

---

# Sherlock Embedded Software Manual

405-00034-00

Version 1.3.1

28 May 2014



# Sherlock Embedded Software Manual

Document number 405-00034-00

Software Version 1.3.1; May 2014

Copyright© 2014 Teledyne DALSA, Inc.

All Right Reserved.

All copyrights in this manual and the hardware and software described in it, are the exclusive property of Teledyne DALSA, Inc. and its licensors. Claim of copyright does not imply waiver of Teledyne DALSA, inc. or its licensors other rights in the work. See the following Notice of Proprietary Rights.

## NOTICE OF PROPRIETARY RIGHTS

This manual and the related hardware and software are confidential trade secrets and the property of Teledyne DALSA, Inc. and its licensors. Use, examination, reproduction, copying, transfer and/or disclosure to others of all or any part of this manual and the related documentation is prohibited except with the express written consent of Teledyne DALSA, Inc.

The information in this document is subject to change without notice. Teledyne DALSA, Inc. makes no representations or warranties with respect to the content of this manual and specifically disclaims any implied warranties of merchantability or fitness for a particular purpose. Teledyne DALSA, Inc. assumes no responsibility for errors or omissions in this document.

BOA, iNspct, Sherlock, and the Teledyne DALSA logo are trademarks of Teledyne DALSA, Inc. All other trademarks are the property of their respective owners.

Do not attempt to reverse engineer the software. Do not attempt to use the Emulator in a production environment without obtaining a license. Your Sherlock Embedded software license is only valid for use with a BOA Pro smart camera.

Teledyne DALSA Industrial Products

**Information:** [info.ipd@teledynedalsa.com](mailto:info.ipd@teledynedalsa.com)

**Support:** [support.ipd@teledynedalsa.com](mailto:support.ipd@teledynedalsa.com)

**Web:** <http://www.teledynedalsa.com/ipd>

700 Technology Park Drive

Billerica, MA USA 01821

**Tel:** 1.978.670.2002 **FAX:** 1.978.670.2010

## Table of Contents

Sherlock Embedded .....	1
Software Manual.....	1
BOA Pro System Setup.....	5
Sherlock Embedded Software.....	5
Client Software Installation.....	5
Getting Started: BOA Address.....	5
Addressing Options.....	5
Change the Address Using iDiscover .....	6
Change the Address Using a Web Browser .....	7
Windows Configuration .....	8
Password Protection.....	9
Web Interface Password Protection .....	9
Client Interface Password Protection .....	9
Upgrading the Sherlock Embedded Software.....	10
Upgrading the BOA Pro Smart Camera’s Firmware.....	10
Upgrading the Sherlock Embedded Client Software .....	10
BOA Pro Web Server .....	11
Sherlock Embedded Software Design.....	13
User Interface Overview .....	14
Getting Started.....	14
How Sherlock Works.....	14
Running a Program .....	14
Programming Sherlock.....	14
On-Line Help .....	14
Sherlock Embedded Client Application Windows.....	15
Main Window .....	15
Image Window.....	16
Program Window.....	17
System Window.....	18
Properties Window .....	18
Instructions Window.....	26
Watch Window .....	26
Monitor Window .....	26
Digital IO Monitor Window.....	27
Saving Exporting and Importing Files.....	28

## Sherlock Embedded Software Manual

Save Programs.....	28
Export Programs .....	29
Import Programs.....	30
Sherlock Embedded Emulator .....	31
Emulator images .....	32
Emulator IO.....	32
Saving Emulator Programs.....	32
Exporting Emulator Programs.....	32
JavaScript in Sherlock Embedded .....	33
Sherlock arrays in JavaScript.....	34
Operators .....	35
Comparisons .....	35
Arithmetic operations.....	35
Logical (Boolean) operations .....	36
Other Topics and Documents .....	36

## BOA Pro System Setup

The hardware installation is covered in the *BOA Pro Installation Manual* available on the BOA Pro CD. All mechanical and electrical specifications, wiring diagrams and cabling information appear in that document. Mechanical drawings are also included on the BOA Pro CD.

## Sherlock Embedded Software

The Sherlock Embedded Software consists of two parts: the **Server** and the **Client**. The Server software is already installed and licensed on the BOA Pro Smart Camera. You must install the Client software on any PC that will access the Sherlock programming interface. You do not need to install the Client software on a PC that will monitor the BOA Pro Smart Camera using a Web Browser.

### Client Software Installation

The Sherlock Embedded Client is a 32-bit Windows application that may be installed on Windows XP and Windows 7. Install the Client on a PC that will be used to program the Sherlock Embedded inspection or “investigation” program.

Insert the BOA Pro CD-ROM into your PC’s CD or DVD drive. The installation utility should run automatically on Windows XP and Windows 7 32-bit installations. Click “Install Software” to begin the installation. The default selection is Sherlock Embedded Client. Click “Start Install” The installation utility allows you to change the destination directory.

Some security settings in Windows XP or Windows 7 can prevent the CD from automatically running the installation utility. In **Windows XP**, you can navigate to the root directory of the CD and double-click on setup.exe. In **Windows 7**, it is recommended you navigate to the “\SherlockEmbeddedClient” directory, right-click on “SherlockBOA.exe” and click on “Run as administrator” in the pop-up menu.

Windows 7 64 bit installations install 32-bit programs under C:\Program Files (x86) not under C:\Program Files. Windows 7 does not tell you it is doing this. Therefore, your programs may not be where you expect them to be.

### Getting Started: BOA Address

Start Sherlock Embedded Client on the PC. The Connect window should display all compatible connected BOA Pro cameras running the Sherlock Embedded software. If your BOA Pro is not displayed, the address is not compatible. The Factory default address is 192.168.0.100. There are a few options for setting the BOA Pro to an address that is compatible with your PC. You can use the iDiscover utility to set the address. You can use Web browser to access the BOA at its factory address and change it.

### Addressing Options

Most industrial factory floor applications use persistent or static IP Addressing, because the addresses are fixed, and never change. In such a “machine network” using persistent addresses, the BOA Pro is usually accessed by its IP Address, but it can also be accessed using the Network Name.

Most office or business networks use DHCP addressing, with a network host computer configured to assign addresses, and “lease” the address for a fixed period (usually a business day). In an office network with DHCP addressing, the BOA Pro is accessed by its network name. You can still use a persistent address on an office network, but you must get an “allowable” address from the system administrator.

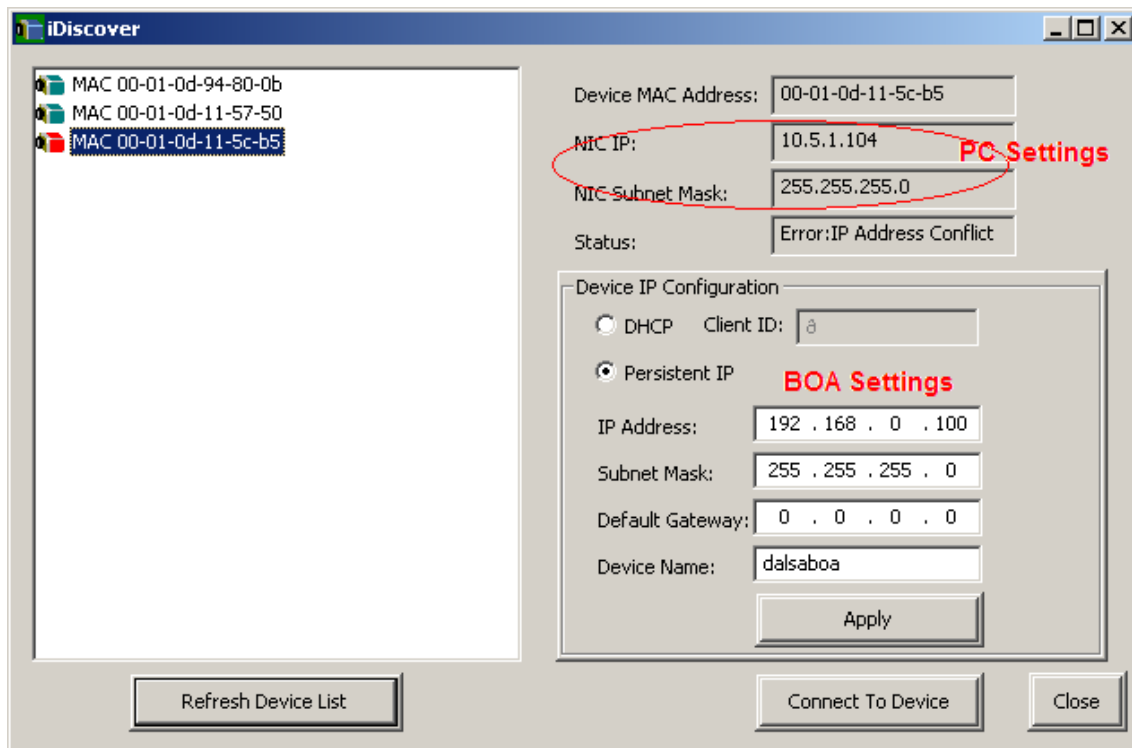
A PC or laptop that is not connected to a network or other peripherals can use the temporary address (example 192.168.0.101) compatible with the BOA Pro. However, this address may make the PC or Laptop incompatible with your router and other peripherals (like printers), unless they use different connections, like a wireless router or USB connection. A single PC or Laptop is not usually configured as a “DHCP Server” to provide DHCP Addresses to connected devices.

### Change the Address Using iDiscover

The easiest way change your BOA Pro address is to use the iDiscover utility. If you have already installed the Sherlock Embedded Client software, this utility is on your PC. (This utility is also installed by connecting to BOA cameras running the iInspect Express software. iInspect Express puts a copy of iDiscover in the system32 directory, and the command “iDiscover” can be typed into the Run command dialog.) You can also run iDiscover from the BOA Pro CD-ROM. A copy of iDiscover is included on the BOA Pro CD-ROM, in the “Utilities” directory.

1. To start iDiscover, use Windows Explorer to navigate to the iDiscover utility, or use the Run command dialog, and the Browse button to navigate to the iDiscover utility. The file name is “iDiscover.exe” The default location is: C:\Program Files\Teledyne DALSA\Sherlock Embedded\Utilities

iDiscover shows all the currently connected BOA and BOA Pro smart cameras, at all addresses, but it does not tell you what software they are running (iInspect Express or Sherlock Embedded). iDiscover also indicates what cameras are at incompatible addresses. The **red** camera icon indicates incompatible cameras. The **blue** camera icon indicates compatible cameras. You can use the iDiscover utility to select an incompatible address camera, and change the address.



2. Click on a camera icon to populate the address fields.
3. Enter a new IP Address or “Device Name”. Change the Subnet Mask if necessary. In most machine networks that use addresses like 10.1.x.x the subnet mask should be **255.255.192.0**. Using the default 255.255.255.0 can cause other devices to disappear.

4. Click “Apply” to set the new address or name. You should reboot (power cycle) the BOA Pro, to make sure the new name or address is fully applied.

Do not use the iDiscover utility to connect to a BOA Pro camera running the Sherlock Embedded software. iDiscover only supports connecting to BOA cameras running the iInspect Express software. Clicking on “Connect to Device” give a “Page cannot be found” message if the selected camera is running the Sherlock Embedded software.

### Change the Address Using a Web Browser

You do not need to load the Client software for this method. To change the address using a Web Browser, you need to know the current address of the BOA Pro and you must: **A.** Temporarily change your PC to a compatible address. The factory default address for the BOA Pro is 192.168.0.100. **B.** Access the BOA Pro and change its address to one that is compatible with your network. **C.** Reboot (power cycle) the BOA Pro. **D.** Change your PC address back.

You may need administrator privileges on your PC to access the BOA Pro. It may be necessary to customize the security settings on your web browser.

The following web browsers have been tested with Sherlock Embedded: Microsoft Internet Explorer version 8, Mozilla Firefox version 5, Google Chrome version 13. **NOTE: Internet Explorer version 6 is not supported. In IE6 you will see the page header and Logo, with no tabs or options.**

#### A. Change the PC Address

*In Windows XP:* From the Windows Start menu, select “Settings”, and “Network Connections”. Right-click on your network connection (Local Area Connection) and select “Properties”. Click on “Internet Protocol (TCP/IP)” and click on the “Properties” button.

