SmartLink

Installation & User Guide

Documentation Control Number MAN-101-REV-B

Table of Contents

Chapter 1 - Introduction

Welcome to the large family of DVT product users! We are very pleased that you purchased our products and look forward to helping you build a Smart Factory. We at DVT have made a few improvements and changes to the content of this manual to better assist you in your endeavors. We hope that you find this version of the manual both helpful and accommodating. If you have any questions, suggestions, or comments in regards to any of our user manuals, please help us to better help you by contacting our Technical Editor at docs@dvtsensors.com.

How to Use this Guide

The Intellect SmartImage camera Installation and User Guide is your basic resource for both our hardware and software platforms. There are other sources of documentation for Intellect. One such source is the DVT Script Reference Manual, a manual for the programming language used in all scripts in Intellect. Another source is the Intellect HTML Help, the HTML- based help files, which will be installed on your PC's hard disk as you install Intellect. It is also available from both the Help menu and the DVT web site at http://www.dvtsensors.com under the Support section.

You may want to use this guide in one of the following ways with both the SmartLink User Interface and the SmartLink system:

- Read the guide straight through with a SmartLink at hand. Be sure to perform the hardware setup described in Chapter 3 and work through the configuration examples in Chapters 3-7.

- Use the guide and the Hardware Emulator to learn the software aspects of Intellect.

This guide assumes users are familiar with Microsoft Windows.

Structure of this Guide

To make reading this guide easier, tips and hints are highlighted with icons in the margins.

Notes are important information not to be overlooked.

Warnings are specific information that if not followed may result in unexpected results in the operation of your SmartLink.

Hints are extra information that may be useful to you, depending on your application.

Getting More from DVT Software

In addition to the SmartLink Installation & User Guide, you can learn more about DVT software from the following resources:

On-line Documentation

After SmartLink has been installed on a PC, all the necessary documentation is included along with the application software. HTML Help and PDF (Portable Document Format) versions of the written documentation are included.

- PDF Files

  The written documentation provided by DVT includes this Installation & User Guide. In addition, upon installation of Intellect, the Intellect Installation & User Manual and the DVT Script Reference Manual are loaded on your computer and are available on-line in a PDF format. A hard copy of the Script Reference Manual can also be purchased separately through a DVT Distributor. Soft copies of these documents are also included with all SmartLink shipments from DVT.

  The PDF files are created with Adobe Acrobat® and can be both viewed and printed from the free Acrobat Reader software. Acrobat Reader installs as part of a standard Intellect installation.

- Intellect Seminars/Workshops

  DVT offers SmartImage camera products through a worldwide network of Automation Solution Providers (ASPs). Local seminars and workshops can be found on the DVT website. You can find this information on the web under the Education portion of the DVT website. Also, you can contact your local ASP for dates and locations of the next Intellect Seminar/Workshop. These workshops offer an introductory view of both the

SmartImage camera hardware platform and the Intellect software. The ASP Seminar/Workshop is a perfect way to "get your feet wet" on DVT's inspection systems.

- Intellect Training Courses

  Comprehensive four-day training courses are taught at DVT Headquarters. These courses are a great way to get a quick start on learning SmartLink as well as other DVT platforms. This course is free of charge, with users responsible only for their transportation, dinners, and lodging. To learn more and to sign up for a class, please contact your local Automation Solution Provider (ASP) or go to http://www.dvttraining.com to register.

- CD-Rom/Self Paced Training

  Training is also available on a multimedia CD-Rom called the VirtualTour, and is available for free from DVT. The information on the CD-Rom is also available on our website. Visit the DVT web site at http://www.dvtsensors.com and click Request Information and follow the directions to the on-line training area.

Subscribe to the SmartList

Join the long list of SmartImage camera users who regularly discuss topics via DVT's SmartLIST listserver. To subscribe, visit the DVT web site at http://www.dvtsensors.com/support/SmartListManager.php

Discussion topics include (but are not limited to):

- Application questions: "What light source should I try?"
- Intellect software questions: "Which SoftSensor is best for this application?"
- Integration questions: "Has anyone gotten the Series 500 & a XYZ PLC to communicate?"

Smartlist posts are emailed to the listmembers on a realtime basis. For information on SmartLIST, visit the DVT web site, http://www.dvtsensors.com and click "Support".

DVT Certified Automation Solution Provider (ASP)

DVT sells all of its products through a worldwide network of Automation Solution Providers. These ASPs are certified annually based on in-depth knowledge of both Intellect and the SmartImage camera inspection system. If you are unsure of who is your DVT Solution Provider, you can search for that information on the DVT website at http://www.dvtsensors.com/support/find.php. Please contact your local ASP with your technical support questions.

Stay informed about New Releases

Visit the DVT Web site for up-to-date information about forthcoming releases of Intellect. There are two easy ways to reach the web site:

From Intellect, click on the Help menu and choose DVT on the Internet.

Start your web browser and type http://www.dvtsensors.com.

What's New in SmartLink Version 1.2.4

- Support for Touch Screen Monitor Input
- Compatibility with Color DVT SmartImage cameras
- Containers for configuring multiple views on the same SmartLink
- Buttons to allow triggering, inspection control, and custom commands
- Real time feedback can be stopped on an inspection failure
- Static bitmaps can be added for company logos or master images
- Support for the VDX communication protocol

Chapter 2 - What is a SmartLink?

Figure 1: SmartLink Unit

The SmartLink is a software and hardware product that works in conjunction with DVT SmartImage cameras to expand their information sharing capabilities. It allows users to easily create a sophisticated monitoring system that will display inspection images and application specific information from several SmartImage cameras. The customized user interface can be displayed on a PC using the software development user interface or a standard video monitor using the SmartLink hardware package. In addition, certain versions of the SmartLink hardware will serve as a communication bridge to industrial networks like Profibus, or DeviceNet.

Some applications require an operator to monitor images and inspection data during normal operation. Although this functionality has always been available to DVT customers, it involved a dedicated PC running Intellect or FrameWork. In some cases, the extra cost and maintenance associated with this PC were not feasible. In other cases, giving all operators full access to the inspection device via Intellect was not acceptable. In response to customers' needs, DVT developed the SmartLink. It provides the flexibility to monitor inspections in a customized interface and the option to monitor one or more SmartImage cameras without the need for a dedicated PC.

SmartLink and its software are an optional complement to Intellect, but not a replacement for it. Again, SmartLink's primary purpose is to monitor inspections. You will still need to use Intellect with a PC or laptop to configure SmartImage camera inspections.

DVT's SmartLink unit has accessories that vary depending on which model you receive. The model number of your SmartLink unit indicates the configuration for a specific system.

The following accessories come with a typically configured SmartLink package:

- SmartLink Unit (DVT-SL)
- Power Supply (CON-6901)

Depending on your ordering options, our SmartLink may come equipped differently. Below are some common items shipped with the SmartLink unit:

- Serial Adapter (CON-9TM), which converts the RJ-12 connector to a female DB-9 connector for easy connection to a PC

- Serial RJ-12 Cable (CBL6-Cxx4), which supports the serial RS-232 communications. A Category 5, 4 twisted pair, shielded USOC cable should be used.

- Ethernet RJ-45 Cable (CBL6-CxxE), which supports the Ethernet communications. A Category 5, 6 twisted pair, shielded (STP), EIA-568 cable should be used. Use a crossover cable or connector if connecting the SmartLink directly to a PC or SmartImage camera

- System Power Supply (typical part: ACC 24V-NA). If there is already a power supply on a breakout board, then you may simply use the smaller power supply that comes with the unit to connect the SmartLink to that breakout board.

* xx is the length in feet for cables

The SmartLink product is composed of:

1. The SmartLink Hardware Unit

The SmartLink hardware unit is a DIN rail mountable device powered by 24V DC. In its standard version, it has four connectors located in the front of the unit: one RJ-12, one RJ-45, one HD-15, and one power connector. The RJ-45 connector in the SmartLink is an Ethernet port that supports communications via TCP/IP to one or more SmartImage cameras. The RJ-12 connector is a serial RS-232 port used for downloading firmware to the unit and connecting the unit to a touch screen display or a mouse with PS2 to Serial powered converter. As an output, the SmartLink provides a video signal that can be observed with a standard VGA monitor.

2. SmartLink Firmware

Like a SmartImage camera, the SmartLink hardware unit is an embedded system running firmware (RTOS and application). The firmware is downloaded from a PC via the serial port. The application firmware running on the SmartLink has two major functions: It originates and manages all the TCP/IP connections to the different SmartImage cameras and it supports the video functions.

3. The SmartLink User Interface (PC Application)

Rather than providing a simple video output with overlaid text, the SmartLink User Interface allows the user to create a monitoring application that looks like a familiar Windows program. The User Interface also allows the display of multiple images at different sizes and resolutions on the video monitor attached to the SmartLink, originating from one or more SmartImage cameras. In addition to images, other components like tables and text can be displayed. With these capabilities the user can create a monitoring application that has a totally customizable appearance, such as the sample User Interface in Figure 2.

One key feature of the SmartLink User Interface is the “Test Mode”; this feature performs a PC emulation of the SmartLink embedded system. This allows the user to effectively test the project before downloading it to the unit. This process is analogous to developing an inspection using the SmartImage camera Emulator on the PC. Figure 2 was captured in Test Mode. In addition to setting up screens for the SmartLink hardware, the “Test Mode” may also be used as a stand alone application on the PC. This enables the end user to design a secure portal to their DVT systems without running the Intellect application or writing significant amounts of software.

Figure 2: SmartLink Sample User Interface in Test Mode

The SmartLink was also designed to be a communications bridge to industrial networks. This is achieved by adding an Anybus module to certain SmartLinks. These modules are interface cards that provide a link to different industrial networks like Profibus, DeviceNet, ControlNet and many others. The SmartLink will gather data from connected SmartImage cameras and distribute it on the corresponding network.

System Specifications

This information describes the physical properties of the SmartLink Hardware unit and what it needs to work properly.

- **Size:** 54 X 121 X 105 mm.
- **Mounting:** Snaps to standard 35mm DIN Rail.
- **Weight:** 285 grams.
- **Power Requirements:** A regulated and isolated 24 V DC, 210 mA (equivalent to a minimum 5 W power supply)
- **External Ports:** 15 pin high density D Subconnector (VGA output), RJ-12 (RS-232 serial), RJ-45 (10/100 megabit Ethernet, TCP/IP.)
- **Monitors Supported:**

  Color: 640 X 480, 800 X 600, 1024 X 768 @60-85 Hz

  Monochrome: 640 X 480, 800 X 600, 1024 X 768 @60-85 Hz, 1280 X 1024 @ 56 Hz

- **Certifications:** CE Certified.

Dimensions

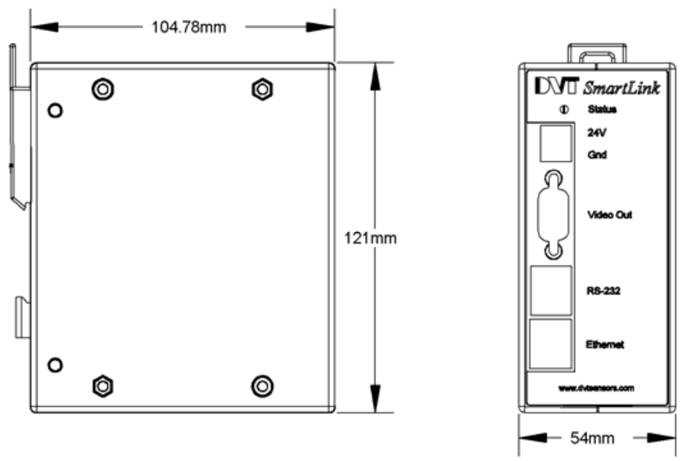


Figure 3: Dimensions of SmartLink

Chapter 3 - Getting Started

The Getting Started section outlines the process of installing the User Interface software and firmware, physically connecting to a SmartLink unit, and communicating with SmartLink.

