line gage results for each line gage, and math tool results are communicated through the results blocks. The VIM2 module generates two results blocks for each inspection cycle.
- **Row** – A line of pixels across the image in the horizontal (X) direction.
- **Shift Registration** – This feature allows you to adjust the inspection tool positions based on a permanent change in the nominal workpiece position in the workstage area.
- **Standoff** – The distance between the inspected workpiece and the camera body.
- **Statistics Blocks** – Data tables stored by the VIM2 module which can be transferred to the PLC or other host. The statistics blocks contain accumulated statistical data for the brightness probe, line gages, windows, and math tools; the statistics are gathered while the VIM2 module operates in "learn" mode. The statistics include the triggers processed, inspections failed per tool/ probe, and tool/probe minimum, maximum, and average values. Statistics are collected for measurements which fall within the respective acceptance range only (statistics are not kept for failed tools). The VIM2 module generates four statistics blocks for each inspection cycle.

Definition of Terms
(continued)

- **Swingarm** – A screw terminal connector installed on the front panel of many 1771-I/O modules, including the VIM2 module. It's used to connect wires to the module.
- **Threshold** – A gray level used to transform a gray-scale video image into a binary image. Pixels whiter than the threshold are converted to white (1), values darker or equal to the Threshold are converted to black (0).
- **Vision Tool** – The VIM2 module vision tools include the line gages and windows. Vision tools are used to take measurements and generate accept/reject decisions.
- **Window** – Windows are shapes which define localized image areas to be inspected in measurement operations. The user defines the window size, shape, and location. The vision operation used in VIM2 module windows is area measurement by pixel counting.
- **Workpiece** – The part or object inspected by the VIM2 module.
- **Workstage** – The specific area where the vision inspections for an application are to take place.

(This page intentionally left blank)