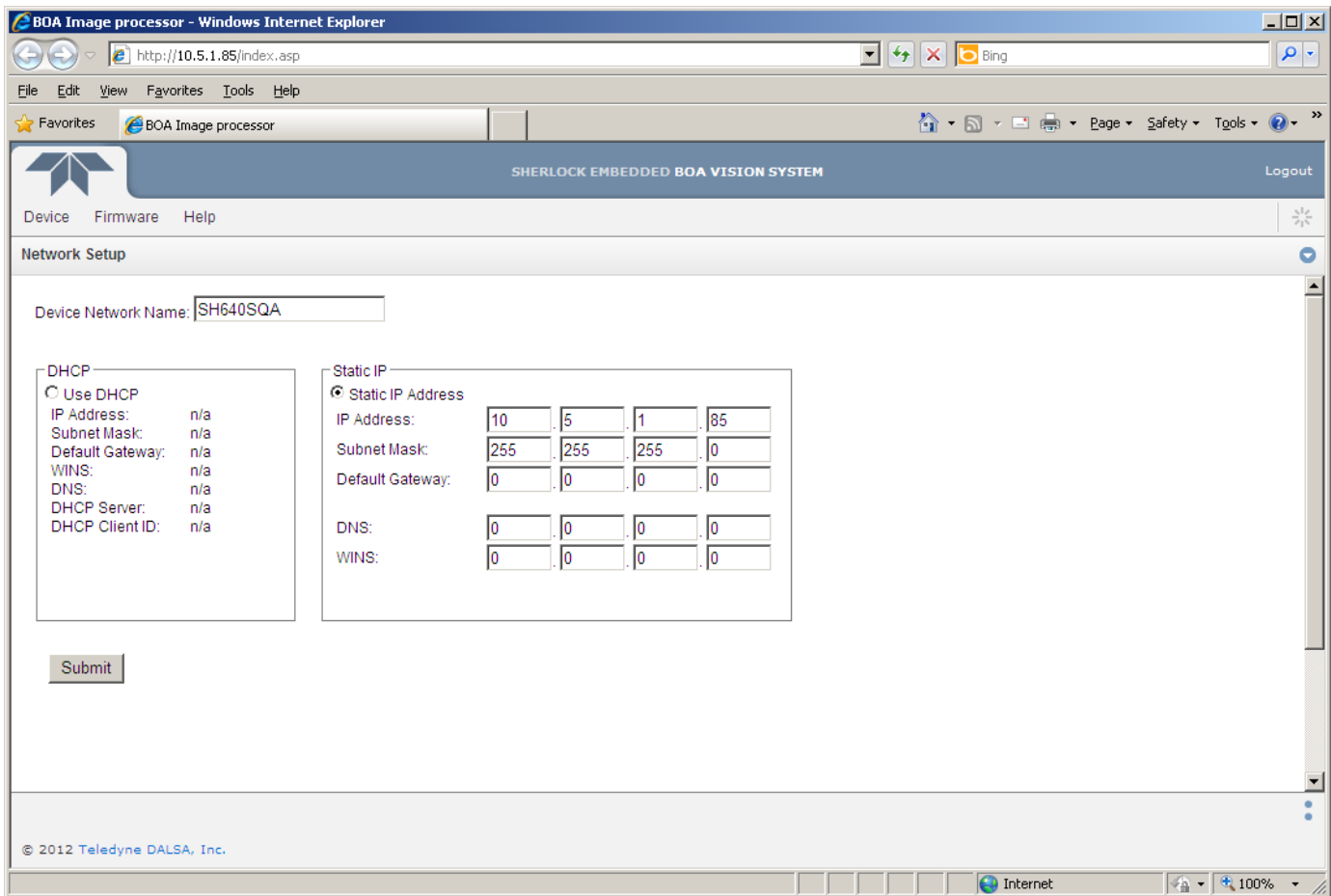
*In Windows 7:* From the Windows Start menu, select “Control Panel”. Select “Network” and then select “Manage Network Connections”. Right-click on your network connection (Local Area Connection) and select “Properties”. Click on “Internet Protocol Version 4 (TCP/IPv4)” and click on the “Properties” button.

**NOTE:** Before making any changes, write down or capture the current settings.

Click “Use the following address” if it is not already selected. Enter an address in this BOA Pro’s “network neighborhood”, for example 192.168.0.101. Click in the Subnet mask field, and the value 255.255.255.0 should appear. Leave the Default gateway blank. Click “OK”.

#### B. Access the BOA Pro

1. Use your web browser (Internet Explorer, Firefox, or Chrome) and enter the address of the BOA in the address field (do not add www prefix). The factory default address is 192.168.0.100. The browser will add “http://” at the beginning of the address. A page titled “Sherlock Embedded Image processor” should appear, displaying the current Device Settings page.
2. Hover your cursor over “Device” and select “Network Settings” in the drop-down menu.



3. On the Network Settings page you can change the IP Address and Network Name. Change the Subnet Mask if necessary. **NOTE:** In most machine networks that use addresses like 10.1.x.x the subnet mask should be 255.255.192.0. Using the default 255.255.255.0 can cause other devices to disappear or become inaccessible.
4. Click the “Submit” button to accept your changes. You should reboot (power cycle) the BOA Pro, to make sure the new name or address is used.
5. Close the Web Browser. Change the PC network connection settings back to the original settings. You do not need to reboot the PC after changing the Network connection.

## Windows Configuration

The Sherlock Embedded Client software is designed for a minimum display resolution of 1024 by 768 but a higher resolution is recommended. With a higher resolution, more views or windows can remain opened and rearranged in the Main Window. With smaller screen resolutions, you will need to drop views on top of each other, creating tabs, and thereby seeing fewer items or views at one time.



## Password Protection

There are two separate and unique password protection schemes implemented on the BOA Pro. The Web Server interface and Sherlock Embedded Client interface have separate password protection. The two passwords can be different.

### Web Interface Password Protection

The BOA Pro comes from the factory with no password, and normally is always logged in. (“logout” appears in the page). If you click “logout”, only the unprotected pages will be visible. If you click “login” the password dialog appears. If no one has added a password, you can click “Submit” to enable access to the protected pages.

**Unprotected Pages:** The “Device Information”, “Run Monitor”, and “Online Help” pages are not password protected. These pages can be reached without logging in.

**Password Protected Pages:** The “Update Firmware”, “Restore Factory Firmware”, “Network Setup”, “Change Password”, and “System Console” pages are password protected. If you logout, these pages are not accessible.

Use the “Change Password” page to change the default password. Remember your password! There is currently no way to reset the password for the BOA Pro Server interface, unless you return it to the factory for full reprogramming. This will delete all your programs and reset any configuration changes.

### Client Interface Password Protection

The BOA Pro comes from the factory with no password, and any PC that has the same version of the Client software can access the BOA Pro programming interface. If you enable password protection, anyone accessing the BOA Pro must supply the correct password in the Connect dialog, to open the Client Software interface. The client software will not connect to the BOA Pro if the correct password is not entered. Remember your password! The password is stored in the BOA Pro camera, not on the Client PC. The password for the Sherlock Embedded Client interface can be reset by using the Web interface to upgrade or reinstall the BOA Pro firmware, or by performing a factory reset (“Reset Factory Firmware”).

## Upgrading the Sherlock Embedded Software

You must upgrade both the BOA Pro camera and the Client Software to the same version. The BOA Pro upgrade file is binary file named “update.pak”. The Sherlock Embedded Client upgrade file is named “SherlockBOAexe”. Copy both of these to a PC that will be used to perform the upgrade.

### BOA Pro Smart Camera’s Firmware

You should **Export all programs** to the Client PC before upgrading the BOA Pro to a new version. All programs may be deleted during the update process. Do not export your programs to the default address: “C:\Program Files\Teledyne DALSA\Sherlock Embedded\Applications”. All files and directories below “\Sherlock Embedded” are deleted when you uninstall the Client software.

Upgrading the BOA Pro server is done from the Web Interface, using a web browser. You must also upgrade the Sherlock Embedded Client Software to same version.

1. The BOA Pro upgrade file is binary file named “update.pak”. Copy this file to your PC.
2. Open your web browser, and enter the BOA Pro camera’s address in the address bar.
3. When the “Sherlock Embedded Image Processor” page opens, hover your cursor over “Firmware” and select “Update Firmware”.
4. Click the “Browse” button and navigate to the “update.pak” file on your PC.
5. Click the “Submit” button. The file is uploaded, and the EagleEye reboots itself.
6. After a minute the Device Info page is displayed. The update is now complete.

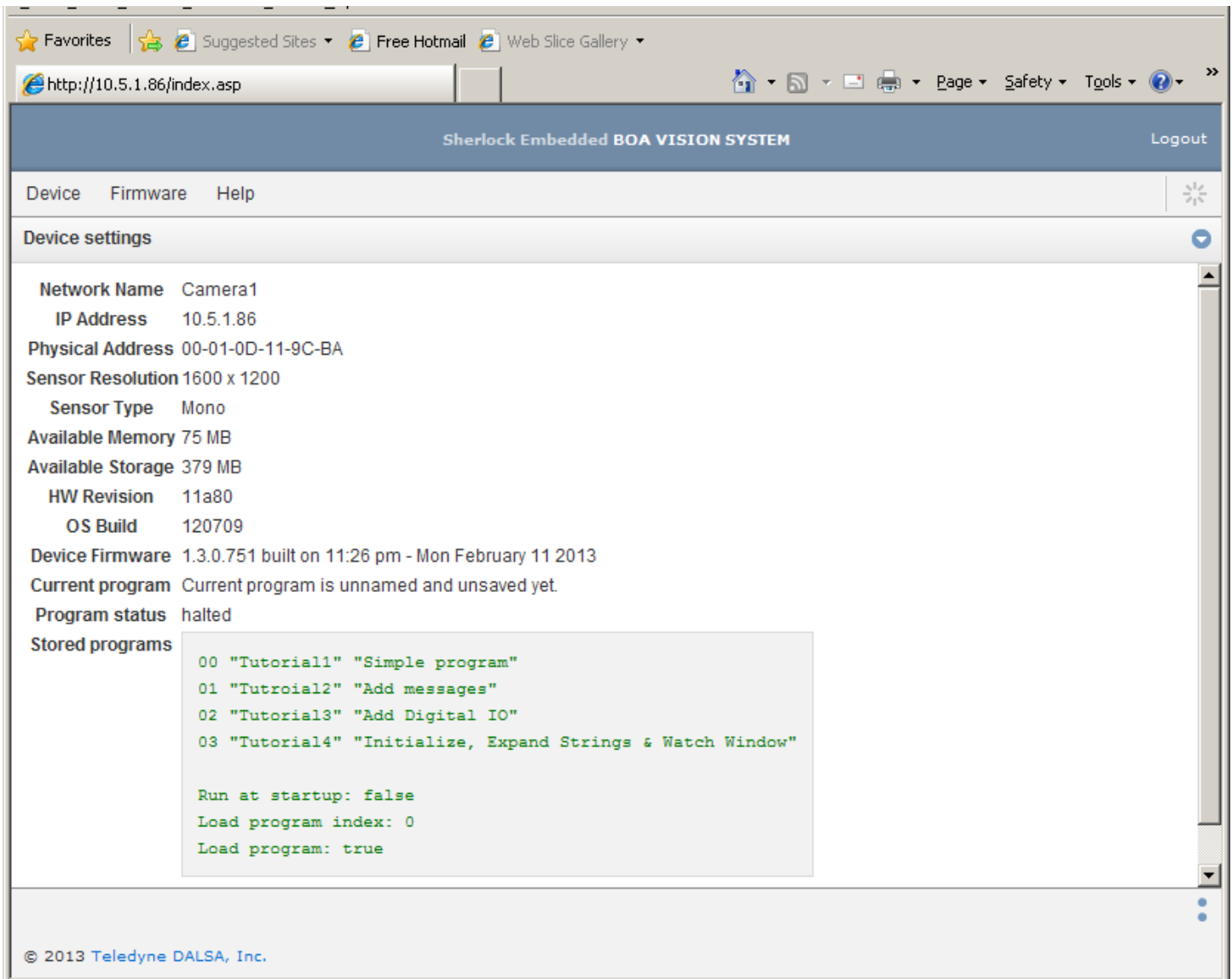
### Sherlock Embedded Client Software

The Sherlock Embedded Client and Emulator are incorporated in one software installation. If you have saved any programs using the Emulator, or exported programs from the BOA Pro camera to the PC, be aware that all files below the \Sherlock Embedded directory are deleted when you uninstall the Sherlock Embedded Client software. The default directory for saving programs is \Sherlock Embedded\Application.

1. The Sherlock Embedded Client software upgrade file is named “SherlockBOA.exe”. Copy this file to your Client PC.
2. Open the Control Panel and select “Add Remove Programs” (Windows XP) or “Uninstall Software Programs” (Windows 7).
3. Click on “Sherlock Embedded (remove only)” and click on the “Change/Remove” button.
4. Confirm the un-installation. When un-installation is complete, close the message window and reboot the PC.
5. When the PC reboot is completed, use Windows Explorer to navigate to the copy of “SherlockBOAexe”. Double-click on the file (or right-click and Run as administrator) to start the installation.
6. Confirm or change the destination directory.
7. When the installation is complete, click the “Close” button.

## BOA Pro Web Server

You may need administrator privileges to access the BOA Pro. You may need to change your browser's security settings to reliably access the BOA Pro. Refer to the document "Configuring Windows for BOA" included on the BOA Pro CD.