Unpacking and Powering SmartLink

<table>
<tr><td>  </td><td>Mounting the Unit:

You may wish to mount the SmartLink unit. The SmartLink unit simply snaps to a standard 35mm DIN Rail.</td></tr>
<tr><td></td><td>A screwdriver with a simple twisting action can be used to extend the tab at the back in order to remove the unit</td></tr>
<tr><td>
</td><td>Connect the power supply. Insert the power connector into the socket labeled with 24 V and GND. Wire the screw terminals to a suitable power supply.</td></tr>
</table>

The serial cable connects to the RJ-12 socket labeled RS-232. Connect the other end of the serial cable to the RJ-12 side of the serial adapter.

The standard DVT adapter is gray and should not be confused with the black RS-232 to RS-485 adaptor used with the Series 600 system.

The Ethernet cable connects to the RJ-45 socket labeled **Ethernet**.

Note: If you are connecting the other end of the Ethernet cable directly to a PC (as opposed to a hub, make sure to use either a crossover Ethernet cable (CBL-C10EX) or a crossover connector (CON-RJ45N).

Connect a standard VGA cable to the HD-15 socket labeled **Video Out**. Connect the cable to a supported monitor.

Status LED

The Status LED located on the front cover indicates different states of the SmartLink hardware unit.

Status LED	State
Constant Green	Ready
Flashing Red once/sec	Diagnostics Mode
Flashing Red Rapidly	Booting Phase

Figure 4: Status LED States and Their Meaning

Communicating with SmartLink

In order to establish communications with SmartLink, you will first need to verify the version of firmware in the unit matches the version of the software on your PC. Follow these steps to check the version numbers:

1. If you received a CD with the SmartLink software, install it on to your PC. If you're trying the software or didn't receive a CD, please download and install the SmartLink software from the DVT website. This can be found at http://www.dvtsensors.com in the support section under downloads.

2. Ensure that the SmartLink unit is powered and wait until the Status LED shows a Ready state (Constant Green). If the LED is indicating the Diagnostics Mode state (Flashing Red once a second), please skip to the section on "Loading new Firmware into the SmartLink Unit".

3. If your SmartLink already has SmartLink firmware version 1.2.5 or newer loaded, you can use Intellect to browse to it and set the IP address and subnet. Open any version of Intellect, open Network Neighborhood, and look for devices on the network that have a camera type 'SmartLink'. If none are found then your unit probably has an older version of firmware and you must continue following the instructions in this section. If you do find your device on the list, you can change any network settings through Intellect and then use the indicated IP address to connect through SmartLink software (skip the rest of this section).

4. Connect one end of the serial cable to the RJ-12 side of the serial adapter.

5. Connect the other end of the serial cable to one of your PC's serial (COM) ports using the RJ-12 to DB-9 converter (CON-9TM).

6. Run the SmartLink User Interface by double clicking on the executable file named SmartLink.exe. This file may also be found in the **Start** menu under **DVT Applications**.

7. Click on Help | About SmartLink and note the version of the user interface software. Click the Close button.

8. Select **Remote System** from the **Configure** drop down menu. This brings up the Configure Remote System dialog box.

9. In the Configure Remote System dialog box, expand the Serial node and select the corresponding COM port.

10. Click the **Connect** button to open the Manage Remote System dialog box.

11. Select the System Info tab and note the Firmware Version on the second line of information. If the version matches what was found in step 6, then skip to the section on assigning an IP address. If the version numbers are different, then you'll need to load new firmware into the SmartLink Unit.

Loading new Firmware into the SmartLink Unit

DVT provides software updates for the SmartLink free of charge. Occasionally you may wish to upgrade the firmware in the SmartLink hardware to enjoy new functionality in the system.

Another reason to upgrade the firmware is to ensure that the version of the user interface software matches the version in the hardware unit. To load firmware, follow these steps:

1. Ensure that the SmartLink unit is powered and wait until the Status LED shows a Ready state (Constant Green) or the Diagnostics Mode state (Flashing Red once a second).

2. If your SmartLink already has SmartLink firmware version 1.2.5 or newer loaded, you can use Intellect to browse to it, set the IP address and subnet, and load firmware. Open any version of Intellect, open Network Neighborhood, and look for devices on the network that have a camera type 'SmartLink'.

    a. If your device is listed, right click on it and select 'Load Firmware'. Browse to th e firmware file on your computer. When the firmware download is complete the device status in Network Neighborhood will change back to 'Up'. At this point you must open the SmartLink software to connect to the device. Go to 'Configure Remote System', add your device under the Ethernet section(configure all settings), and click 'Connect'. When prompted to save the firmware click Yes. You are now done and can continue on to designing your application.

    b. If your device is not listed then your unit probably has an older version of firmware and you must continue following these instructions.

3. Connect one end of the serial cable to the RJ-12 side of the serial adapter.

4. Connect the other end of the serial cable to one of your PC's serial (COM) ports using the RJ-12 to DB-9 converter (CON-9TM).

5. Run the SmartLink User Interface by double clicking on the executable file named SmartLink.exe. This file may also be found in the **Start** menu under **DVT Applications**.

6. Select **Remote System** from the **Configure** drop down menu. This brings up the Configure Remote System dialog box.

7. In the Configure Remote System dialog box, expand the Serial node and select the corresponding COM port.

8. Press the **Load Firmware** button. Browse to specify the firmware file to download (*.slf). When the software asks if you wish to reformat flash memory, click "yes". After a 30 to 60 second delay, a message box should appear indicating the progress of the operation.

9. After the download is complete, the "Configure Remote System" dialog box will reappear. Click on the COM port again and click the "Connect" button. A message box will appear, asking if the firmware should be saved to flash memory. Click **Yes** to keep the new firmware after cycling power.

10. If a Message box appears, indicating that the system is not in Diagnostics Mode or that communication is not possible, check cables, port selection and SmartLink Status and try again.

Forcing the SmartLink Unit into Diagnostics Mode

If the SmartLink hardware device crashes or is running an alternate driver on the serial port, it may be necessary to force the system into diagnostics mode before loading firmware.

1. If the SmartLink is running firmware version 1.2.5 or newer, it can be forced into diagnostics from any version of Intellect. Open Intellect and look for it in the Network Neighborhood. If it is listed you can right click on it and select ‘Force to Diagnostics’ and then continue loading firmware from Intellect (skip the rest of this section). If it is not listed continue with this section.

2. Start the Hyperterminal program on the PC. It is usually found under the Start menu in Programs|Accessories|Communications. If you do not have this option, try going to Start | Run and type “hypertrm.exe” in the box.

3. Enter a name for the “Connection Description” and hit OK.

4. On the “Connect To” dialog box, select the drop-down button on the “Connect Using” line and choose the serial port that is connected to the SmartLink. Click OK.

5. On the Port Settings tab, select 38400 for Bits per second, 8 for Data bits, None for Parity, 1 for Stop bits, and None for Flow control. Click OK.

6. Hit the disconnect button on the menu bar. Open up the Properties dialog box under the File menu. On the Settings tab, select vt100 for the emulation option. On the Connect To tab, hit the Configure button. Hit the OK button. This process forces the hyperterminal program to use the settings entered in step 4 rather than “auto detect”. Click OK to close the dialog box

7. Click the Connect button on the hyperterminal menu bar.

8. On the computer keyboard, start pressing the “+” button repeatedly. While doing this, cycle power on the SmartLink Unit.

9. After the system has been powered up for about 10 seconds, stop pressing the ‘+’ key and wait for the system to finish booting.

10. You should see a diagnostics message on the hyperterminal screen and the LED should be blinking red once per second. The system is now ready for the firmware download.

11. Close hyperterminal and continue with the process outlined in the firmware loading section.

Assigning an IP Address to the Camera and SmartLink Unit

The PC and SmartLink unit communicate using the TCP/IP networking protocol. TCP/IP stands for Transmission Control Protocol / Internet Protocol and is based on identifying elements on a network by unique numbers, called IP addresses. In order for two devices to communicate they must reside on the same Local Area Network (LAN), be part of the same subnet, and have unique IP addresses.

All SmartImage camera systems come with a random IP address, which must be changed to a number compatible with the addressing associated with your computer and LAN. An IP address is the equivalent to a telephone number for your computer. The SmartImage camera will assign itself a default IP address. You must then re-assign a new address in order to communicate with your computer. Likewise, in order to communicate using SmartLink, you must assign a correct IP address for the connection as well.

If you need to configure your camera before beginning, please use the following steps to establish correct and valid IP addresses for your camera and computer.

1. Start the Intellect User Interface.

2. Click and Open Network Explorer. Network Explorer is what allows you to connect to and identify cameras in a network. Your computer will search your local network for SmartImage cameras. When Ethernet Adapters is highlighted, the right pane will show the current IP address of your computer. Record your computer's (Host) IP address on a piece of paper. If no camera has been found, wait about 10 seconds and reset the list by right clicking in the right pane. There is sometimes a small delay due to the time it takes for your computer to recognize the camera.

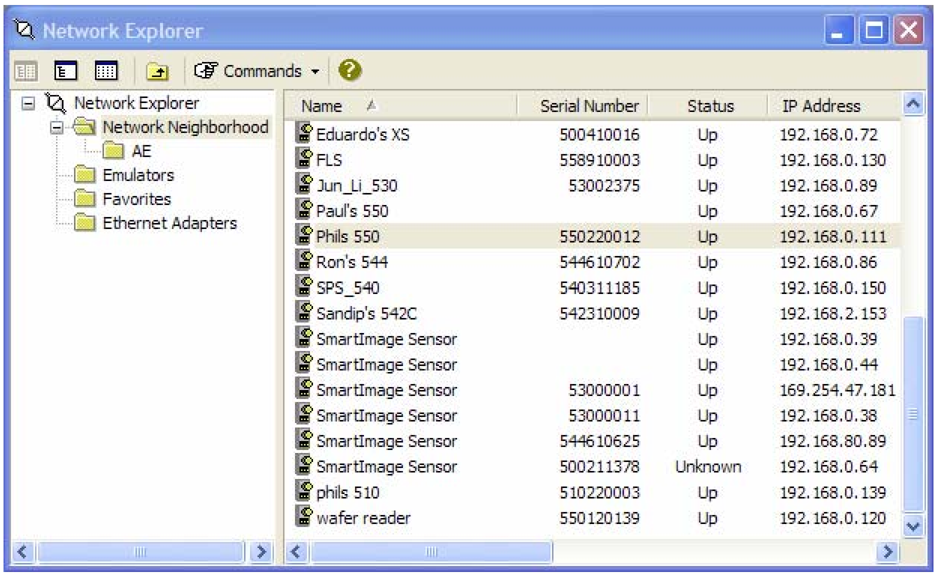


Figure 5: Network Explorer Window

3. Click the camera on which you wish to begin communications.

4. Check the camera's firmware version and Intellect's User Interface software version to confirm that they are the same. The firmware version for the camera will be displayed in a field to the right of the camera name. The User Interface version can be found by opening the Help menu and clicking **About Intellect**. If the two versions are not the same, please load the correct firmware into the SmartImage camera or run the appropriate version of Intellect.

5. With the camera still highlighted, the properties window to the right will display all pertinent camera connection information.

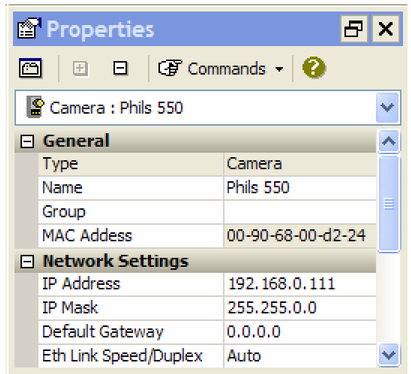


Figure 6: Camera connection properties window

6. In the camera properties window, you can change the camera's IP address, subnet mask, default gateway, name and link speed. Set the IP address to be the same as your computer's IP address with the exception of the last number. For instance, computer: 192.168.0.113; SmartLink: 192.168.0.111. When you click Apply, this information will be permanently stored in the unit's flash memory.