Appendix J Specifications

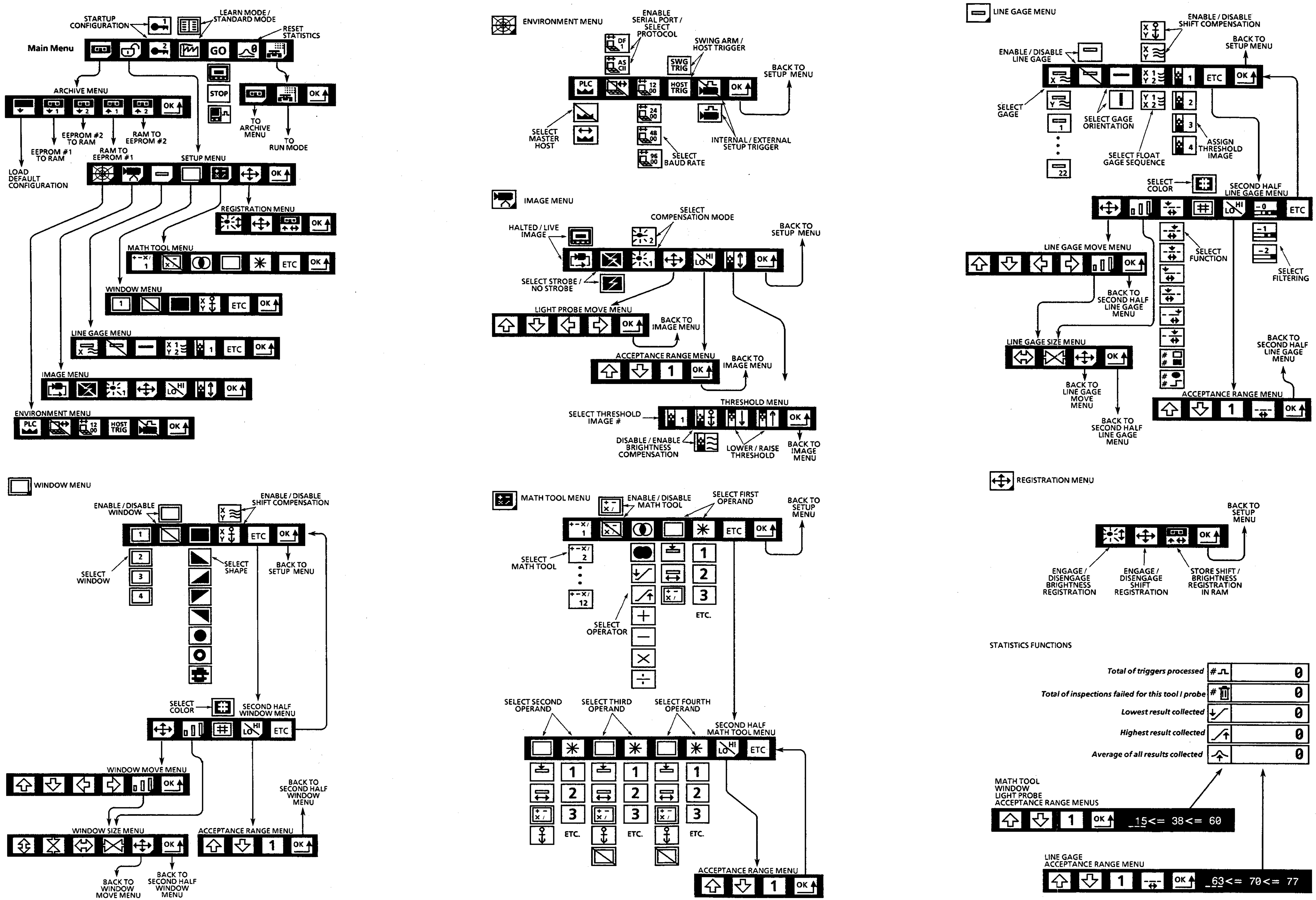
Specifications

I/O Rack Installation	2-slot module installs in any 1771 series B I/O chassis.
Cameras	Catalog No. 2801-YB, -YC, -YD
Backplane Current	3 A at 5 volts
Swingarm Trigger Input	
Trigger input (isolated)	3.3 VDC to 32 VDC
Minimum trigger pulse width	50 µs
Decision and Busy Swingarm Outputs	
Discrete outputs	3 to 32 VDC
Maximum current	1 A
Isolation	1500 VAC RMS from other I/O
Connection	Polarity insensitive, shared common connection.
Swingarm Strobe Output	
Strobe trigger output	5 volt active
Connection	Non-isolated, shares input power common.
Compatibility	Compatible with A-B machine vision strobe light sources.
Serial Communications	RS-485, half-duplex ASCII or DF1 protocol.
Environment	
Ambient temperature	
Operating	0-55° C (32-131° F)
Storage	-40 to 85° C (-40 to 185° F)
Relative humidity	5-95%, non-condensing