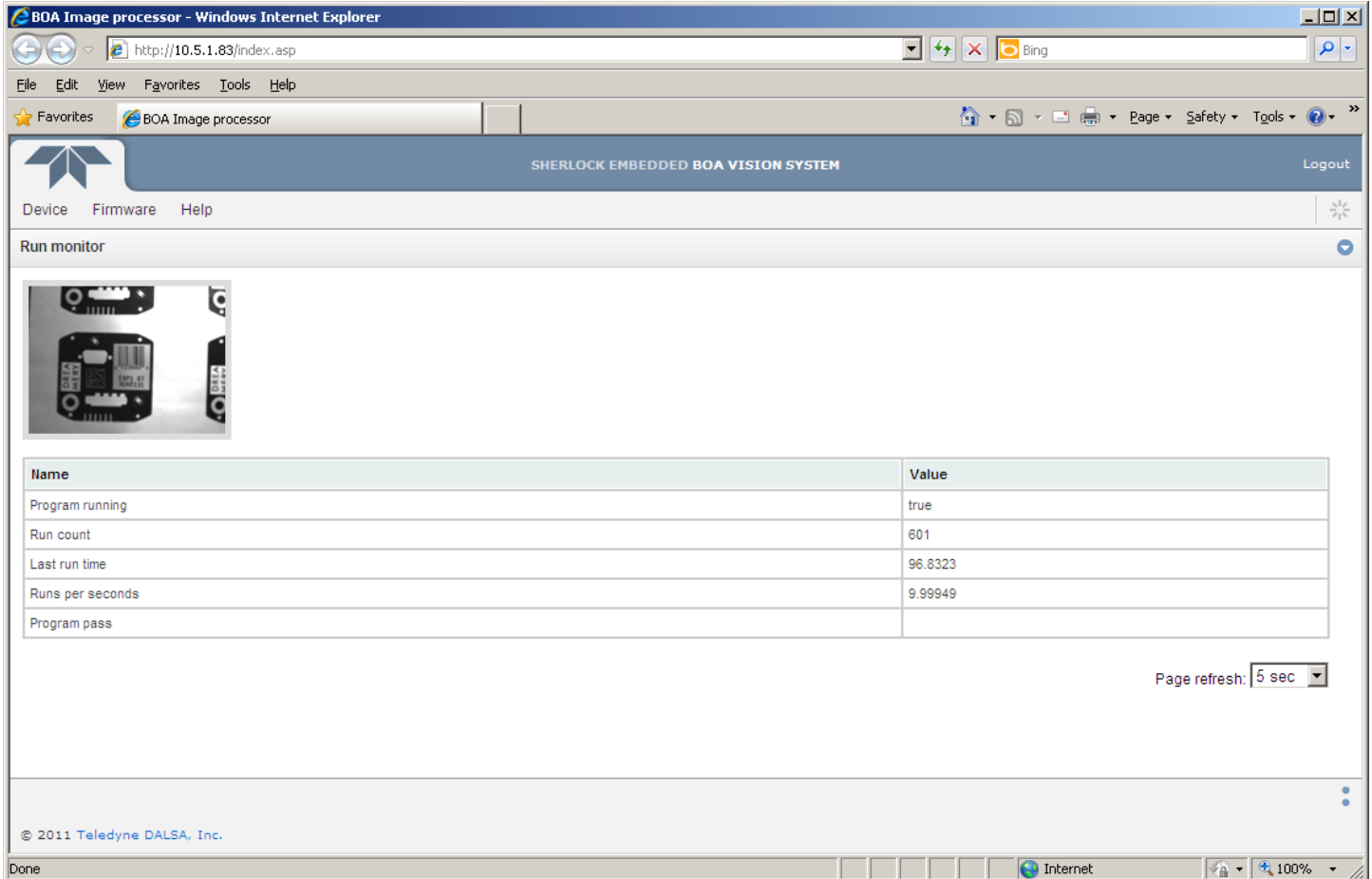


- The Device Information, Run Monitor and BOA Help pages are not password protected. These pages can be reached without logging in.
- The Update Firmware, Restore Factory Firmware, Network Setup, Change Password, and System Console (command) pages are password protected. If you logout, these pages are not accessible.

### Device Menu

**Device Information** is the default page (shown above). Gives information about the current BOA Pro configuration: Network Name, IP Address, Physical Address, Sensor Resolution, Sensor Type, Available Memory, Available storage, Hardware version, Software version and date. (shown on previous page)

**Run Monitor** page allows any PC to monitor the inspection progress (shown on the next page). You change the refresh (update) rate for this page. The displayed Image size can be changed in the Sherlock Embedded Client, in the System Properties.



**Network Setup** page allows you to change the address and mode, change network name. (shown on page 8)

**Restart Device** reboots the BOA Pro camera.

**Change Password** page allows you to add password protection to the Web Server interface. The protected pages will not be accessible without entering the correct password. The Device Info and Run Monitor pages will still be accessible without a password.

**System Console** page allows you to enter commands to retrieve information on programs, load a program, or change the run state. Type **help** in the field and click the "Run command" button to get a list of valid commands.

Note: The same commands are available through TCP/IP ascii commands on port 4505 (version 1.2.0 and later).

## Firmware Menu

**Update Firmware** page allows you to upgrade to a later software version, using the "update.pak" binary file. Click the "Browse" button to find the upgrade file on the client PC. Click the "Submit" button to start upgrade process. The page will change and ask you to allow the BOA to restart.

**Restore Factory Firmware** page allows you to return the BOA Pro smart camera to its factory installed or default state. This resets the software version and resets the address to the factory default.

## About Menu

**Online Help** page displays help for the Web Server interface.

## Sherlock Embedded Software Design

Sherlock Embedded and the BOA Pro vision system represent a whole new generation of the Sherlock machine vision software. Previous designs of Sherlock ran on a fully configured Windows PC platform with keyboard, mouse and display for programming and configuration. After programming, Sherlock could be run on one PC and viewed remotely on a second PC.

Sherlock Embedded has been redesigned to run on a remote processor or “headless server” with no keyboard, mouse or display, and only accessed and programmed remotely. There are no IPP or ActiveX components in the Sherlock Embedded software.

All formulas, instructions, algorithms and preprocessors are now treated as instructions. All system resources and instructions are treated as objects.

1. Everything is an “object” or program object.
2. The **System Window** exists to define and manage objects that are global or system-wide resources. Some resources are predefined (cam0, inputs, outputs, serial ports). The System Window allows you to create some objects or resources (variables, alignment objects, calibration objects, some Communication objects, etc.).
3. All objects have properties.
4. The **Properties Window** exists to display or modify an object’s properties. Not all properties can be modified. The Properties Window has two tabs: “Basic” and “Extended”. The Basic tab is a table of values. All properties that do not conform to this table format, appear in the Extended tab. The Extended tab is used to display line profile information, train OCR and Search algorithms, train target calibrations.
5. The **Program Window** exists to assign instructions to objects or resources, and define an order of execution. The Program window allows you to create instructions and subroutines, and assign variables to instruction inputs or outputs. Algorithms and Preprocessors are image processing instructions that can only be created inside or under the scope of an ROI. ROIs can only be created inside or under the scope of an Image Window.

## User Interface Overview

### Getting Started

See the **Sherlock Embedded Tutorial** included on the BOA Pro Product CD! The tutorial provides an introduction to the Sherlock Embedded GUI and programming features of the Sherlock Embedded Software and BOA Pro I/O.

### How Sherlock Works

#### Running a Program

**Run Once:** When Sherlock executes an "investigation" program once, each instruction in the "main" routine is executed in sequence, from the top of the Program window, to the bottom (including any branches or subroutine calls). After the last instruction is executed the investigation stops (the Sherlock Embedded software does not exit).

**Run Continuously:** When Sherlock executes an "investigation" continuously, each instruction in the "main" routine is executed in sequence, from the top of the Program window to the bottom (including any branches or subroutine calls). After the last instruction is executed, the sequence is repeated from the beginning (top) of the "main" routine. This repeats until the user halts running the investigation (the Sherlock Embedded software does not exit).

You can leave the investigation program running (continuously) on the BOA Pro vision system, and disconnect the Client application or close the Client application. The investigation will continue running on the BOA Pro without supervision.

[When you reconnect to a running BOA, the investigation slows temporarily while the connection is established, then resumes running a full speed.](#)

#### Programming Sherlock

A Sherlock "investigation" program is based on the Image Window and Program Window. You draw one or more ROI ("Region Of Interest") in the Image Window, and add image processing to the ROI. Right-click in the ROI (in the Image Window), or right-click on the ROI instruction in the Program Window, to open a popup menu and add processing (algorithms and preprocessors). In general, a preprocessor changes the image data, and an algorithm extracts data or measurements from the image. The measured results are called "readings", "output properties" or "objects". In the Program Window, you add instructions to manipulate the readings, and communicate results or decisions to the user or to peripheral equipment, through the Serial, Ethernet, Modbus or PLC interfaces.

The best way to become familiar with the Sherlock Embedded application, is by following the **Sherlock Embedded Tutorial** available as a PDF file on the BOA Pro CD-ROM that ships with the BOA Pro Smart Camera.

#### On-Line Help

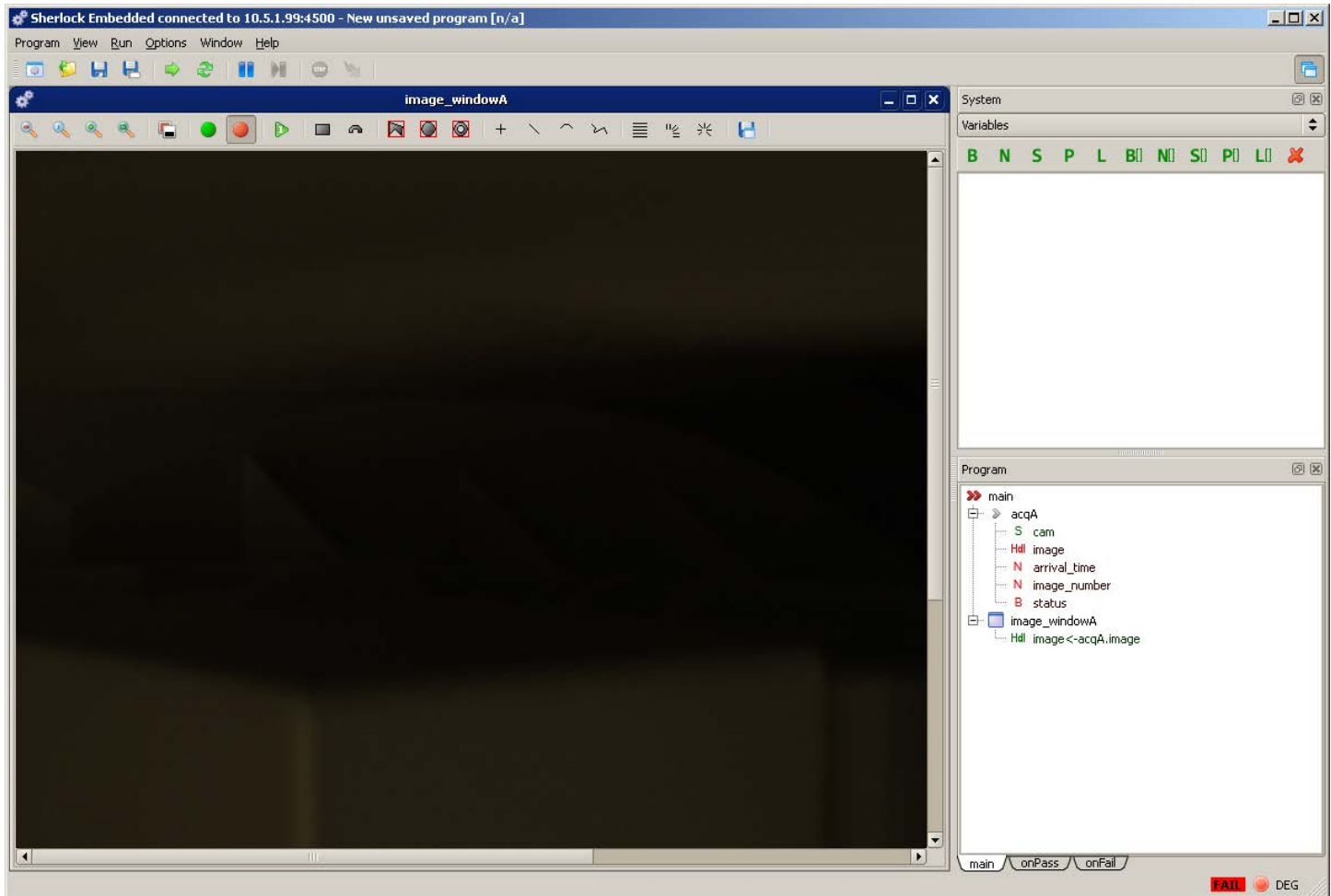
The On-line Help documents on all of the system resources and instructions. After following the Tutorial, you can access the Help for additional details and useful information.

## Sherlock Embedded Client Application Windows

Sherlock Embedded uses several windows to display features and allow the user to control them. When you exit the Client interface, your window layout is saved to a file, and is restored the next time you open the Client interface. This file is stored on the Client PC, not the BOA Pro. This means that the window layout can be different on each PC.

### Main Window



The Sherlock "Main" window opens when you connect to the BOA Smart Camera from the Sherlock Embedded Client software. The Main window contains all the other window panes or views. The Main window is never the focus.



At the top of the Main window is the Menu bar and main Toolbar. The toolbar buttons have float-over tips that identify their actions.

The Windows® minimize button at the top right corner of the Main window, minimizes the Sherlock Main window (and all other windows or views) to the System Taskbar.

The Status bar at the bottom right corner of the main window shows the Pass/Fail status, the current run status (Stop or Run Continuous) and units for angles (Degrees or Radians).