7. If the IP address is already being used by another device, you will se an error in the bottom field of the properties window. Simply go back and edit the systems' IP address once more to a number that is not currently being used.

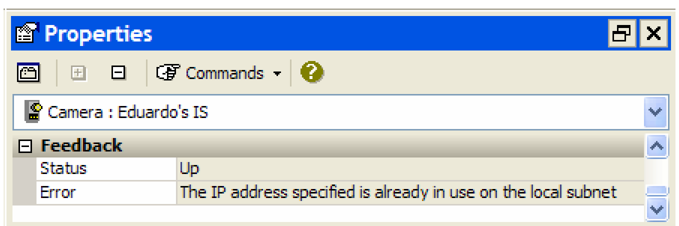


Figure 7: IP Address Error

Hint: After configuring your Camera with its new IP address, you should affix a label to the camera with the Ethernet properties so that you do not forget this important information.

Repeat these steps for each SmartImage camera you have. When complete, the PC and all cameras will have unique IP addresses. Connect all of the systems to a hub (or multiple hubs) and the hub to your PC. You now have a "DVT network."

For more advanced network configurations, you will need to adjust the Sensor Address Mask. For a SmartLink and PC or camera to communicate, they must be on the same subnet. This means that the masked portions of the SmartLink's IP address must be identical to the PC's and camera's. For example, with the default mask of 255.255.255.0 on your SmartLink, only the last digit of the PC's IP address may be different from your SmartLink. In this example, a SmartLink at 192.168.0.242 can communicate with a PC at 192.168.0.200, but not with a PC at

192.168.1.200. The following table summarizes the use of the camera IP Address Mask property with Ethernet networks.

Camera IP Address Mask	SmartLink IP Address	Valid Sample PC IP Addresses
255.255.255.0	192.168.0.242	192.168.0.200, 192.168.0.100, 192.168.0.240

Warning: Whenever connecting a node to an existing network, please exercise extreme caution. Connecting a SmartLink system onto a network without the Network Administrator's knowledge is not recommended.

Hint: If connecting to an existing Ethernet network and unsure about which IP addresses are already in use, choose an IP and Ping that address to see if it is in use. To Ping, open a DOS prompt and type "ping" and then the IP address. If that address is in use, a reply will come back; if not, you will receive "Request timed out."

To configure the Ethernet settings for the SmartLink unit, please use the following instructions:

1. If using SmartLink version 1.2.5 or newer, use Intellect to configure the IP address. Otherwise, continue with this section.

2. Open the SmartLink software by either clicking on the SmartLink icon or by going to the Start menu and selecting DVT Applications. From there, select SmartLink.

3. Select the Configure Menu and highlight Remote System.

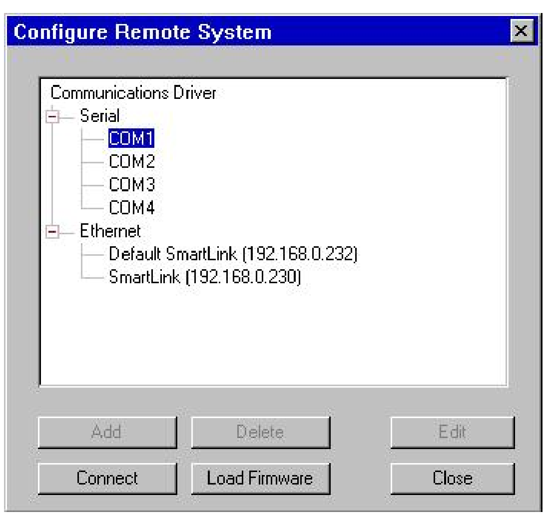


Figure 8: Configure Remote System Window

Note: The Ethernet networking parameters indicate the IP, Mask and Gateway of the SmartLink unit. They do not refer to the PC being used to create the SmartLink project. While in runtime the SmartLink unit uses these settings. While in 'Test Mode' on the PC, the Ethernet networking parameters of the PC are used.

4. After selecting the correct communications port, hit the Connect button. This will bring up the Manage Remote System dialog box.

5. Select the Ethernet tab and enter the correct IP address, subnet mask, and gateway (if used). If you don't know what a gateway is or you're not using one, leave it as all zeros.

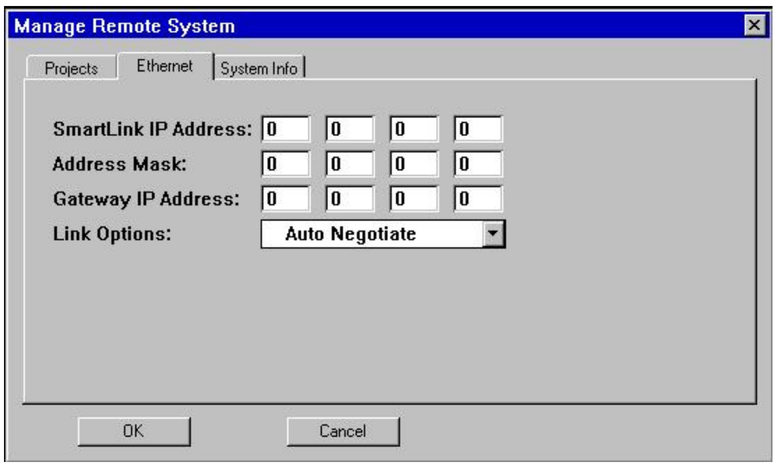


Figure 9: Manage Remote System Window

Hint: In most cases the Link Options can be left as Auto Negotiate. This will allow the SmartLink to automatically determine if the network is running at 10 or 100 Mb/s and if the flow is half or full duplex. If you are experiencing problems with maintaining a connection with the SmartLink and SmartImage cameras, try setting the link options manually. Normally, the 10Mbps, full duplex is the first choice when manually setting the parameters.

6. Click the OK button when the parameters are set. Close Remote System Management box and cycle power to the SmartLink Unit. This step is important in registering the Ethernet parameters within the device.

Once the initial IP address is configured, further communication with the SmartLink unit can be accomplished over Ethernet.

Setting up the SmartLink User Interface

The original purpose for the SmartLink product was to allow users to view SmartImage camera images without running the Intellect software on a PC. Additional features were added to customize the display and provide a table layout for inspection information. These display elements can be shown on a PC running only the SmartLink user interface or they can be downloaded to the SmartLink hardware unit to avoid placing a PC out on the shop floor for monitoring.

This section describes the display elements that will be most useful for a SmartLink configuration without a touch screen or mouse input to a PC running the software. The next chapter will discuss advanced features that require user input to the SmartLink.

When designing a layout for the SmartLink interface, there are 4 elements that can be added to the display:

- Image – A scalable bitmap image of what the SmartImage camera is inspecting. All of the images that the system can send to the SmartLink will be displayed or a filter for passing only, warn and fail, and failed images may be used to simplify monitoring.

- Table – A customizable table of information can be displayed along with the image. The elements of the table are defined in the Intellect interface on each SmartImage camera. The SmartLink accepts the information and displays it in a tabular format.

- Text – Custom text describing the station or the image can be added to the display. The text is static and is designed for descriptive purposes only.

- Bitmap – A fixed bitmap or JPEG image can be added to the workspace at design time. This can be a company logo or a master image of a good part.

Step One: Determine the Layout for the User Interface

The first thing you need to do is determine how you want to use DVT's SmartLink. How many cameras do you want to monitor? What types of inspections do you want to monitor? What do you want to allow the operator to do? What is the skill set of the operator? Will you need to add instructions or golden images to the User Interface for the operator? These questions will require some thought and should be fully addressed before beginning the User Interface design process. You need to have a good idea of how you intend to use the interface and who will be using it in order to create an effective one. Proper planning for the SmartLink project will prevent a great deal of backtracking in the future.

Step Two: Familiarize Yourself with the User Interface

Once you start the SmartLink User Interface, you will get the screen shown below in Figure 10. It is important that you familiarize yourself with the content of the interface before going further in these instructions.

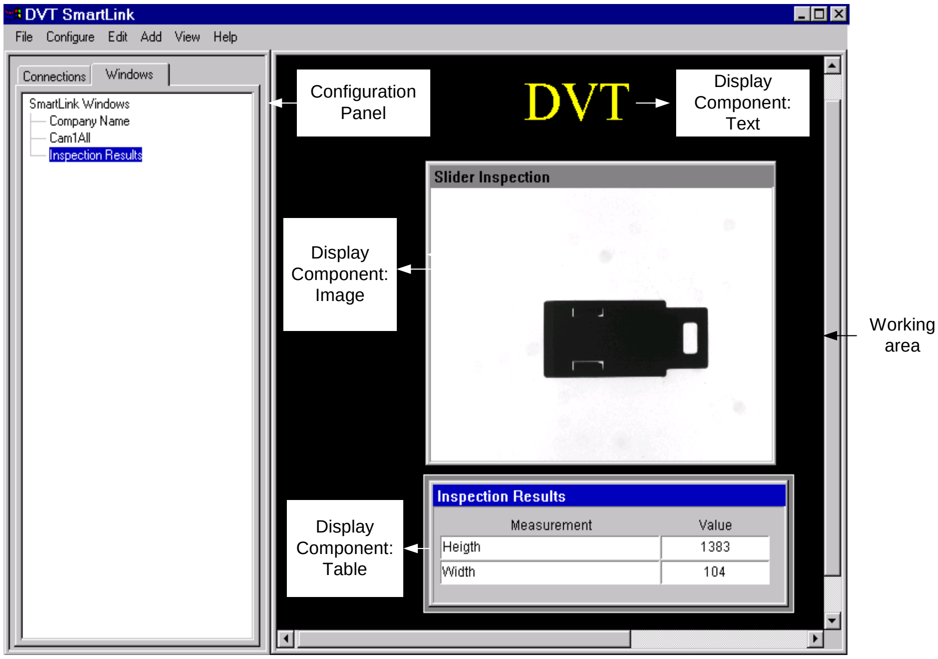


Figure 10: SmartLink User Interface

Main Window-Working Area

The main window in the SmartLink User Interface shows a view of the target display. We call this view the working area. The monitoring application, with its display components, can be seen through the working area as the project development progresses. Scroll bars are provided to allow a user to develop applications for a target monitor with higher resolution than the monitor used to create the project. A typical case would be creating a SmartLink project intended for a 1024 X 768 display using an 800x600 laptop.

The Configuration Panel

The configuration panel is a SmartLink User Interface window that can be accessed from the Configuration Panel selection from the View drop down menu. The panel is shown by default and making the menu selection for the first time will hide it. The user can show the window again by repeating the menu selection. The configuration panel contains two panes labeled Connections and Windows.

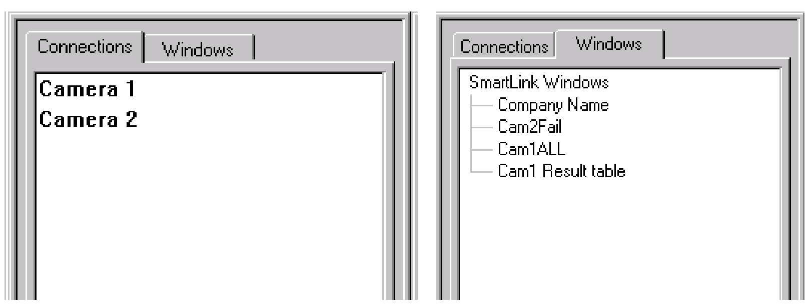


Figure 11: SmartLink Configuration Panel

In the Connections pane, the user can see a list of camera connections that have been configured for the current project. The list provides an easy way to access connections in order to view or change settings during project configuration.

In the Windows pane, the user can see a list of display components that, together with the connections, make up the current project. All display components are windows with different appearance and functions. The list of components inside the pane provides an easy way to access them for editing and deleting during project configuration.