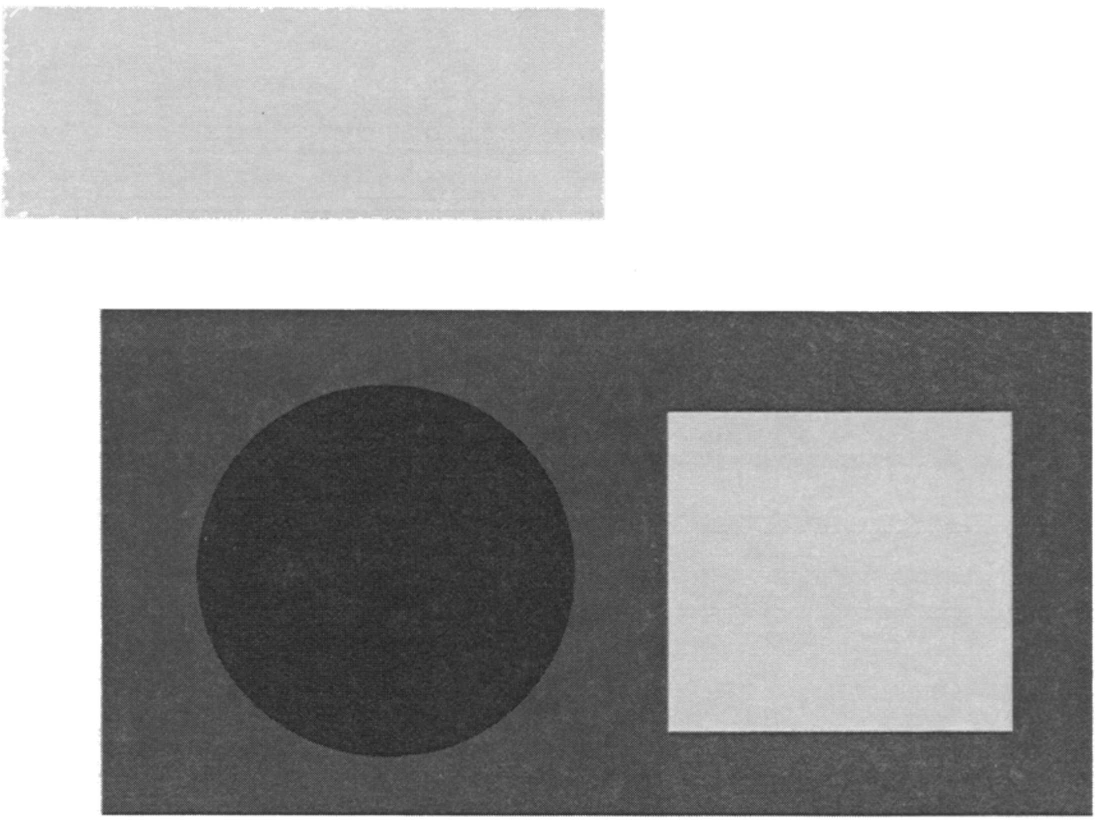
Index

Section *Page*

Menu Branching Diagram for the 2803-VIM2



TARGET PATTERN



Rockwell Automation

Rockwell Automation helps its customers receive a superior return on their investment by bringing together leading brands in industrial automation, creating a broad spectrum of easy-to-integrate products. These are supported by local technical resources available worldwide, a global network of system solutions providers, and the advanced technology resources of Rockwell.

Worldwide representation.

Argentina • Australia • Austria • Bahrain • Belgium • Bolivia • Brazil • Bulgaria • Canada • Chile • China, People's Republic of • Colombia • Costa Rica • Croatia • Cyprus Czech Republic • Denmark • Dominican Republic • Ecuador • Egypt • El Salvador • Finland • France • Germany • Ghana • Greece • Guatemala • Honduras • Hong Kong Hungary • Iceland • India • Indonesia • Iran • Ireland • Israel • Italy • Jamaica • Japan • Jordan • Korea • Kuwait • Lebanon • Macau • Malaysia • Malta • Mexico • Morocco The Netherlands • New Zealand • Nigeria • Norway • Oman • Pakistan • Panama • Peru • Philippines • Poland • Portugal • Puerto Rico • Qatar • Romania • Russia • Saudi Arabia • Singapore • Slovakia • Slovenia • South Africa, Republic of • Spain • Sweden • Switzerland • Taiwan • Thailand • Trinidad • Tunisia • Turkey • United Arab Emirates United Kingdom • United States • Uruguay • Venezuela

Rockwell Automation Headquarters, 1201 South Second Street, Milwaukee, WI 53204-2496 USA, Tel: (1) 414 382-2000 Fax: (1) 414 382-4444
Rockwell Automation European Headquarters, Avenue Hermann Debroux, 46, 1160 Brussels, Belgium, Tel: (32) 2 663 06 00, Fax: (32) 2 663 06 40
Rockwell Automation Asia Pacific Headquarters, 27/F Citicorp Centre, 18 Whitfield Road, Causeway Bay, Hong Kong, Tel: (852) 2887 4788, Fax: (852) 2508 1846
World Wide Web: http://www.ab.com

Catalog No. 2803–ND001 – Series A

40062-201-01(A)