There is a "Docking Control" button in the Main window Toolbar, which toggles between Docking enabled  and Docking Disabled . With Docking disabled, windows will stay separated or "float". With Docking enabled, you can

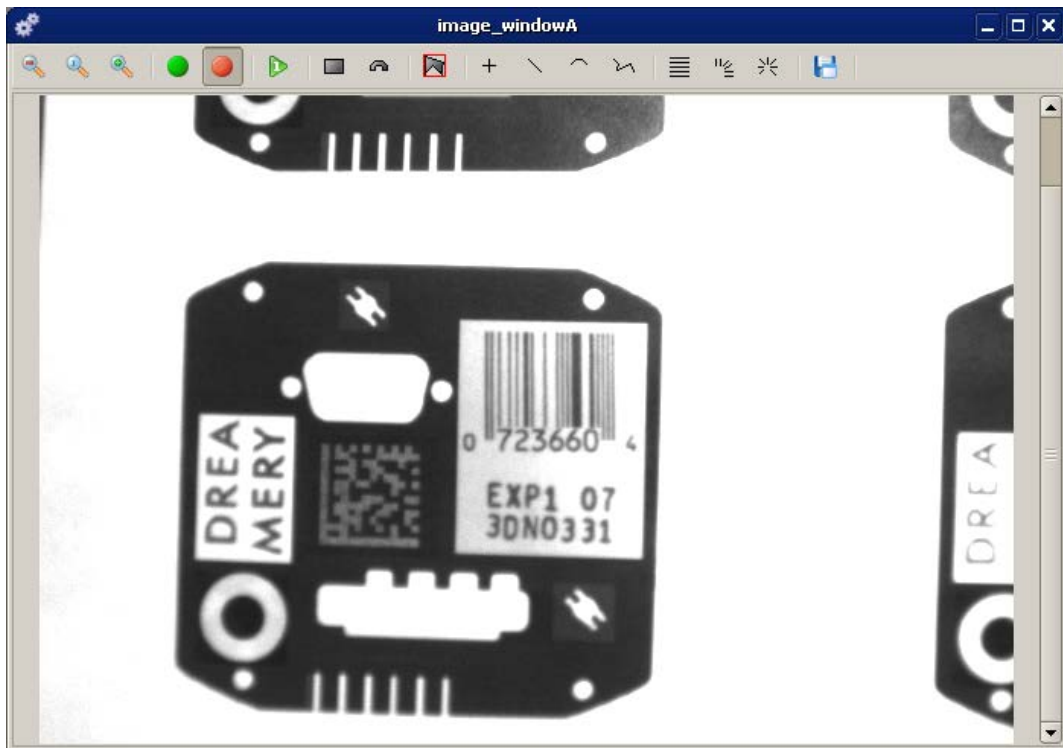
drag and drop windows onto each other. Tabs will appear at the bottom of the stacked windows. You can resize windows. You can Dock and Un-dock all windows within the Main window, except the Image Window. You can drop a window into another window, to create **Tabs**, for viewing multiple windows. You will also see docked windows shift around to allow dropping a window beside docked windows. The cursor, not the Title Bar, determines where an undocked window tries to attach or dock.

**If a window is not visible**, but it is enabled in the View menu, it may be a tab inside another window. Check for Tabs at the bottom of visible panes or Views.

Your window layout is saved each time you close the Client application, in a file "layout.ini" in the Sherlock Embedded\Application directory. This file is local to the Client PC, and is not saved to the BOA Pro camera. The layout can be different on each PC.

### Image Window

The Image Window contains or displays images; normally from the BOA Pro Smart Camera, (or in the Emulator a sequence of image files). You can have multiple image windows open inside Sherlock. You can minimize the Image Window.



The Image Window cannot be docked, or dropped onto another window. You can minimize, maximize, or close the image window with the standard Windows® controls. Once you have maximized the Image window, these controls appear in the top right corner, below the controls for the Main window. You can also delete an image window in the Program Window. You can right-click in the image window to select a color "palette" to add colorization to the image.

You use the Image Window to draw ROIs in the image, on or around features you are interested in modifying or measuring. You add image processing to the ROI either in the Image Window or in the Program Window. You can right-click inside an ROI to add image processing algorithms and preprocessors, copy, or delete the ROI. When you copy and



paste an ROI, the new ROI appears directly on top of or at the same location as the original ROI. You can drag the new ROI to a different location. **NOTE:** If you delete an Image window, all ROIs and processing in that window will be lost.

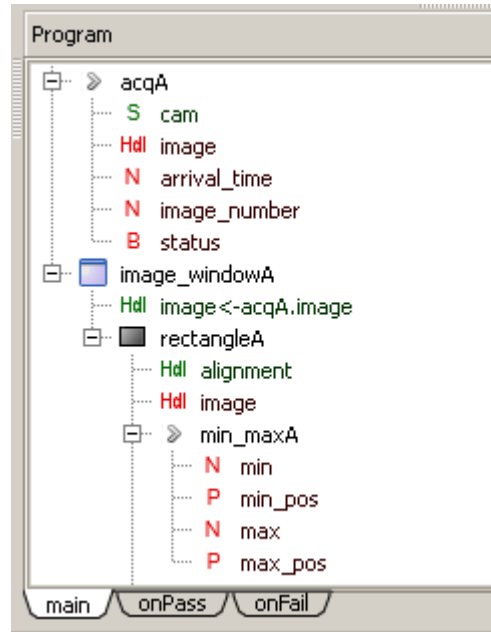
You can right-click on any reading displayed in the Image Window, to hide the reading or connect it to an instruction.

## Program Window

The Program Window assigns instructions to objects or resources, and defines the order of execution. The Program Window contains the structure of your investigation program. Your Image Windows, ROIs, and Instructions are listed here. You can have more than one image window in the Program Window. You can have more than one ROI in an image window.

There are tabs at the bottom of the Program Window, for the main routine and subroutines. Three routines are shown: main, onPass, and onFail.

The order of instructions is in your control. The Sherlock Embedded program does not restrict or pre-define the order.



You add subroutines and instructions to the Program Window by one of two ways:

- Drag and drop instructions from the Instructions Window, if it is open. New instructions appear **below** instruction the new instruction is "dropped on".
- Right-click on any instruction in the Program Window to get a pop-up menu of the instructions. New instructions appear **below** the highlighted item. If you have selected a ROI instruction, the Algorithms and Preprocessors are also shown in the pop-up menu of instructions.

The pop-up menus are context sensitive. You do not get the Instructions popup if you right-click on an instruction's input or output. The Instructions popup contains Algorithms and Preprocessors only when you right-click on an ROI instruction.

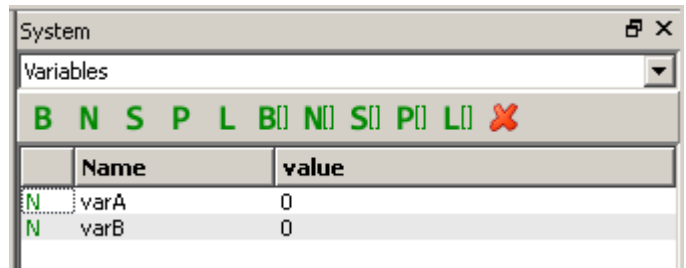
- You can right-click on an instruction's input or output to connect it to a variable or another program resource.
- You can right-click on an output (or reading) to hide the reading in the Image Window (where an algorithm may have several outputs you are not interested in).
- You can drag outputs from the Program Window into the Watch Window.

When you copy an instruction, you get the choice to "paste" or "paste single instruction". When you select "paste single instruction" only the highlighted instruction is pasted into the Program Window. The children instructions are not pasted. When you select "paste" the highlighted instructions and all children instructions within the scope of the copied instruction are pasted into the Program Window.

## System Window

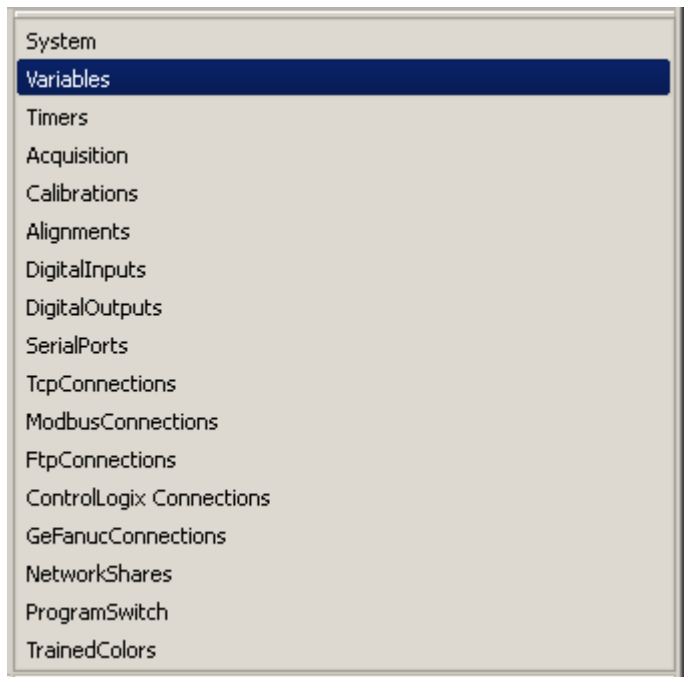
The System Window exists to define and manage objects that are global or system-wide resources. Some resources are predefined (cam0, COM3). The system window allows you to create some objects or resources (variables, calibration, alignment, etc.). A menu bar or toolbar may appear when you select different Resources in the System Window.

Pre-defined resources that always exist (Serial Port, Camera, Digital Inputs, Digital Outputs) do not provide a toolbar or menu bar.



There is a "combo box" or drop list in the System Window ("Variables" in the previous image). Use this to select a category.

Resources that you create as needed, provide either a menu bar or Toolbar (Variables, Timers, Alignments, Calibrations, TcpConnections, and ModbusConnections). These resources can only be created or deleted in the System Window.



## Properties Window

Double-click on any instruction in the Program Window, or any resource in the System Window, to open the Properties Window. The Properties window displays an object's properties. Not all properties can be modified. The Properties Window has two tabs: "Basic" and "Extended". The Basic tab is a table of values. All properties that do not conform to this table format appear in the Extended tab.

### Basic Properties




The Basic tab has three columns: Type, Property and Value. The **Type** column shows the data type. Most types match those shown in the Variables window: Boolean, Number, String, and arrays of these data types. Other types are specific to data types defined in the structure of the Sherlock environment, such as Points, Lines, and handles to Images, Routines, hardware and peripherals.

The **Property** column shows the name of each property.

The **Value** column shows the current value of the property. The Value column has several different controls for entering data or values. The controls change with the accepted entries.

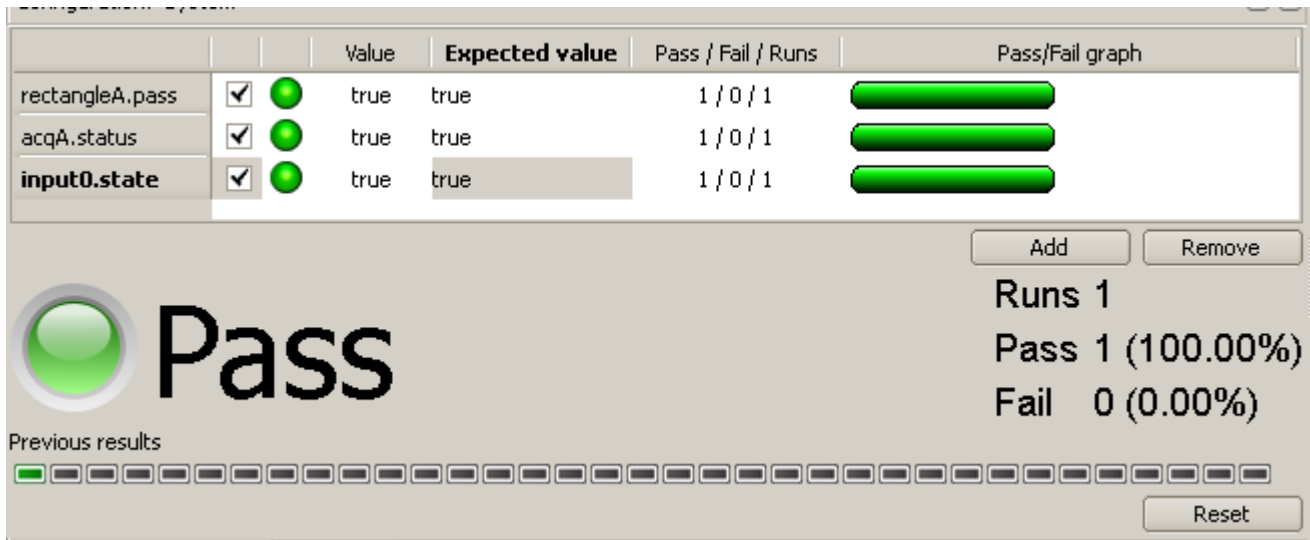
Type	Property	Value
Properties		
S	name	acqA
B	enabled	true
N	time	1.95
Inputs		
S	cam	cam0
Outputs		
Img	image	[handle]
N	arrival_time	56895634063.72
M	image_number	413

Most editable values have a field mode button at the far right. This button shows one of two entry modes. Click on the field control button to toggle between the two modes:

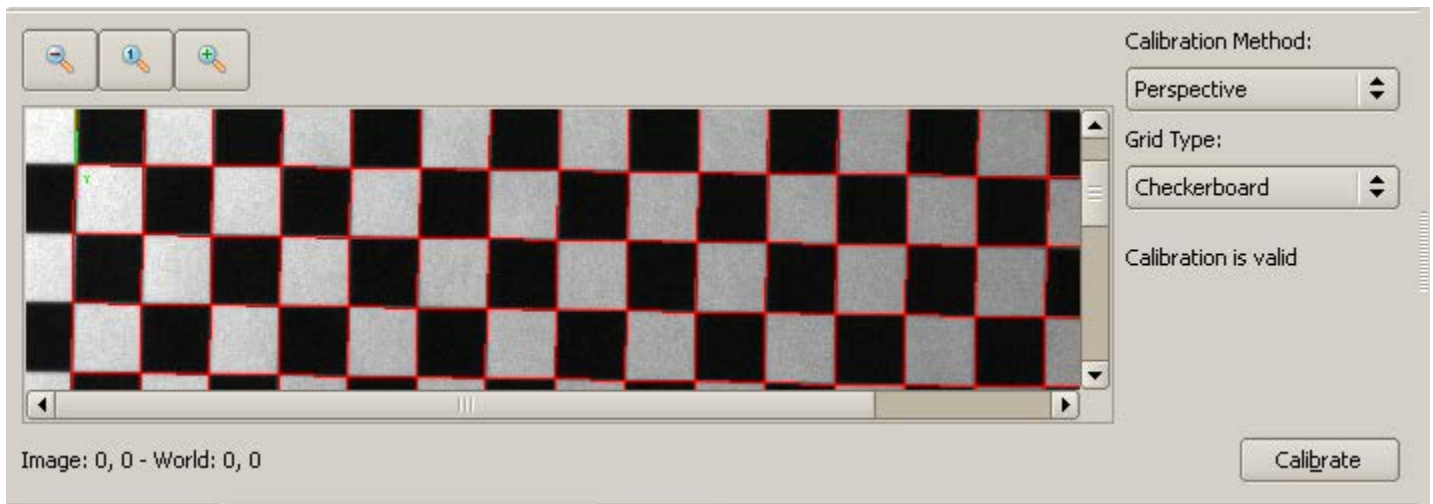
-  Entry field - enter numbers or strings in this field, or use other controls (number wheel, slider) for value entries.
-  Link - attach the property value to a variable or program object. The Value field changes to a drop list.
-  Advanced Edit - open an editing dialog or window. The type of dialog depends upon the property being edited. Examples are: String, Array, Matrix and Image.