Working with Connections and Display Components

You may manipulate the connections and display components very easily. Here are some suggestions for manipulation of the components:

To Add a Connection, Select Connection from the Add drop down menu. This brings up the Connection Configuration dialog box for that connection.

To Edit an Existing Connection, highlight it in the Configuration Panel and select Properties from the Edit drop down menu. This brings up the Connection Configuration dialog box for that connection.

To Delete an Existing Connection, highlight it in the Configuration Panel and select Delete from the Edit drop down menu.

To Edit an Existing Display Component, highlight it in the Configuration Panel or single click anywhere in the window. Then select Properties from the Edit drop down menu. This brings up the appropriate configuration dialog box for that component.

To Delete an Existing Component from the current project, highlight it in the Configuration Panel or single click anywhere in the window and select Delete from the Edit drop down menu.

To Add a Display component Select Image, Table or Text from the Add drop down menu. This brings up the Configuration dialog box for that component.

Step Three: Configure Project and Remote System Parameters

Now that you know where the configuration information resides within the SmartLink software, it is time to make the necessary configurations to the User Interface. Configuring the project parameters involves defining the characteristics of the target display through the SmartLink User Interface. Under the Configure menu, there are two options: Project Parameters and Remote System. The Project Parameters selection refers to the project that is currently being developed in the SmartLink User Interface. You want to configure this box to indicate the characteristics of the target monitor, serial port, and Fieldbus. This is done in the following dialog boxes accessed by selecting Project Parameters from the Configure menu:

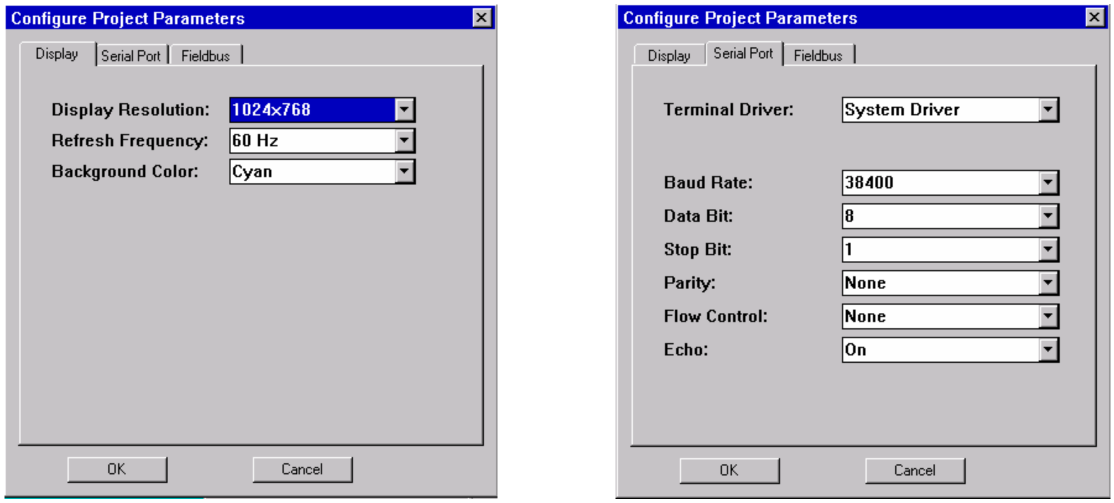


Figure 12: Project Parameters Display and Serial Port Parameters

In the Display tab, the display resolution refers to the resolution of the target monitor. It is not related to the resolution of the monitor attached to the PC used for creating the project. The other two parameters in this tab are the refresh rate of the target monitor and the desired background color.

The next tab is the Serial Port tab. Here, you can choose which Terminal Driver you need to use. Typically, you will use the System Driver. However, if you are using a MicroTouch or ELO touch screen monitor or a powered mouse device, you will need to select that particular driver in this tab. This tab also enables you to configure the Serial connection as you would in HyperTerminal.

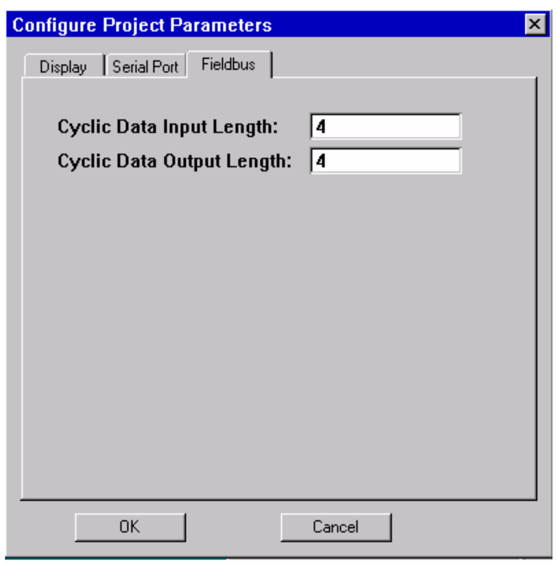


Figure 13: Fieldbus Tab of Project Parameters

The Fieldbus tab enables you to configure the Cyclic Data Input/Output lengths in the event that you are using Profibus or Devicenet. Please refer to chapters 6 and 7 for more information on these methods of communication. The default values are 4 for each.

The other option in the Configure menu is the Remote System. As mentioned before, the Remote System dialog box is used to connect to SmartLinks and/or upgrade the firmware.

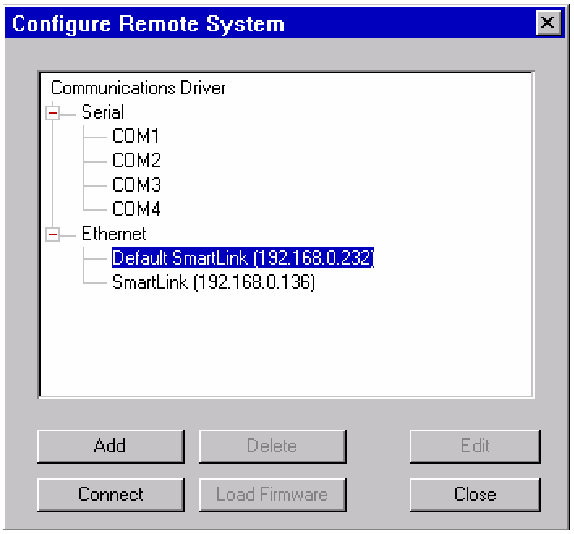


Figure 14: Configure Remote System via Ethernet

This tree view contains nodes that represent the two types of ports the PC can use to communicate to a SmartLink: Serial and Ethernet. Upon expanding the serial node the different COM ports available on the PC are shown. Upon expanding the Ethernet node a list of target IP addresses is shown. Unlike the Network Explorer feature in Intellect, the SmartLink Ethernet entries must be added manually to the system. Use the Add and Delete Button to manage this list.

Remote system management requires that a connection be established between the PC application and the SmartLink. Once a node in the Communications tree view is highlighted, the Connect button is enabled allowing the user to make a connection to the selected IP address or COM port. In the case of selecting a COM port, the user also has the option of clicking Load Firmware to download firmware from the PC. After a connection is established, the Manage Remote System dialog box appears. This box allows you to download the final SmartLink project to the hardware unit after it has been created on the PC.

Step Four: Make All SmartLink User Interface Configurations

Connections

After all of the Project Parameters have been set, it's time to develop the User Interface. The Add menu is the next stop in the development process. The first thing we want to do is add Connections. The SmartLink software can communicate to a maximum of 16 different SmartImage cameras on the same network. For each camera that is going to send an image or data to the SmartLink, a connection must be added to the project. An example is shown for you in Figure 15. Complete this step for every camera you need for the project.

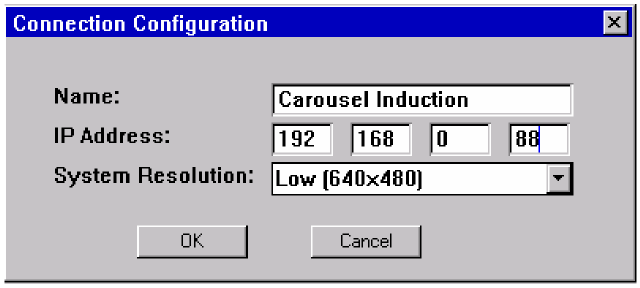


Figure 15: Connection Configuration Under Add Menu

For each connection that is added, the SmartLink will attempt to create a separate TCP/IP connection to a SmartImage camera when the project runs. It is useful to think of connections as dedicated pipes of two-way data flow that are opened between a SmartLink and SmartImage cameras. There may be one or more connections to a single SmartLink. There may also be several connections to different SmartLinks. If a connection is lost during operation, the SmartLink will try to reestablish it automatically. Some display components like images and tables use connection names to determine where their information comes from.

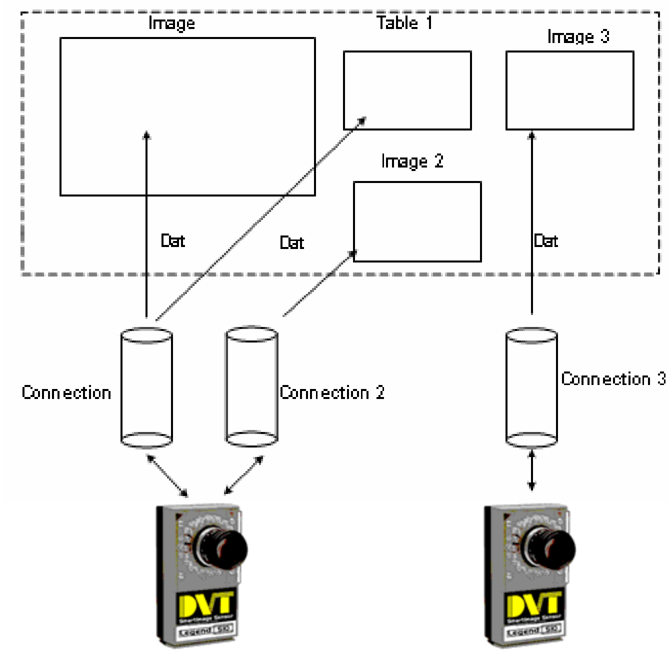


Figure 16: Schematic to Illustrate Connections

Images

Once you have all of your cameras added, you may add an Image to the User Interface. Adding an Image is what will enable you to see what the camera is inspecting. Once you add an Image to the User Interface, a parameters box will pop up for you to configure. An example of the box is shown in Figure 17.

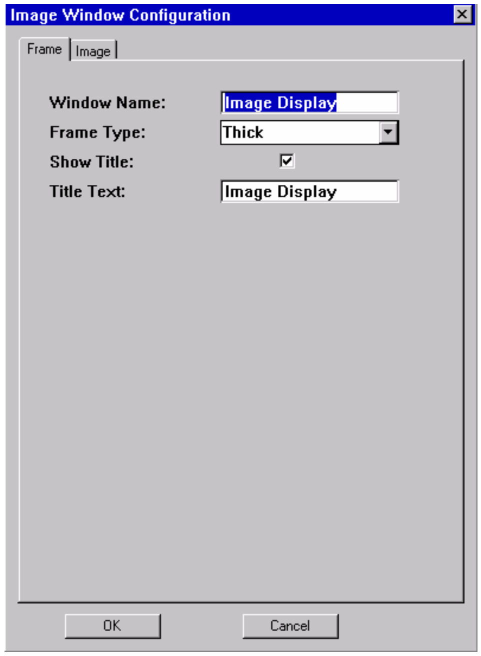


Figure 17: Frame Tab of Image Parameters Box

The first tab in this parameters box enables you to name the Image and provide for it a frame if you wish. Note that you are prompted to provide information for the Window Name and the Title Text in this tab. The Window Name is what will appear on the left side of the User Interface, in the Configuration Panel. Name this in such a way that it will mean something to you later. The Title Text is what is going to be displayed in the title bar of the actual image. You may choose not to have the title show by simply unchecking the Show Title box in this tab.

The second tab in this parameters box enables you to configure the Image itself. Here, you can choose what size and resolution it will be. Figure 18 shows the Image tab associated with one of the Images in the sample User Interface.

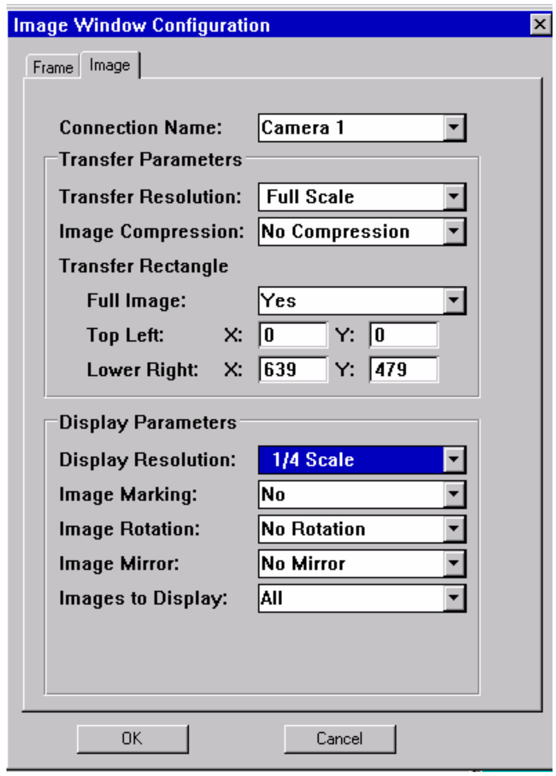


Figure 18: Image Tab of Image Parameters Box

One of the first things you want to do in this particular tab is to assign a camera connection to the Image. This will allow the proper Image to be displayed in the SmartLink User Interface. After you have selected a camera, you need to decide how you want your Image to look. You can do thigh through the Transfer Parameters and the Display Parameters.

Transfer Parameters

The Transfer Parameters are what determines what portion of the Image will be coming from the SmartImage camera to the SmartLink. You have several means of configuring the size of the image in this parameters box. The Transfer Resolution determines the resolution of the Image to be transmitted across the connection. You have a choice of Full Scale to 1/256th of the actual Image. Next, you can configure the Image Compression, which will allow for the compression of Images when/if only a small amount of bandwidth is available. You have a choice of No Compression and 2 to 8 bit compression.

Hint: On a small local network, it is more efficient for the camera and SmartLink to use no compression. For sending images over the internet or a large plant network, compression can be used to take less bandwidth and actually speed up image transmissions.

Finally, you can configure the Transfer Rectangle, which will send only the partial image within the designated X and Y coordinates.

How Images are Transferred - Speed of Image Updates

In addition to the Transfer Rectangle parameter, the speed of the image updates is dependent on two other image parameters: the Transfer Resolution and the Image Compression. As mentioned previously, the Transfer Resolution parameter determines the resolution of images to be transmitted across the connection. A setting lower than Full Scale causes less image data to be sent from the SmartImage cameras. This considerably increases the speed of the image updates at the price of lower resolution images on the monitor. For instance, if a Transfer Resolution of ¼ scale is used, then only half of the image's rows and half of the image's columns are sent across the connection. This Data is reassembled by the SmartLink to form a lower resolution image. The image compression function can take more time on the SmartImage camera CPU than sending the entire image on a small network. Compression should only be used when operating over a large plant network or the internet.

The network configuration and hardware used also affect the speed of image transfers. Setting up a dedicated subnet with full duplex, switching hubs will give best results.

Display Parameters

The Display Parameters are used to manipulate what's actually in the window itself. The Display Resolution parameter determines the resolution used to show the image inside the window. The combination of the Transfer Rectangle and the Display resolution determines the size in screen pixels of the Image Window. The Display Resolution parameter does not affect the speed at which images are updated. A lower Display Resolution results in a grainer view of the image but the amount of image data that is transferred across the connection does not change. The Transfer Rectangle does affect the speed of image transfer because less data is sent across the connection if a region smaller than the CCD size is specified.

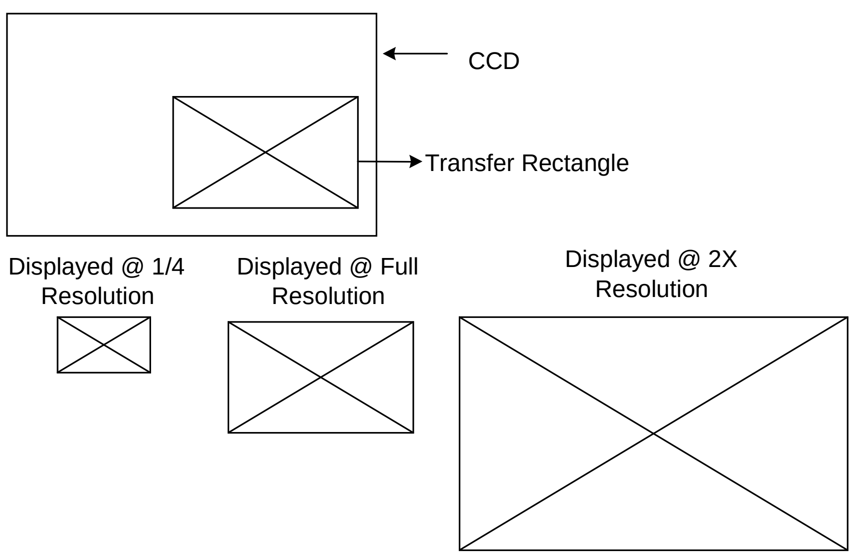


Figure 19: Illustration of Transfer Rectangle and Display Resolution Parameters

The Image Rotation parameter provides 4 possible orientations for displaying the image. They are spaced at 90-degree increments and are illustrated in Figure 20.

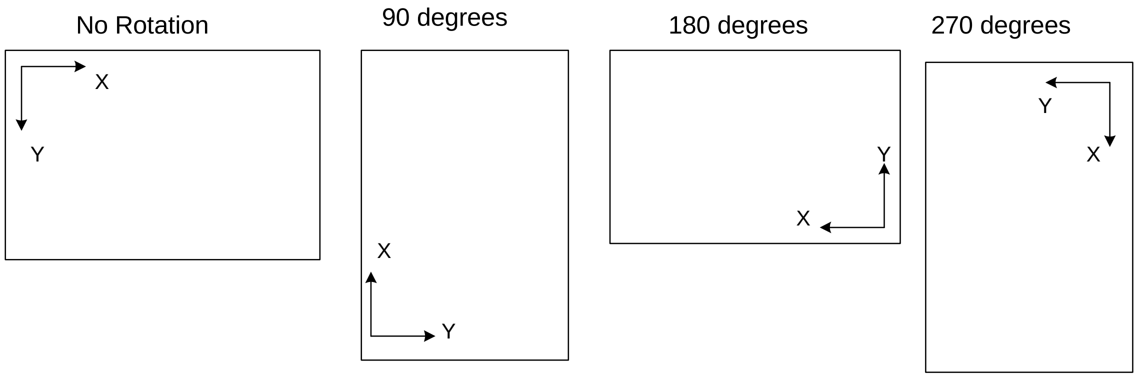


Figure 20: Illustration of the Image Rotation Parameter

The Image Mirror parameter allows the user to mirror the image about the horizontal and vertical axis.

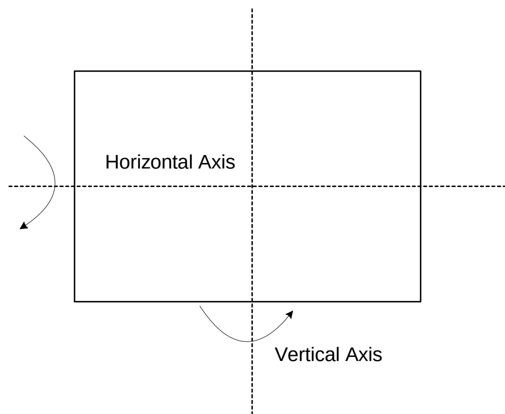


Figure 21: Illustration of Image Mirror Parameter

The user can decide if SoftSensor marking is displayed through the Image Marking parameter.

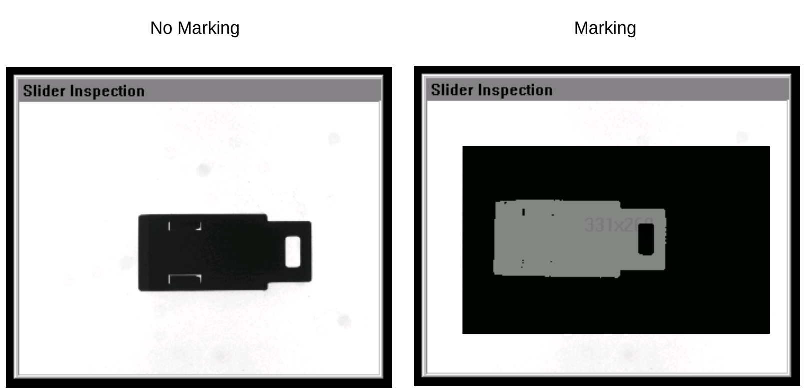


Figure 22: Illustration of the Image Marking parameter

Display Resolution vs. Transfer Resolution

Display Resolution and Transfer Resolution should normally match. There will be cases where a Display Resolution higher than Transfer Resolution is needed. This would produce larger sized image displays but with lower quality. The reason for this is that the SmartLink needs to scale the image data to fit the larger display area. This results in a grainy appearance since one image pixel is mapped to several screen pixels. The benefit of this configuration is that less information is sent to the SmartLink, thereby speeding up the image transfers. The Opposite case, Display Resolution lower than Transfer Resolution, is also allowed but wasteful. In this case, the camera is sending a lot of information to the SmartLink, but only a fraction of it is displayed. If only one image is attached to the connection, having a Display Resolution lower than Transfer Resolution will result in a lower quality image at an update rate slower than possible (see next section).

Multiple Images: One connection vs. Multiple Connections

A fundamental concept to remember is that more than one image can be associated with a single connection. This is a feature that allows great flexibility to the user. Different types of images from the same SmartImage camera can be viewed at different sizes and resolutions without the bandwidth overhead of several connections. For instance, a project could display every image from a SmartImage camera in a small, low-resolution window and also display its failed images in a full-resolution, full-size window.

When there are several Image Windows attached to a single connection, the Speed of the image updates is governed by the highest setting of the Transfer Resolution parameter in any of the corresponding images. Consider a SmartLink project that has 2 Image Windows attached to a single connection. If one of the Image Windows has a Transfer Resolution of Full Scale and the other 1/4 Scale, the speed of the image updates on both would be the same and equal to the speed for a full image transfer. This is because several Image windows attached to the same connection work from the same image data. The SmartLink connection knows that one of the Image Windows requires full resolution so it asks the SmartImage camera for complete images. Once the image data is in the SmartLink, it is displayed in all the Image Windows associated with the connection according to their display parameters.

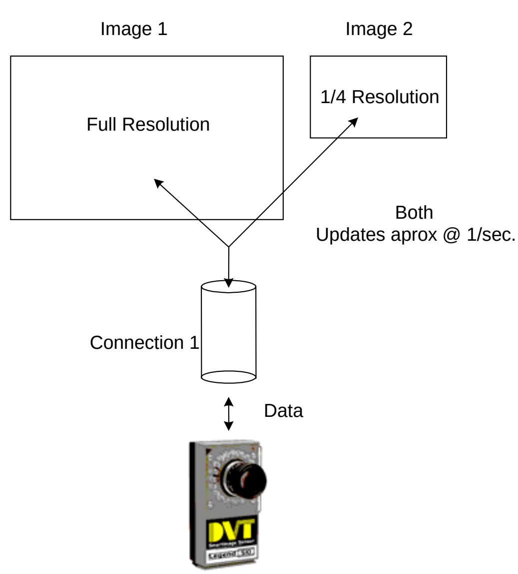


Figure 23: Case of 2 Images Attached to the Same Connection

In cases where a fast update, low-resolution image and a slower update, high-resolution image are needed from the same SmartImage camera, two separate connections should be used. In this situation individual Image Windows should be attached to each. The bandwidth of the network is the only limiting factor as the number of connections increases.