**Extended Properties**

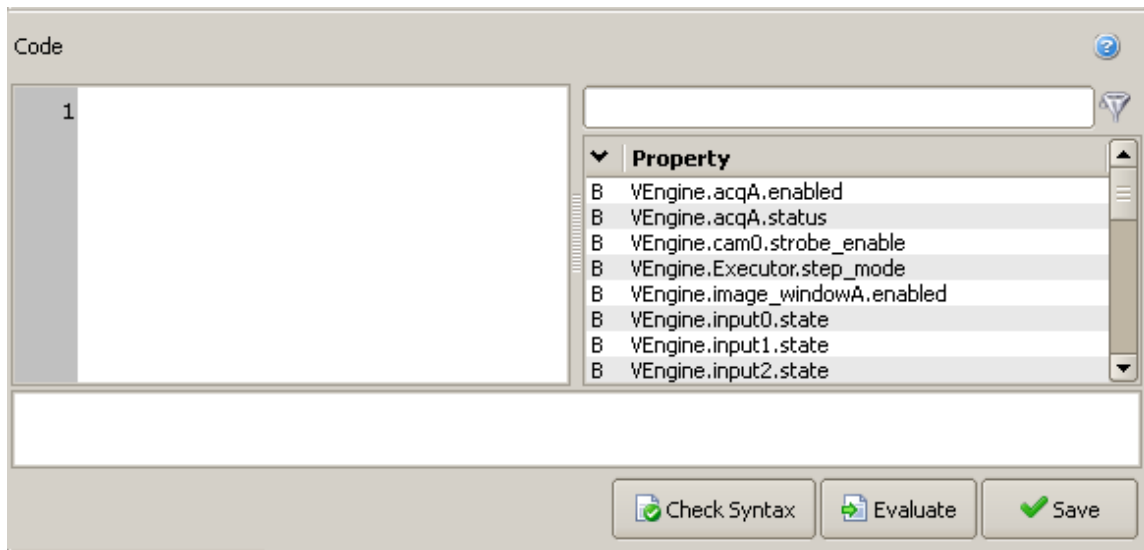
The Extended tab exists to display or modify object properties that do not fit the table presentation of the Basic tab. The Extended tab changes with the selected object:



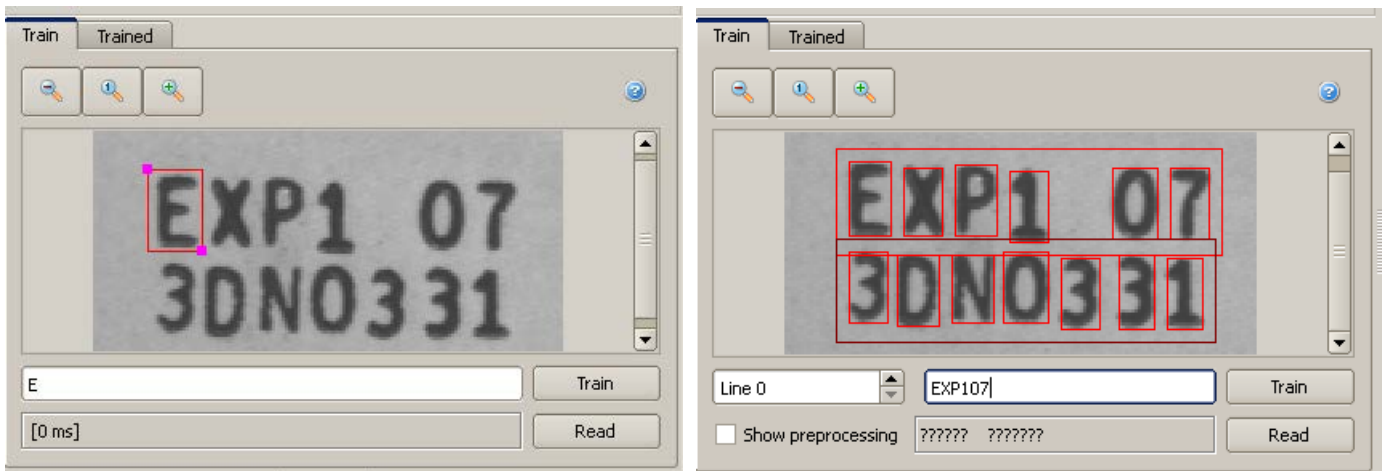
**Decision Table** – appears when you click on the system object in the System Window. The upper half of this window is for adding, deleting, enabling and disabling individual variables or program objects in the decision table. The bottom half of this window shows the composite results for all enabled objects. If all objects pass, the composite result passes. If one or more objects fail, the composite result fails.



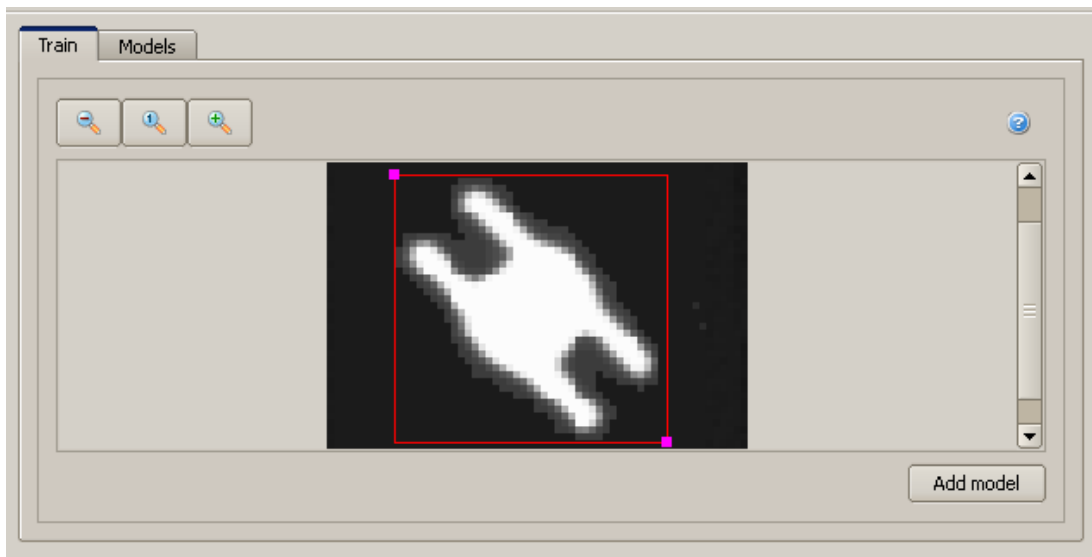
**Calibration Training** – appears when you click on a Calibration object in the System Window. The Target calibration method uses actions in this Configuration Window to perform the Calibration. Manual calibration uses the Properties window and this window together. The programmed calibration methods calib\_using\_transform and calib\_using\_pts do not use actions in this window to perform the calibration.



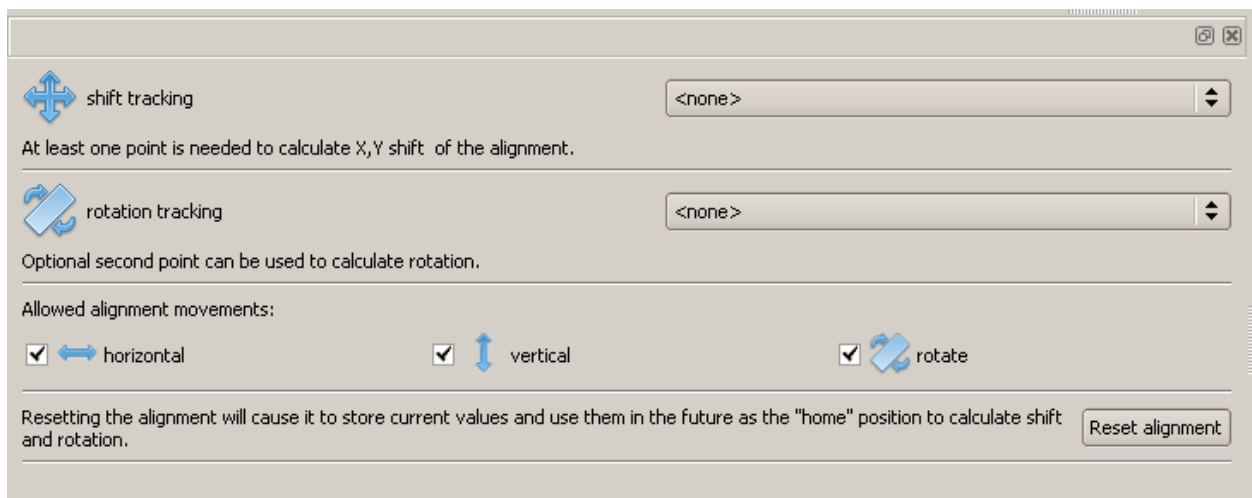
**Script Editing** – appears when you click on a script instruction, or an ROI instruction with the "enable\_pass\_fail" property true (enabled) in the Program Window. The left pane contains the statement. The right pane displays the variables and objects defined by the Sherlock environment and your program statements. You can click on the headings (right pane) to sort the defined objects by type or name. The third pane across the bottom is a message pane, and will display errors when you click the "Check Syntax" button, or will display the result when you click the "Evaluate" button. **NOTE:** You must click the "Apply" button to add the script to your program.



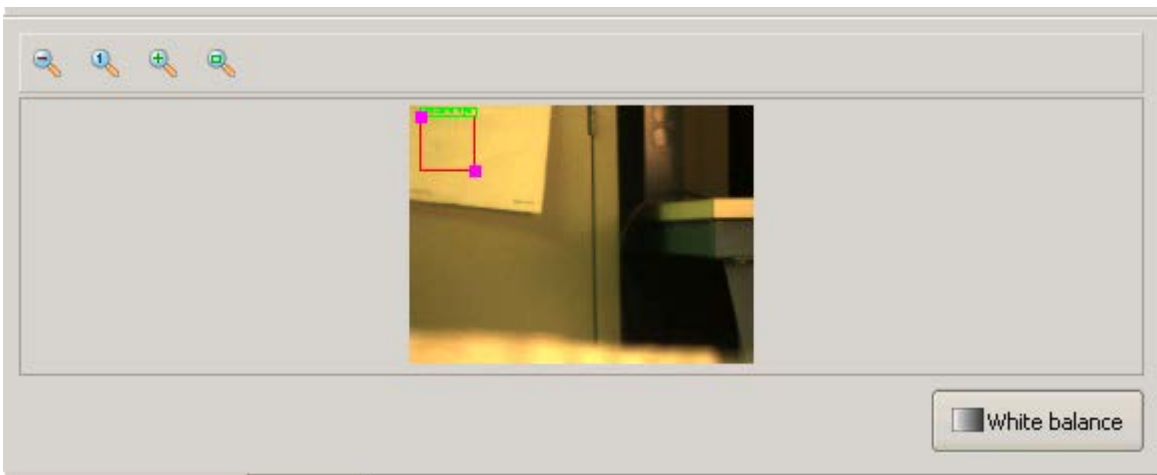
**OCR Training** – appears when you click on any OCR algorithm in the Program Window. There are two tabs at the top of this window, for "Train" and "Trained". The **Train** tab shows the ROI image, and has buttons at the top for zoom in, out, and reset. In ocr\_grayscale (above left) there is a rectangle to position over each single character to train. In ocr\_binary (above right) the characters are separated from the background, and rectangles appear over the characters automatically. You can adjust the threshold settings (in the Properties Window) to adjust the separation. The **Trained** tab lists all characters trained in this ROI. You can select characters and use the buttons at the bottom of the panel to enable, disable, rename and remove (delete) trained characters.