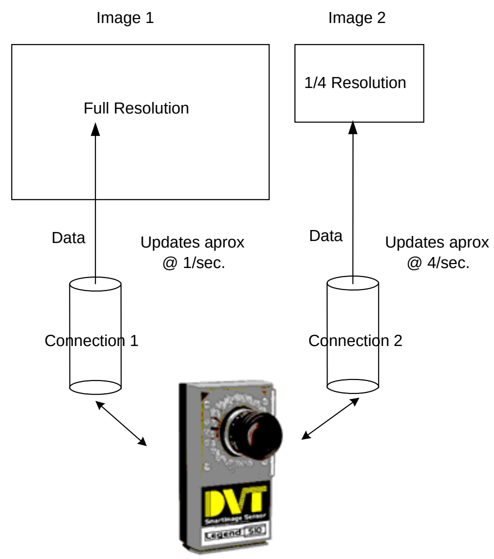


Figure 24: Case of 2 Images Attached to Two Connections

Types of Images

The type of images displayed in an Image Window is controlled by the Images to Display parameter. The options are All, Pass Only, Failed and Warned, and Failed Only. These options are identical to the ones presented in the Intellect Display Parameters dialog box (warn only applies to FrameWork). This parameter offers a number of useful possibilities for the user. For instance, two Image Windows attached to the same connection can be configured in a project. One could be configured to show all images and the other could be configured to show only failed images. With this scenario, the operator would be able to monitor the inspections and observe the last failed image simultaneously.

Tables

The next option available to configure through the Add menu is a Table. If there are System, Product or Sensor settings and outputs from Intellect that you would like to carry over to the SmartLink User Interface, then you will need to add a Table to your SmartLink project. Tables are display components that show settings and data from the SmartImage camera. Unlike images, tables require configuration steps in both the SmartLink User Interface and Intellect User Interface.

A table is essentially a custom Result Table (See Intellect User Guide) in that the information shown is only a sample of the total inspections. Again, each table is associated with a connection (one SmartImage camera) as the source of its data. It is also important to remember that more than one table can be associated with a single connection. The information for a particular table is synchronized with all the images that are attached to the same connection. Figure 25 illustrates a Table used in a SmartLink User Interface.

Warning: If two or more images and a table are attached to the same connection, and each image shows a different type (i.e. one shows Pass only and the other shows Fail only), the table information will update with every image update on any of the images. If needed, add a connection per image type and associate a table with each.

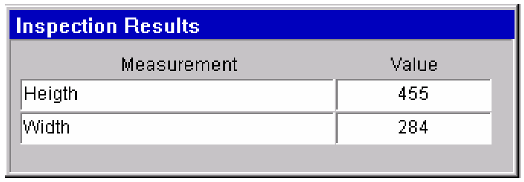


Figure 25: SmartLink's Table Display Component

Table Window Appearance

There are a number of options associated with a Table Window. The user can choose the type of frame that surrounds the Table Window, its title and whether to show a Title bar. The following dialog tab is used to enter some of these options.

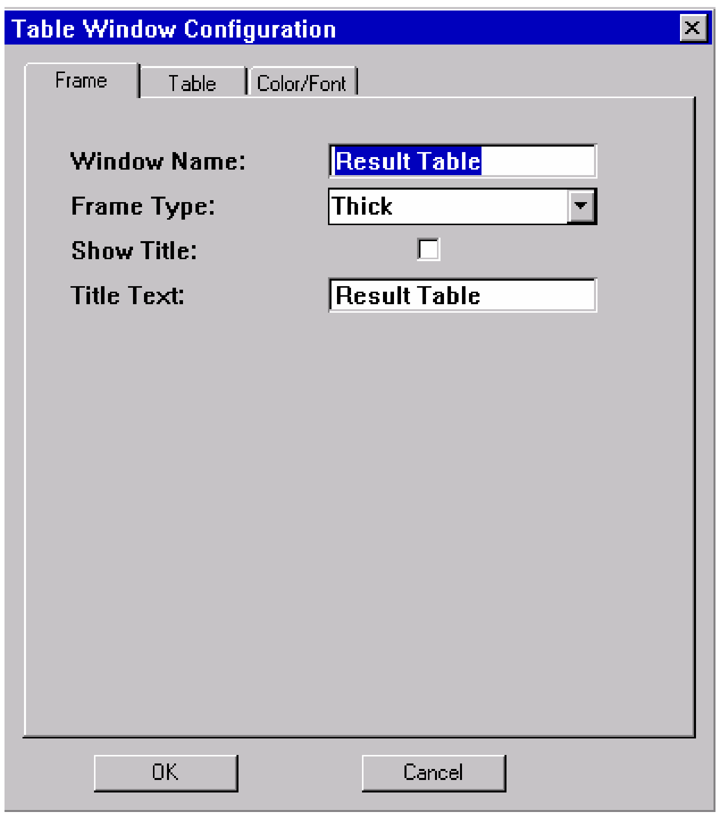


Figure 26: Table Window Configuration Dialog

The Window Name is used to identify the Table Window in the SmartLink project, while the Window title or Title Text is a label displayed on the title bar. They may be different if needed. For instance, it may be useful to name the table with a reference to the connection name to aid readability during project configuration. The Table Window title may be something descriptive for the operator and specific to the application. For instance, a Table Window may be named “Cam1Results” and titled “Gear Tooth Measurements”.

Table Configuration

The Table tab contains the parameters to configure the size and heading labels of the table and the mapping to the Intellect SmartLink SoftSensor in the DVT system. As mentioned before, the connection name is used to determine the source of the data; recall that a specific SmartImage camera is associated with every connection.

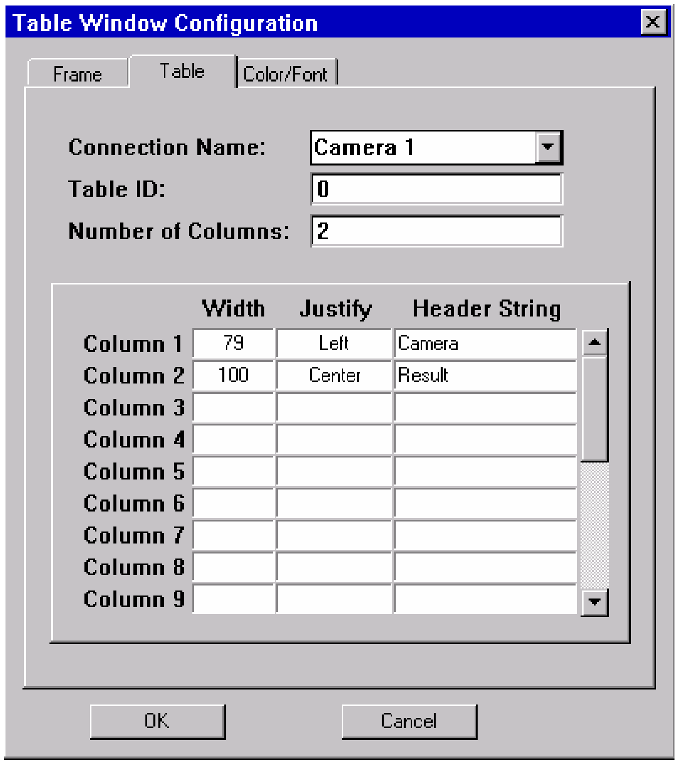


Figure 27: Table Tab of Table Window Configuration Dialog

How Tables are Created

The data shown in a SmartLink table is specified inside the SmartImage cameras via the Communications Explorer. Create a table in Intellect by selecting SmartLink in the Communications Explorer and double clicking in the right hand pane. Click on the newly created table name to view the table’s cells. Each cell is displayed as a position in a matrix of base one, thus the top left table position, Cell (1,1), is the cell in row 1, column 1, and Cell (2,1) is the cell in row 2, column 1.

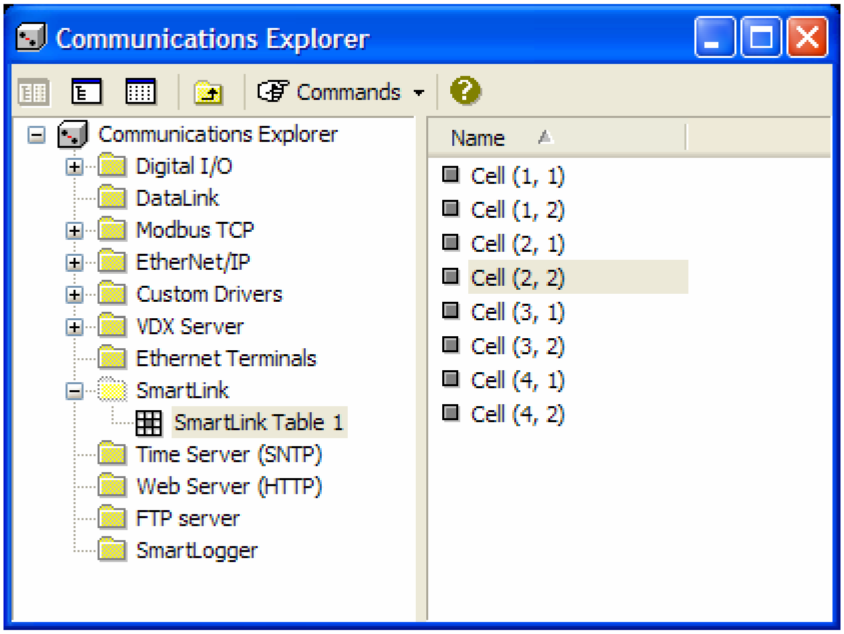


Figure 28: Create a SmartLink table in the Communications Explorer

Clicking on the SmartLink table's default name in the left hand pane of the Communications Explorer displays the table properties in the Properties window to the right. This is where the table name, size and ID can be changed. Table size is determined by rows and columns of base one, so a row size of 4 creates a table with rows 1-4, not 0-3. The Table ID is used to link a particular table's data in Intellect to the table you create in the SmartLink user interface. Note that this Table ID must be the same as the ID of the table included in the SmartLink project. The information in the Intellect table identified in figure 29 will thus be used in the SmartLink project shown in figure 27.

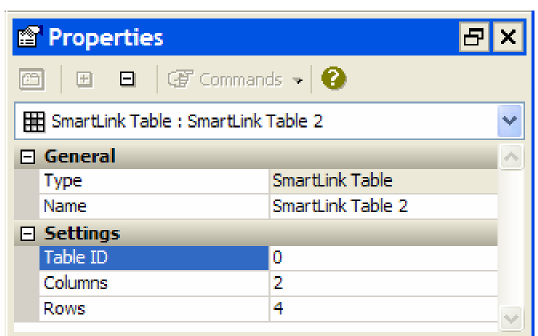


Figure 29: SmartLink Table Properties

Table Cell Population

Clicking on any of the table's cells, displayed as Cell (col,row) in the right pane of the Communications Explorer, will display that cell's properties in the properties window. There the currently selected cell is identified by its row and column position. Also shown are settings for the information that will be displayed in the cell and how it will be formatted. The Fail character is the character that will be displayed in the event that there is no data to send due to a failed inspection. The default is the pound character '#'. The Place holder determines how numbers shorter than their specified format length are filled. The options are 'spaces' and 'zeroes'. For instance, if a number is formatted to display 5 characters it may display 00173.0 rather than 173.0.

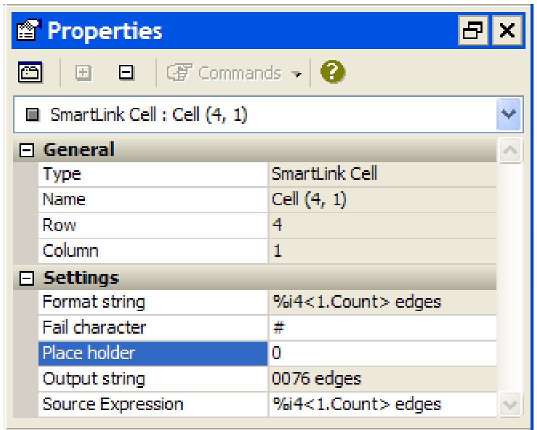


Figure 30: SmartLink Table Cell Data

To specify the information that will go into each cell, click in the area to the right of Source Expression. This brings up a familiar looking dialog box called the String Expression Editor. This editor can also be opened by double clicking on a cell in the Communications Explorer window. Just as in DataLink's Strings tab, this dialog provides a tree view of the system including system parameters and the current Product and its sensors. The user can then type any string intermixed with the data from the system tree. The resulting output string as well as its source expression and formatting are displayed in fields of the properties window.

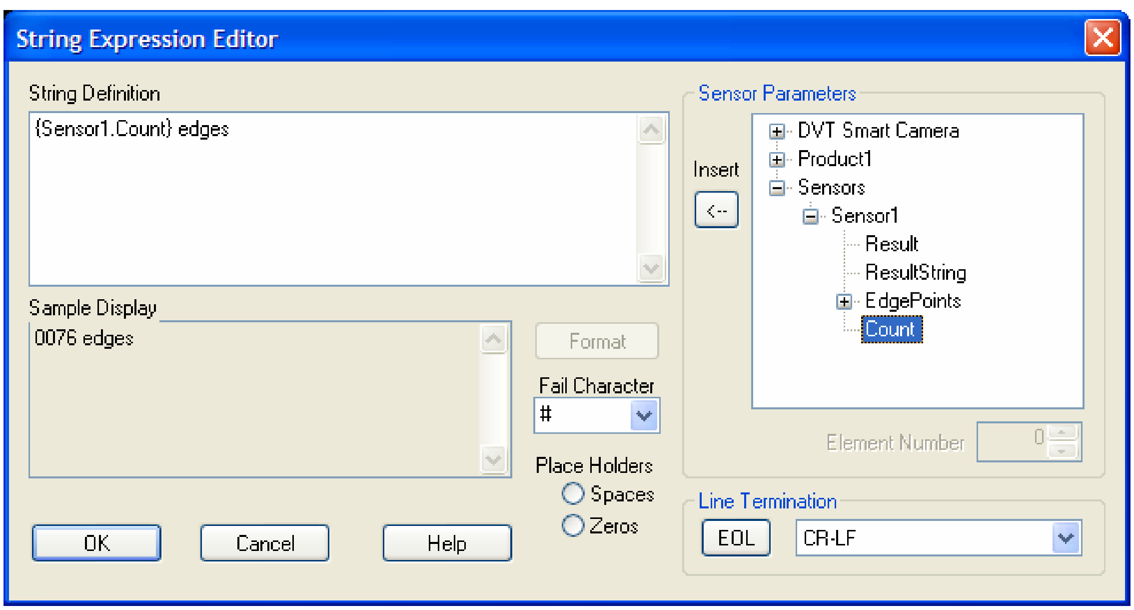


Figure 31: Intellect's String Expression Editor

For instance, if an integer with default formatting is specified, then a 76 will be shown. Keep in mind that a static string can also be typed in by the user. Consider that the String parameter of a Script SoftSensor can also be referenced here. This allows the user to access any information that resides in the DVT registers and display it in a cell inside a SmartLink table component.

Warning: The size of the table (# of Rows and Columns) specified in the SoftSensor must be equal to size of the table display component specified in the SmartLink project. Any discrepancies will result in truncated information or blank cells.

Warning: In a given Product in Intellect there may be more than one SmartLink SoftSensor. Each SoftSensor must reference a different table in a SmartLink Project (via different Display ID's). If two SmartLink SoftSensors reference the same SmartLink table, information will be lost or truncated.

Warning: The information displayed in the tables is only a sample of the total inspections. This information is the same type as the Result Table in Intellect's User Interface. Although the data is synchronized with the images attached to the same connection, an update is not necessarily made with every inspection. In this respect, the information shown in SmartLink tables is different from DataLink.

Text and Bitmaps

Once you have added Connections, Images, and Tables, you probably want to add some finishing touches to the User Interface. The SmartLink software enables you to add both bitmaps and text to the User Interface.

The text display component allows the user to include any static text string as part of the display application created through the SmartLink project. For instance it may be desirable to place a text string with the name of the machine or identifying a specific line number at the top of the display.

Figure 32: SmartLink's Bitmap and Text Display Components

Text Window Appearance
Text is added by selecting Text from the Add drop down menu. This brings up the Text Window Configuration dialog box. There are a number of options associated with a Text display component. The user can choose the type of frame that surrounds the Text Window, its title and whether to show a title bar. The following dialog tab is used to enter some of these options. In the case of the Text display component the defaults are set to not show a frame and a title bar.

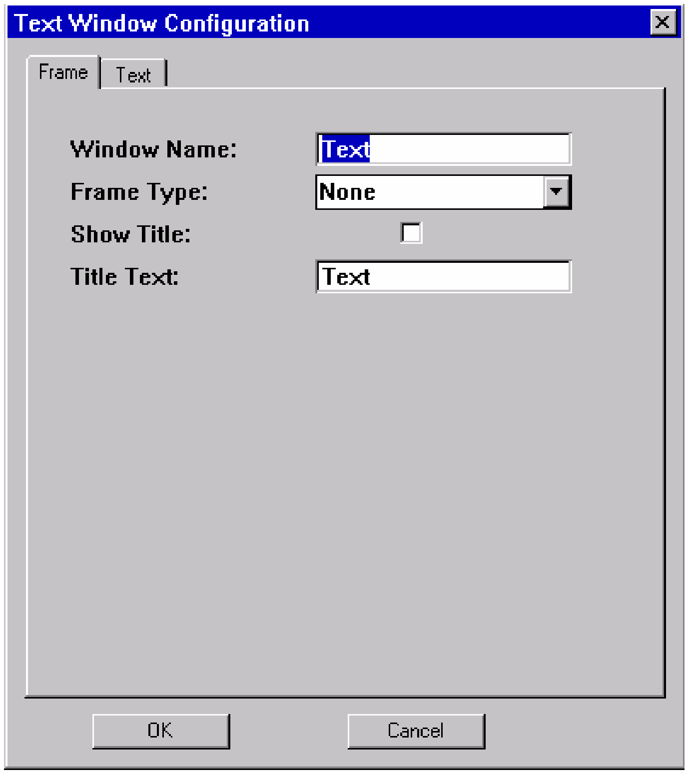


Figure 33: Text Window Configuration Dialog

As you have seen before, the Window Name is used to identify the Text Window in the SmartLink project, while the Window title or Title Text is a label displayed on the Window bar. They may be different if needed. Most often you will not display the window title.

Text Appearance
The Text tab contains a box to specify what text will appear on the display. It also contains options for controlling the font, color, size and wrapping of the text string.

Warning: Remember to size the window appropriately if you use larger fonts. Since the text string is centered on the window it may not be visible if the font size is large and the window is small.

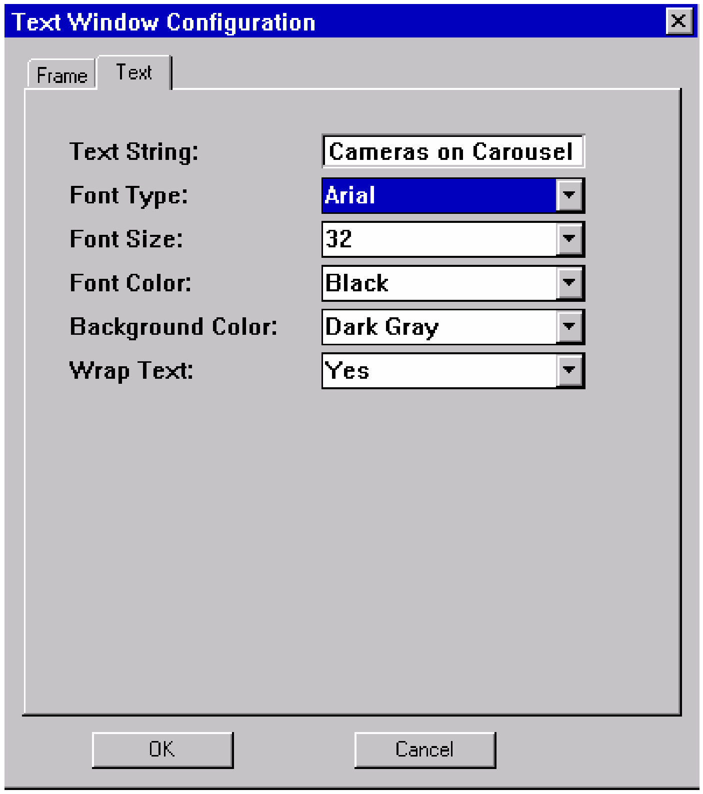
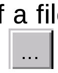


Figure 34: Text Tab of the Text Window Configuration Dialog

Adding Bitmaps to Your SmartLink Project

In the event that you want to add a bitmap or JPEG to the User Interface, such as a company logo or a sample golden image, then you may do so from the Add menu. This display component will allow the user to include any static Bitmap as part of the display application created through the SmartLink project. To add a bitmap or JPEG to the User Interface, go to the Add menu and select Bitmap. From here, a parameters box will appear, such as the one in Figure 35. You will configure the Frame tab as we have done on previous Frame tabs. The Bitmap tab only consists of a file name description that will tell the software where to look for the bitmap. You may click the  button and search for the file.

The bitmap or JPEG image will be copied from the PC hard drive to the project file on the SmartLink. The image will be saved on the SmartLink hardware unit and embedded in any configuration files that are stored on the PC.

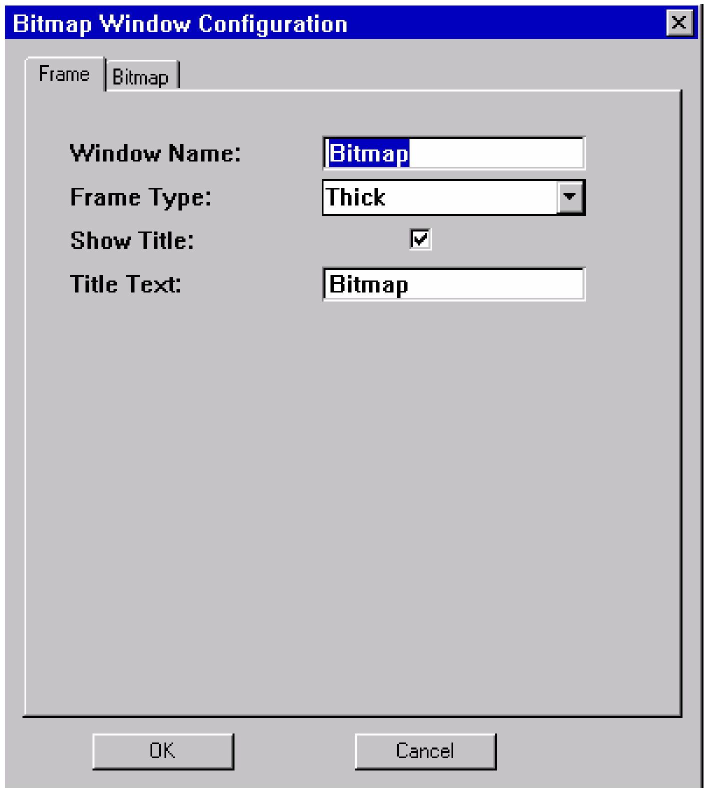


Figure 35: Frame Tab of the Bitmap Widow Configuration

Step 5: Testing Your User Interface

Once you have completely developed your SmartLink User Interface, you will want to test it to be sure that it is working properly. To do so, you may use the Test Mode option under the View menu. This will start the emulation of the project and images as well as any imported information should be sent to the SmartLink User Interface. This process occurs because the SmartLink Emulation connects to the SmartImage cameras via the PC's Ethernet card. It displays on the PC monitor the same information that would be displayed when the application is running on the SmartLink hardware unit. This allows the user to effectively test the project before downloading it to the unit. To exit from the test mode, hit the Esc key on your keyboard. This will return you to the setup mode.