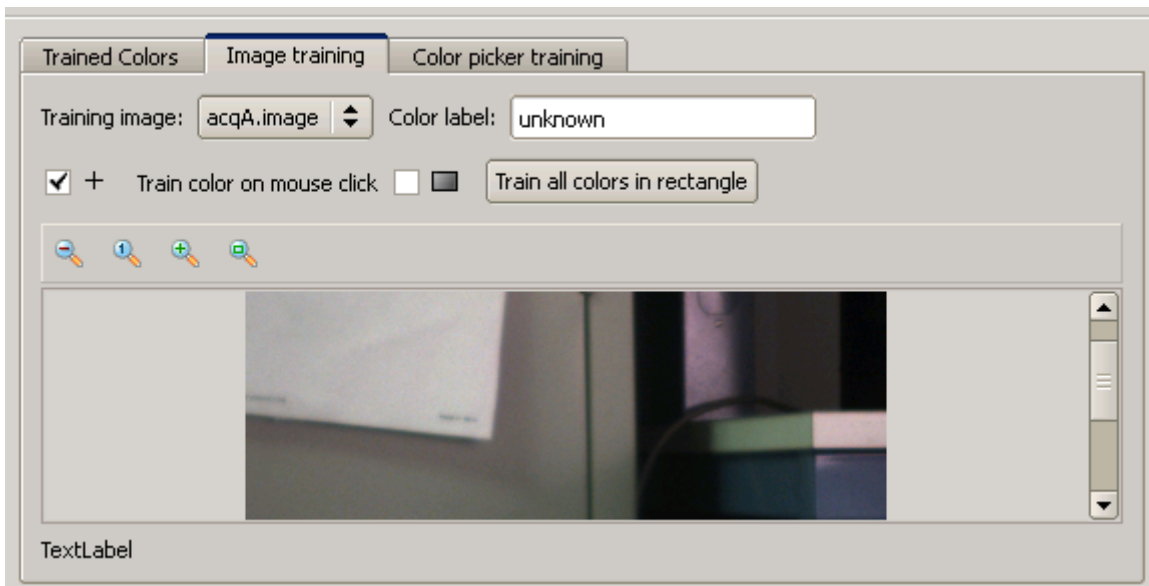
**Search Model Training** – appears when you click on any Search algorithm in the Program Window. The Search Correlation has two tabs at the top of this window, for Train, and Model. The **Train tab** shows the ROI image and a small rectangle to outline the actual feature for the model to be trained. Buttons at top for zoom in, out, reset zoom. The **Models tab** shows the models trained for this ROI, and status. You can select a model and use the buttons to enable, disable, rename and remove (delete) a model. The Search Edge has three tabs at the top, for Train, Model, and Model Editor. You can use the Model Editor to delete some edges from the trained model, such as changing or extraneous features.



**Alignment Training** – appears when you create or click on an Alignment object in the System Window. Use the two drop lists to select existing points in your image, to use as anchors. Use the checkboxes to define the allowed movement. Finally, after you have added your measurement ROIs, and assigned the Alignment object to each of them, click “Reset alignment” in this window or in the Alignment Toolbar in the System Window, to train the positions of the Anchors and all ROIs attached to the alignment. Anytime you later move or resize your ROIs or add new ROIs, remember to click “Reset alignment” again.



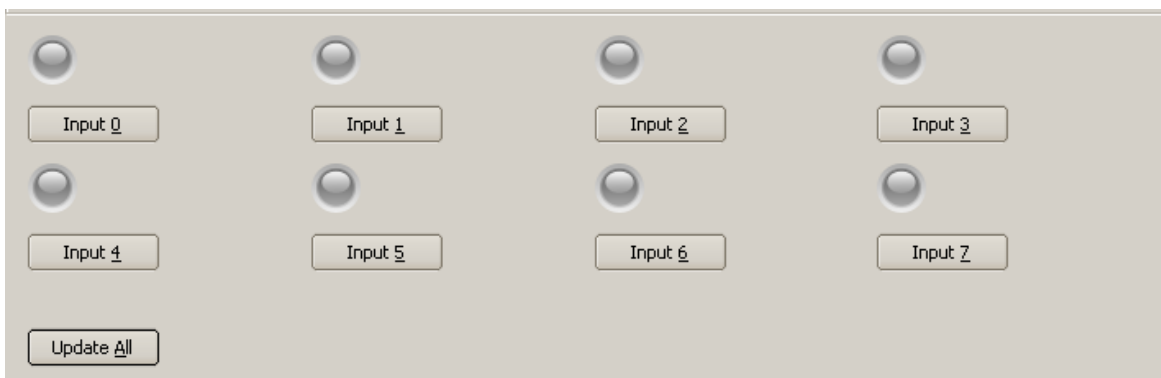
**Color Balance** –if you have a color camera or color image, appears when you click on the Camera Properties in the System Window, the Configuration Window changes to display the camera image. The White Balance button appears in the bottom right corner, and a small red rectangle ROI appears in the top left corner of the image. Position the rectangle over a white area in the image (not a overexposed or “bloomed out” area) and click the White Balance button.



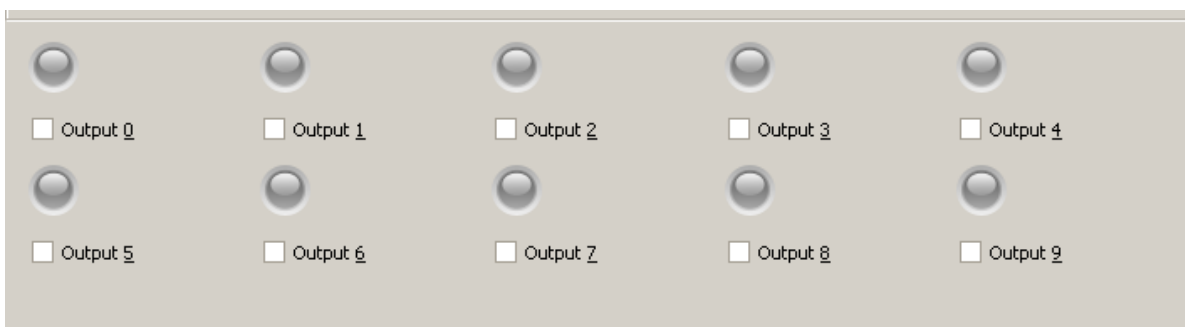
**Color Training** – if you have a color camera or color image, appears when you create a Trained Colors object in the System Window and then click on the Trained Colors A object, the Configuration window changes to a Color Training window. You can enter a label (name) and train either a single point (mouse click in the image) or all samples within a rectangle. This creates a list of trained color samples. When you run the investigation, the list of samples is expanded into a 10M lookup table. **NOTE:** The first time you run an investigation after training or loading a saved program, the run time includes time taken to expand the color data into a lookup table. After the first run, the time is much shorter.



**Line Profile** – appears when you click on any edge detection algorithm in the Program Window. The Line Profile shows the *intensity* plot in black, the *edge strength* or gradient (second differential) plot in red, at each position along the line. A green line (or lines) shows the edge strength *threshold* setting(s) for the algorithm. The gradient shows the rate of change and direction of change; light to dark is a negative gradient, dark to light is a positive gradient. The gradient is useful for determining how strong the edges are (edge strength) for edge-detection algorithms and preprocessors.

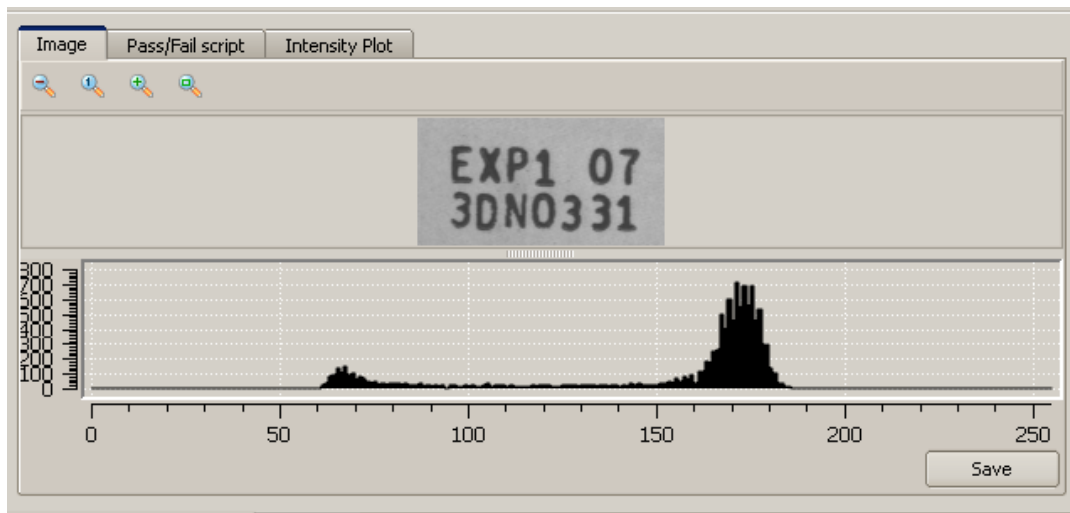


**Digital Inputs** – appears when you click on a (any) digital input in the System Window, the Configuration Window changes to display all of the available digital inputs, and their current status. You can use the buttons in the Configuration Window to read and update the state of any one input, or all inputs. This figure shows the 8 inputs available with the PL-200 module.



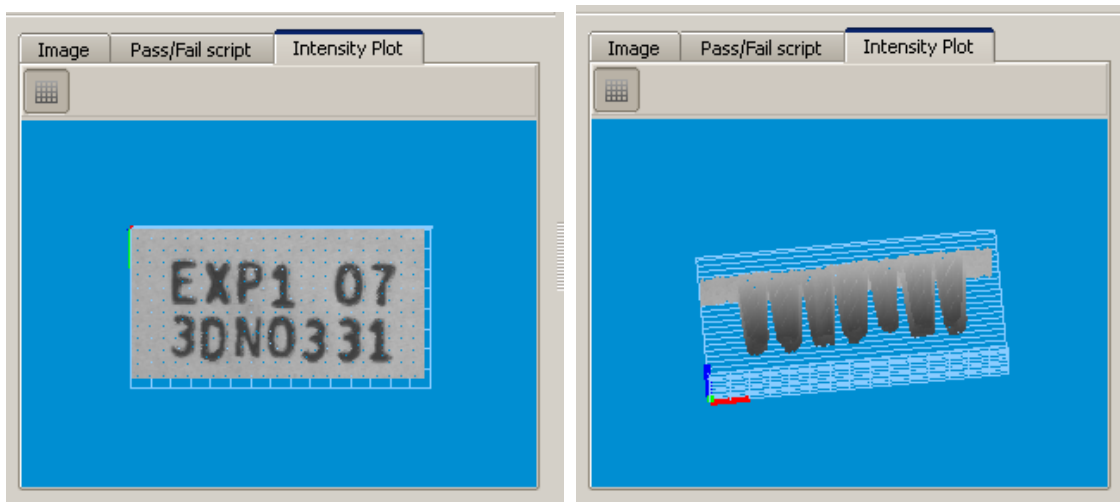
**Digital Outputs** – appears when you click on a (any) digital output in the System Window, the Configuration Window changes to display all of the available digital outputs, and their current status. You can use the checkboxes to turn an output on and off. This figure shows the 10 outputs available with the PL-200 module.





**ROI Profile** – appears when you draw a new ROI, or click on an ROI instruction in the Program Window. The first tab “Image” (above) shows the image and the histogram for an area ROI, or image and the intensity along the line for a Line or Composite ROI. There will be three plots in a color image; for red, green, and blue.

The second tab “**Pass/Fail script**” shows the script edit window (page 21) for the optional ROI pass/fail test (if the pass\_fail property is “true” in the Property Window).



The third tab “**Intensity Plot**” shows a 3-D intensity plot in (X,Y,Intensity). You can rotate (above right) this image to view the intensity characteristics. This tab is not available for a Line or Composite ROI. **NOTE: Not all video cards correctly support the Intensity Plot. If the Client application crashes, try updating your video card drivers. If the crash continues, do not use this tab. Windows does not allow us to trap the error and avoid the crash condition.**

## Instructions Window

There are 3 tabs at the bottom of the Instructions Window.

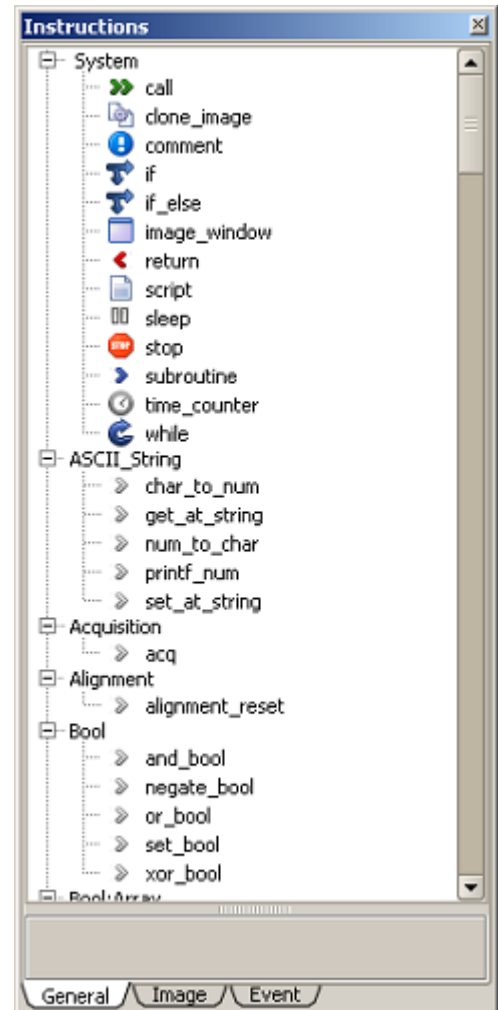
The “General” tab lists all the Instructions (formulas and I/O instructions) available, organized in Groups.

The “Image” tab lists all preprocessors and algorithms for the selected ROI type.

The “Event” tab shows all the Event Handler instructions.

You insert instructions into you program by dragging the instruction from the Instructions Window into the Program Window.

You can also right-click in the Program Window to insert instructions.



## Watch Window

The Watch Window displays live data values as the inspection program executes. Drag outputs into the Watch Window, from the Program Window. Drag variables into the Watch Window, from the Program Window or from the System Window.

Watch		
	Property	Value
N	NumberPassed.v...	3
N	NumberFailed.value	0
N	min_maxA.max	255
N	square_root_nu...	15.9687

Note: The Sherlock Embedded Client window layout is saved in a file on each Client PC, not in the investigation program. The layout may be different on each PC. Opening the Watch Window and saving the program and layout on one PC, does not force the Watch Window to be open on another PC.

## Monitor Window

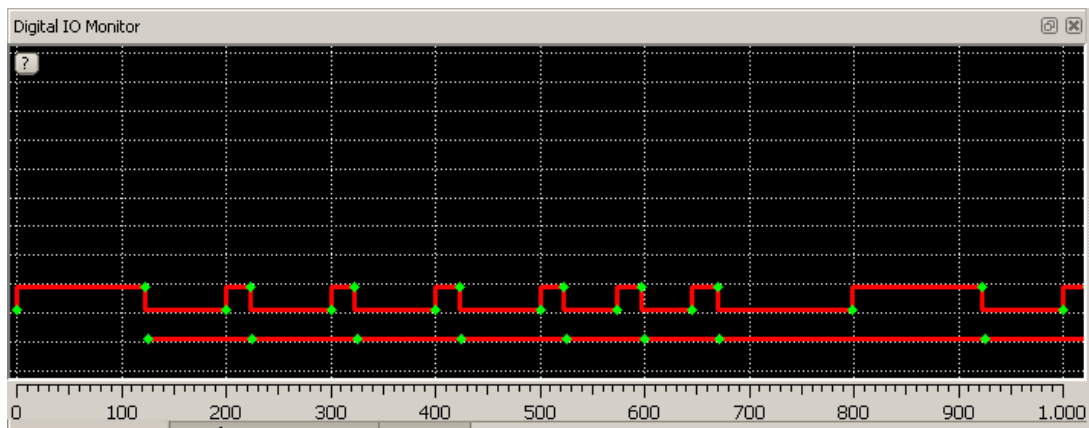
The Monitor Window displays error messages, warnings, and instruction time information. The Monitor Window does not force itself in front of other windows if it is a tab. You can select what items or events are displayed, by clicking on the Monitor level button in the Monitor window toolbar. Right-click in the Monitor Window to clear all messages.

You can send messages to the Monitor window (as shown) with the `display_message` instruction (found under IO:Monitor).

Monitor		
Time	Source	Message
18:17:24.697		Connected to 10.5.1.83:4500 Passed! Count = 1 Passed! Count = 2 Passed! Count = 3

### Digital IO Monitor Window

The Digital IO Monitor Window displays a time plot (graph), of all hardware digital outputs and event timers defined and in use by the program (disabled timer instructions are not displayed). You can also enable (in Application settings) displaying trigger events in this window. Click on the Question Mark button, at the top-left of the graph, for a legend of plot colors, and window controls. You can pan, scroll, and zoom the display in this window.



## Saving Exporting and Importing Files

### Save Programs

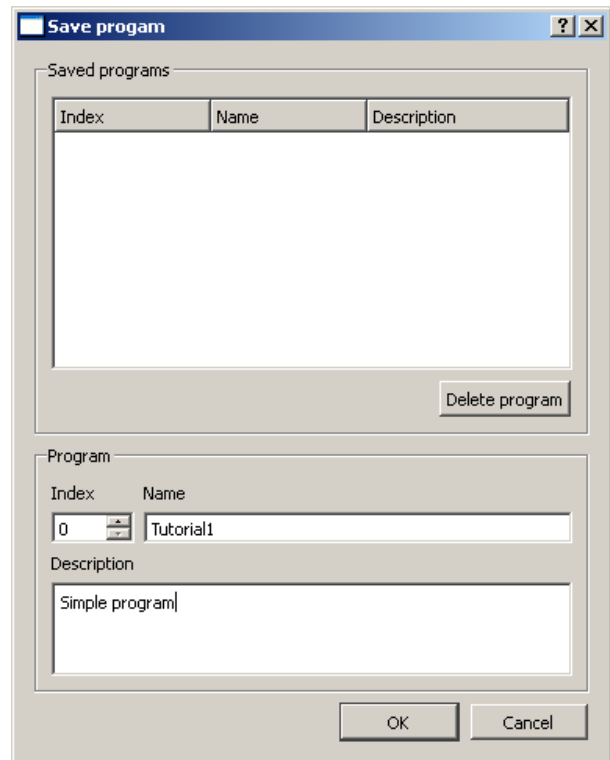
Programs are saved to the BOA Pro program memory. Programs saved in the BOA Pro memory are not lost when you power down or reboot. Currently Program files are not overwritten by a firmware upgrade. But you should Export all programs before upgrading. These programs appear in the Load Program dialog, when you select open or open program file.

In the Main Window toolbar, click the “Save program file” button, or in the menu bar click “Program” and click on “Save”.

In the lower half of the Save program window, select an index number, enter the name, and enter a description.

The index supports numbers from 0 to 99. The actual number of programs you can save depends on how complex your programs are. Calibration, OCR trained characters, Search patterns (or “models”) add complexity and increase the program size.

If programs are already saved in the BOA Pro program memory, they will appear in the top half of this window. There will be checkboxes you can use to select and delete a program.



## Export Programs

In the Main Window menu bar, click on “Program”. Click on “Export”.

The “Export programs” window allows you to save one or all programs stored in the BOA’s memory to your PC. You can use this to copy programs to a second BOA, or to archive your programs.

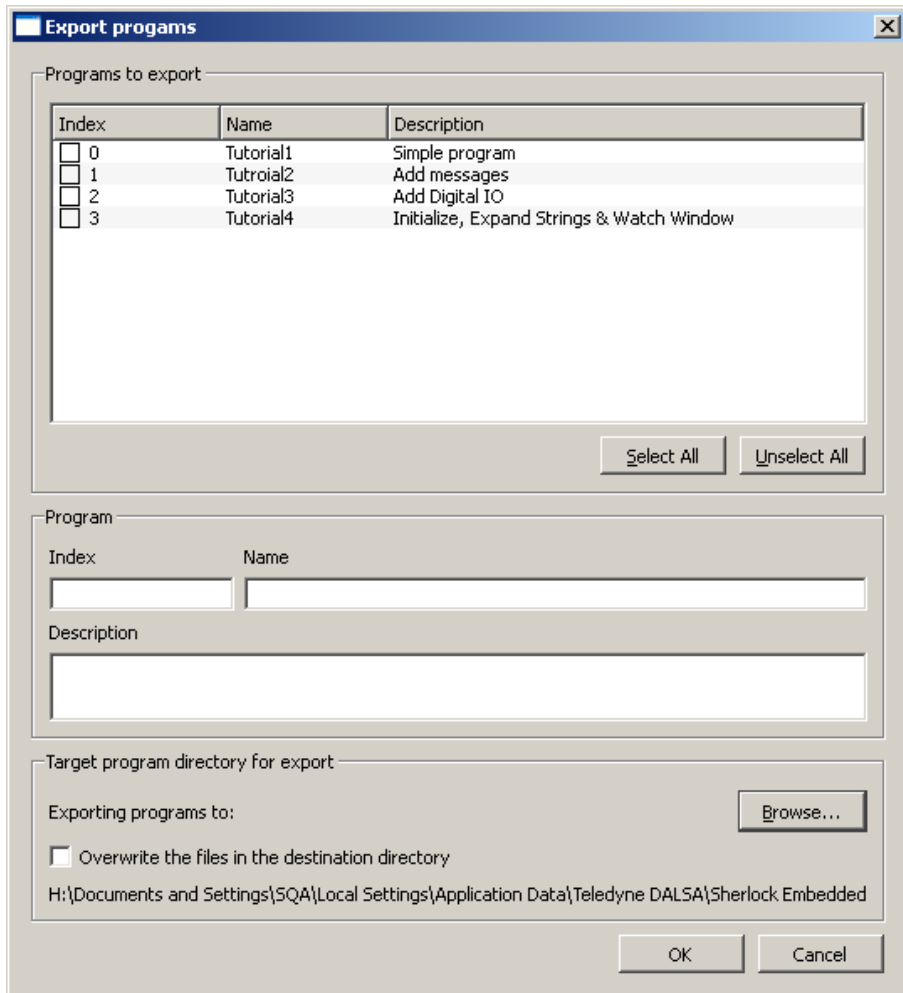
You can use the checkboxes to select individual programs, or use the “Select all” button.

Notice the bottom of this window shows the destination directory. You can use the “Browse” button to change the directory location where programs are saved. The new location will be remembered.

Subdirectories are created in the Target directory. The directory names are the index numbers (0, 1, 2, 3). The program files are saved in these subdirectories.

By default the Overwrite option is not selected. If you leave this box unchecked, numbers will be changed to unused index numbers, each time you export the same program. Use the name and description fields to recognize duplicate programs.

Remember to check the box if you do not want multiple copies of the program.



**NOTE:** By default, the exported programs are saved to your PC using a “User Application Data” environment variable. This variable translates differently in Windows XP and Windows 7. <username> is replaced by the user account name. **The same directory is used when exporting files from the Emulator.** Be careful not to overwrite your programs.

**XP:** C:\Documents and Settings\<<username>\Local Settings\Application Data\Teledyne Dalsa\Sherlock Embedded

**W7:** C:\Users\<<username>\AppData\Local\Teledyne Dalsa\Sherlock Embedded

**NOTE:** The “\AppData” directory may be hidden by default in Windows 7. There is a shortcut in the Start Menu to navigate to the user data files.

**NOTE:** All files under the Program Install directory: “Program Files\Teledyne Dalsa\Sherlock Embedded” are deleted when you uninstall the Sherlock Embedded Client software. Do not save or export programs to this location. Files in the “User Application Data” are not deleted when you uninstall the Sherlock Embedded Client software.

## Import Programs

You can import programs from the Client PC to the BOA Pro program memory. You do not get to change the index when you are importing. Programs with the same index number will overwrite what is stored in the BOA Pro memory.

On the Client PC, the subdirectory name is the index number, and this subdirectory name is used as the index number when you import a program. You can change the subdirectory name/number before importing, to change the index number used in the BOA Pro memory.

In the Main Window click on “Program” in the menu bar. Click on “Import”.

The “Import programs” window allows you to import Sherlock Embedded programs stored on the Client PC, into the storage memory on the BOA smart camera.

The “Import programs” window has a “Browse” button at the top, so you can browse to the directory (Target directory in the Export window) that contains the subdirectories for your programs. (Index numbers.)

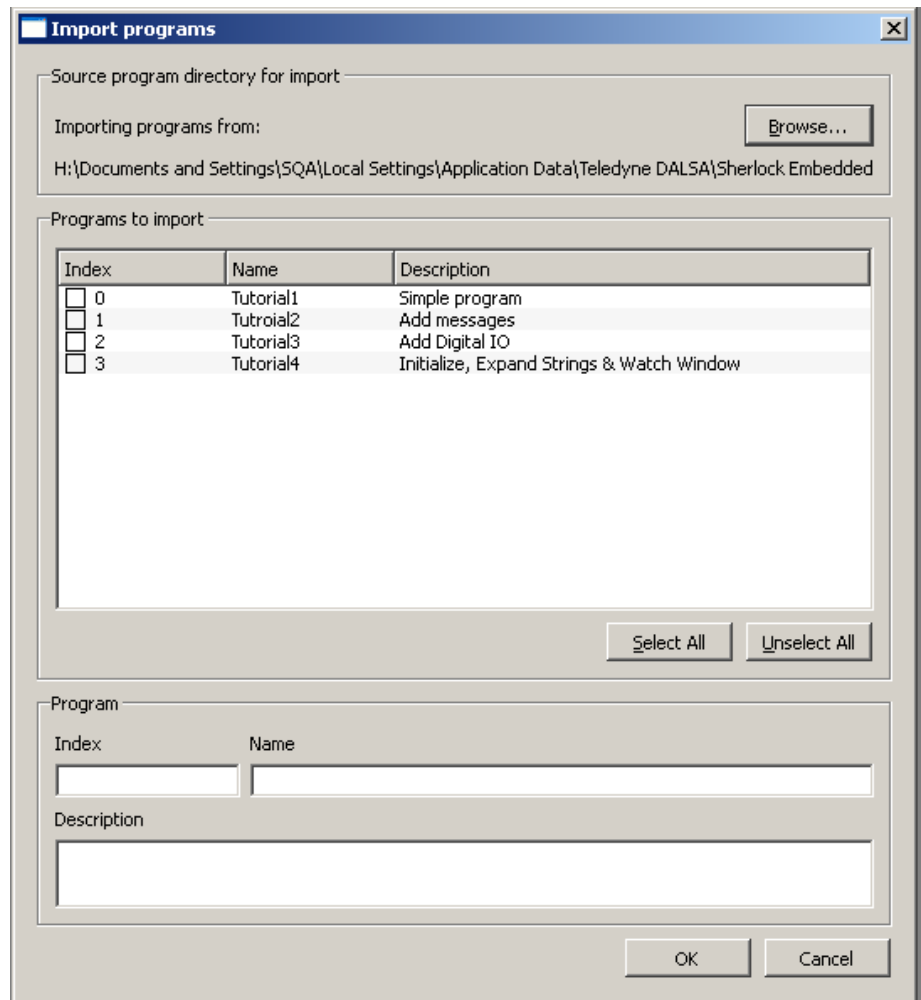
Click “OK” in the “Browse for Folder” window, and the Programs should be listed in the “Programs to import” field, as shown here.

Use the checkboxes to select individual programs, or use the “Select all” button.

The index supports numbers 0 through 99. Larger numbers cannot be imported.

**NOTE:** You cannot change the index numbers as you import programs.

You can change the directory names/numbers using Windows® Explorer before importing. This will change the index number in the Import window, and in the BOA Pro program memory.

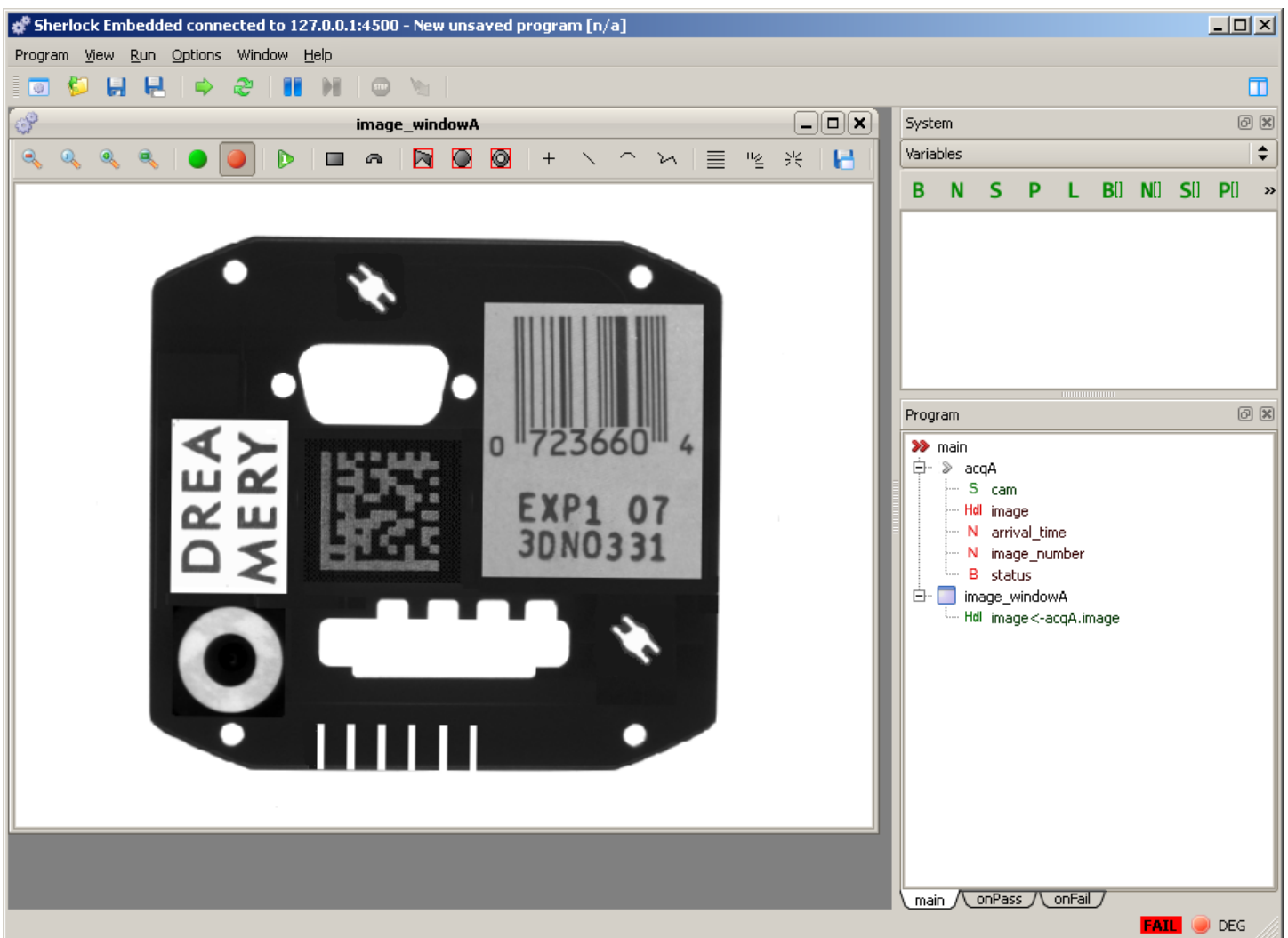
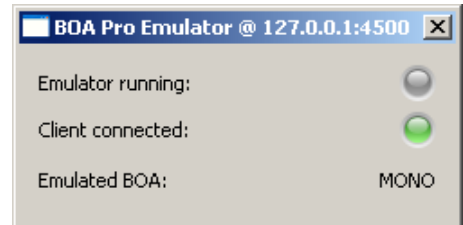


## Sherlock Embedded Emulator

The Sherlock Embedded Emulator is part of the Sherlock Embedded Client software installation. The Emulator can be used to prototype image processing and programs “off-line” without accessing the BOA Pro smart camera. This may be useful if the BOA Pro is already deployed and inspecting product.

Launch the Emulator from the Start menu: Start-Programs->Teledyne DALSA->Sherlock Embedded->Emulator.

When you start the Emulator, two application windows open: a small window titled “Sherlock Embedded Emulator” that mimics the BOA Pro camera, and the Client interface window. The Emulator uses exactly the same interface as the Sherlock Embedded Client, used to access and program the BOA Pro smart camera.



## Emulator images

The Emulator defaults to use images in the “\Sherlock Embedded\Application\images” directory. You can change the directory path in the Application Settings menu, to use your own 8-bit Bit-mapped (BMP) images. In the Emulator or the Client, select: Options -> Application settings and scroll down to the bottom of the page to find “Emulator” and “Image directory”. The entry “./images” is a relative path to “\Sherlock Embedded\Application\images”. You can use a relative path, or a full path (starting with a drive letter and root directory). You do not need to close and reopen the Emulator. The next acquisition will use the new image source.

## Emulator IO

The Emulator reflects the IO and peripherals on the PC. It does not allow you to experiment or prototype IO for the BOA Pro smart camera. The Emulator allows you to experiment with program logic and sequencing, and image processing.

## Saving Emulator Programs

By default, the Emulator programs are saved to your PC using a “User Application Data” environment variable. This variable translates differently in Windows XP and Windows 7. <username> is replaced by the user account name.

**XP:** C:\Documents and Settings\<>username>\Local Settings\Application Data\Teledyne Dalsa\Sherlock Embedded\programs

**W7:** C:\Users\<>username>\AppData\Local\Teledyne Dalsa\Sherlock Embedded\Programs

A subdirectory is created for each program, using the program index number to name the subdirectory.

## Exporting Emulator Programs

By default, the Emulator programs are exported to your PC using a “User Application Data” environment variable. This variable translates differently in Windows XP and Windows 7. <username> is replaced by the user account name.

**XP:** C:\Documents and Settings\<>username>\Local Settings\Application Data\Teledyne Dalsa\Sherlock Embedded\

**W7:** C:\Users\<>username>\AppData\Local\Teledyne Dalsa\Sherlock Embedded\

A subdirectory is created for each program, using the program index number to name the subdirectory.

**NOTE:** The Emulator and the Client use the same directory for Exporting programs. Be careful not to overwrite your programs.

All contents and subdirectories in the program install directory \Program Files\Teledyne Dalsa\Sherlock Embedded directory are deleted when you uninstall the Client/Emulator software. If you wish to save your programs, copy or move them to another directory, not under “\Sherlock Embedded”.



## JavaScript in Sherlock Embedded

All readable and writable values generated by algorithms or instructions, created as variables in Sherlock, or otherwise existent in your Sherlock investigation are available “outside” the Program window through the Sherlock object in the JavaScript Configuration window. This includes the conditional statements for the if/else and while statements, and the JavaScript module itself.

The Property pane on the right side of the Configuration window lists all of the values contained in the VEngine object. To add a value to the Code window of the Configuration window, you can drag-and-drop it from the Property pane.

Or you can type **VEn <Enter>** or **VEn <Tab>** in the Configuration window, and intellisense completes it as **VEngine**.

Enter a “.” and intellisense presents a drop-down list of members of the VEngine object: **rectangleA**, **lineA**, **max\_edgeA**, etc.

Select , for example, **max\_edgeA** and enter another “.” and intellisense presents a drop-down list of properties of the **max\_edge** algorithm: **position**, **distance**, **strength**, etc.

**Warning:** If you type **ven** (lower-case ‘ve’) in a JavaScript window, **VEngine** will pop up, and you can finish the specification as described above. However, JavaScript is case sensitive, so this will generate the runtime error **ReferenceError: Can't find variable: vengine**, because there is no object named **vengine**.

Algorithm and instruction outputs are accessed as **VEngine.alg.output** or **VEngine.instr.output**; for example, **VEngine.add\_numberA.sum**.

To reference one element of an array output, follow the output name by the array index; for example, **VEngine.statisticsA.histogram[32]** or **VEngine.statisticsA.histogram[index]**, where **index** is a number (N) variable.

Variables created in Sherlock are accessed through their value property: **VEngine.varA.value**.

To reference one element of an array variable, follow the output name by the array index: **VEngine.varB.value[index]**

(Variables also have the name property, making it possible to programmatically change a variable’s name: **varA.name = “varB”**. This is probably a really bad idea.)

The **.length** property returns the length of an array.

For an algorithm or instruction output:

```
VEngine.varArrLen.value = VEngine.statisticsA.histogram.length
```

For a variable:

```
Vengine.varArrLen.value = VEngine.varArray.value.length
```

Applying the **length** properly to a non-array value generates NaN (Not a Number).

THIS IS NOT THE SAME AS CHECKING FOR A ZERO-LENGTH (EMPTY) ARRAY.

```
VEngine.strHello.value = “Hello”
```

```
// strHello is a string variable, not an array of strings
```

```
VEngine.varArrLen.value = VEngine.strHello.value.length
```

```
if (isNaN(VEngine.varArrLen.value)){
```

```
  VEngine.IsArray.value = false
```

```

    }
    else
    {
        VEngine.IsArray.value = true
    }

```

Variables created in JavaScript are accessed directly: **index=0**.

Variables created in JavaScript are not directly accessible in Sherlock; they must be written to a Sherlock variable.

### Sherlock arrays in JavaScript

Working with Sherlock references is much faster than with Sherlock objects directly. A reference is resolved only once, and then a native script object is used. If you access a Sherlock object directly, the reference has to be resolved every time.

```

////////////////////////////////////
//    slow !!!!!
////////////////////////////////////
//VEngine.varA.value is a Number array
var idx=100
VEngine.varA.value.length = idx
while(idx-- > 0)
{
    // Reference must be resolved on every write
    VEngine.varA.value[idx] = idx
}

```

```

////////////////////////////////////
//    faster
////////////////////////////////////
//VEngine.varA.value is a Number array
var idx=100
// Reference is resolved once
var na = VEngine.varA.value
na.length = idx
while(idx-- > 0)
{
    // Sherlock array is written to 100 (idx) times
    na[idx] = idx
}

```

```

////////////////////////////////////
//    fastest
// Moving entire arrays in/out to/from Sherlock is the fastest

```

```
////////////////////////////////////  
//VEngine.varA.value is a Number array  
var idx=100  
var na = new Array  
na.length = idx  
while(idx-- > 0)  
{  
    na[idx] = idx  
}  
// Reference is resolved once, entire Sherlock array is written to at one fell swoop  
VEngine.varA.value = na
```

## Operators

### Comparisons

== equal to

!= not equal to

> greater than

>= greater than or equal to

< less than

<= less than or equal to

### Arithmetic operations

+ addition

- subtraction

\* multiplication

/ floating point division (12 /5 = 2.4)

% modulus division. Returns the integer remainder of dividing the two operands (12%5 = 2)

-- (Two minus signs, no space) Unary decrement. Subtracts one (1) from its operand. The operator can be prefix or postfix.

prefix: x = 3 ; y = --x sets x to 2 and y to 2. The prefix decrements the value of x first, then assigns the value of x to y.

postfix : x = 3 ; y = x-- sets y to 3 and x to 2. The postfix assigns the value of x to y first then decrements x.

++ Unary increment. Adds one (1) to its operand. The operator can be prefix or postfix.

prefix : x = 17 ; y = ++x sets x to 18 and y to 18. The prefix increments the value of x first, then assigns the value of x to y.

postfix : x = 17 ; y = x++ sets y to 17 and x to 18. The postfix assigns the value of x to y first, then increments x.

## Logical (Boolean) operations

**&&** (Logical AND) `expr1 && expr2` Returns true if both operands are true; otherwise, returns false.

**||** (Logical OR) `expr1 || expr2` Returns true if either or both operands are true; if both are false, returns false.

**!** (Logical NOT) `!expr` Returns false if its single operand is true; otherwise, returns true.

## Other Topics and Documents

There are several other documents supplied on the BOA PRO CDROM. Some of these are:

*Directory: Documentation*

The **BOA PRO Installation Manual** covers mechanical and electrical specifications and connections.

The **Sherlock Embedded Tutorial** is an illustrated step-by-step tutorial introducing the User Interface, and several programming techniques.

*Directory: Training Self Help*

**Calibration** There are four illustrated step-by-step tutorials on different types of calibrations.

**Alignment** There is a separate illustrated step-by-step tutorial on how to create an alignment, to follow movement of your part in the Image Window.

*Sherlock Embedded Client Software*

The Sherlock Embedded **On-Line Help** installed by the Client Software covers all of the instructions, arguments, system resources, properties, and sections on “getting started” and “how to”.