When testing the project with TestMode, keep in mind the connections are being made from the PC's Ethernet card. During actual operation, they will be made from the SmartLink unit, which has different Ethernet settings.

Once you finish testing the User Interface, be sure to save the project file. This is done by selecting the Save As item under the File menu. Once the user interface is closed, the project file cannot be retrieved from the SmartLink hardware device.

Warning: If the project file is not backed up to the PC when you quit the SmartLink user interface, the project will have to be recreated from scratch. The current version of SmartLink does not support project retrieval from the hardware unit.

Once the project is complete and backed up to the PC, you can download it to a SmartLink hardware unit or run it from the PC using the user interface application.

Step 6: Download the Project to the SmartLink Hardware Unit

The first step to downloading the project is to connect to the SmartLink. Select the Remote System option under the Configure menu.

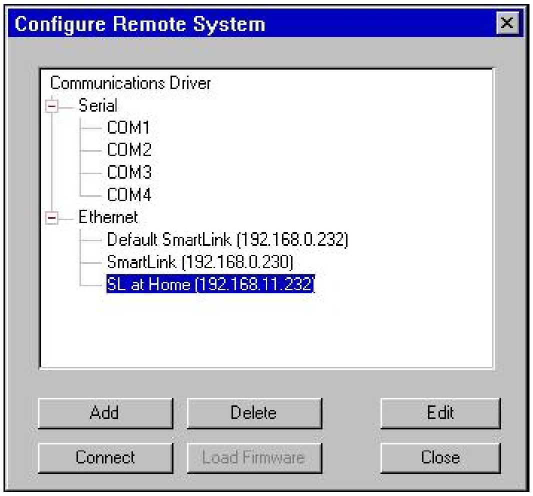


Figure 36: Configure Remote System

The project can be downloaded over the serial connection or through Ethernet. If you don't see the appropriate SmartLink module listed under the Ethernet section, use the Add button to insert another IP address.

After connecting to the SmartLink, hit the Download button on the Projects tab.

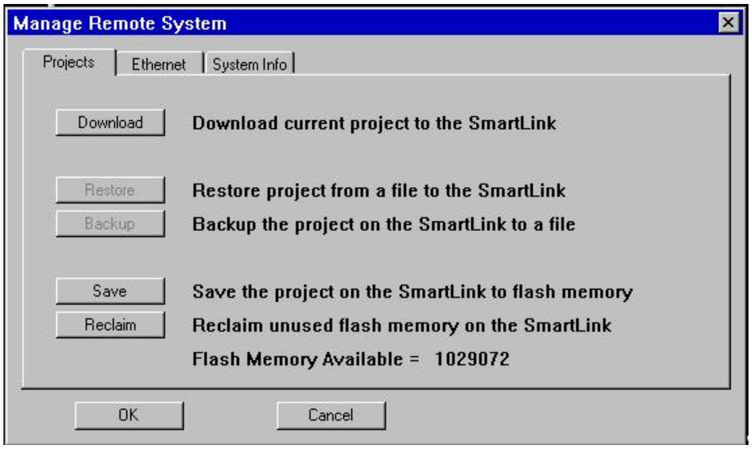


Figure 37: Download the Current Project

When the download is complete, the interface should inform you that the transfer is complete and as if you would like to save the project to flash memory. Flash memory is non-volatile memory onboard the hardware unit. The project that is saved to flash memory will be the one that runs when the unit is powered up. Normally you would select Yes on this dialog box. After a few seconds, the SmartLink will automatically start running the project file. If you have a monitor attached to the hardware device, you'll see the text and fixed bitmaps appear on the screen. As the SmartLink connects to the SmartImage cameras in the project, the image objects and tables will begin displaying images and data. In some cases the SmartImage camera may need to be triggered in order to see the live image and data appear.

Once the project has been downloaded to the SmartLink, you may close the Remote Dialog box and quit the SmartLink user interface program on the PC. The SmartLink hardware unit will run on its own without the intervention of a PC.

Running the Project on a PC

The SmartLink hardware module is designed to replace a PC on the factory floor for the purposes of monitoring inspections. If a PC is already on the machine and the objective is to monitor inspections without running Intellect, then the SmartLink user interface can be run on the PC without a hardware module present.

Set up a SmartLink project as outlined above and save the project file to the same directory on the PC as the SmartLink.exe file. The SmartLink project file can be started from the command line using the following syntax:

smartlink.exe /t project.slp

Where project.slp is the name of the project file you backed up to the PC. Starting SmartLink in this mode will put the project file directly into test mode. Hitting the Esc key will put SmartLink into design mode. To lock the application and only allow the user to quit completely, use the following switch instead:

smartlink.exe /T project.slp

The capability of running the SmartLink on a PC allows an end user to quickly put together a custom user interface without using Visual Basic or similar development programs.

Chapter 4 - Using Advanced SmartLink Objects

In the last chapter, we explored basic objects that made up a SmartLink project. The image, table, text, and bitmap are all designed to provide information in one direction: out of the SmartImage camera. In other words, no external intervention is required once the project is set up.

In this chapter we'll look at some powerful new additions to SmartLink that allow an operator to interact with the SmartLink to provide different views of a project and to input some information or a request directly to the SmartImage camera. To use the objects in this chapter, you need to be running the SmartLink application on a PC or using the SmartLink hardware unit with a compatible touch screen display or have a powered Mouse connected to the SmartLink.

Containers

When connecting to more than one or two SmartImage cameras, it's easy to run out of space on the work area, even at 1280 x 1024 pixels. Starting in SmartLink version 1.2.4, it's possible to create multiple tabbed views on the same project. Each view can have a unique configuration of images, tables, and other SmartLink objects. In the user interface, these views are called containers. Users switch between tabs by using a mouse cursor on the PC application and a touch screen for the SmartLink hardware installation.

In our sample User Interface, each tab represents a different area within the plant. Another example would be to reference the same cameras but in different languages. Figure 38 shows the first tab of the original sample User Interface in English, with the second and third tabs French and Spanish. So rather than having three different User Interfaces for three different languages, only one is needed for a particular situation. Each container can connect to up to sixteen different SmartImage cameras to return images and data.

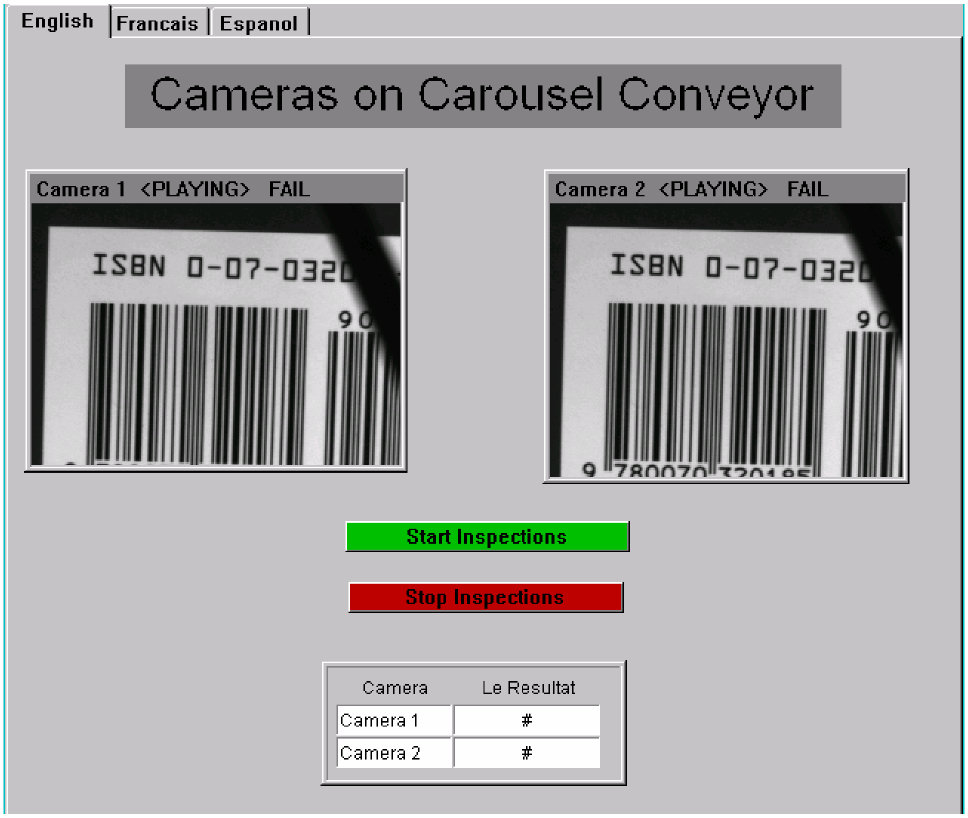


Figure 38:Container Example

To Add a Container, go to the Add menu and select Container. The configurations needed for the container are located in the Frame tab and the Text tab, both of which have been described in previous actions.

Warning: If you want to Add an object to a container, make sure that you have selected that container in the Configuration Panel in the left pane of the user interface. As of yet, you cannot drag that object to the correct area and you will therefore have to select the object and delete it from the Edit menu.

Buttons

SmartLink 1.2.4 introduced the idea of inputting information directly to a SmartImage camera from the screen. With a Button Control, you can enable actions, such as starting or stopping inspections, with the click of the button. In order to add a button to the User Interface, go to the Add menu and select Controls. From there, select Button. As in the past, a parameters box will appear upon selection so that you may configure the button to perform the tasks you need. Figure 39 illustrates the action of creating a button for the User Interface.

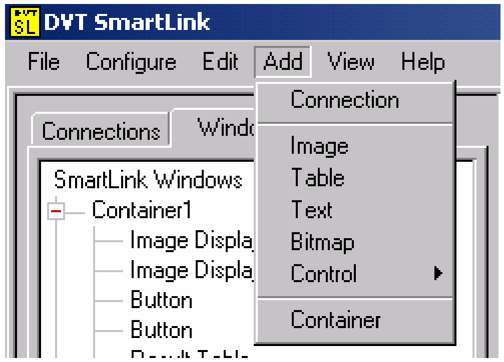
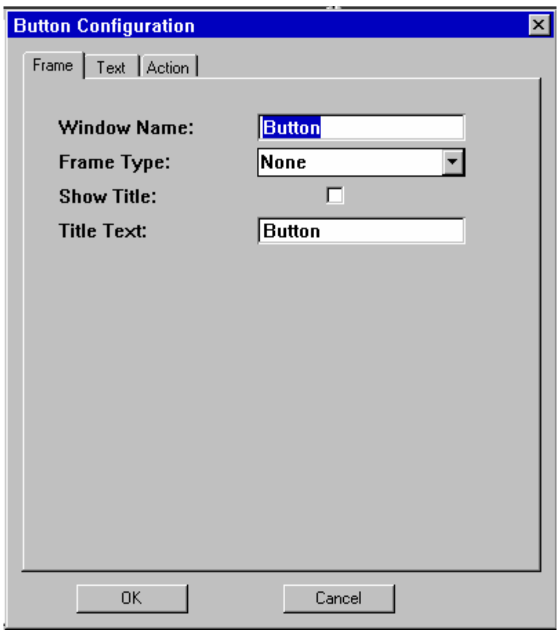


Figure 39: Creating a Button



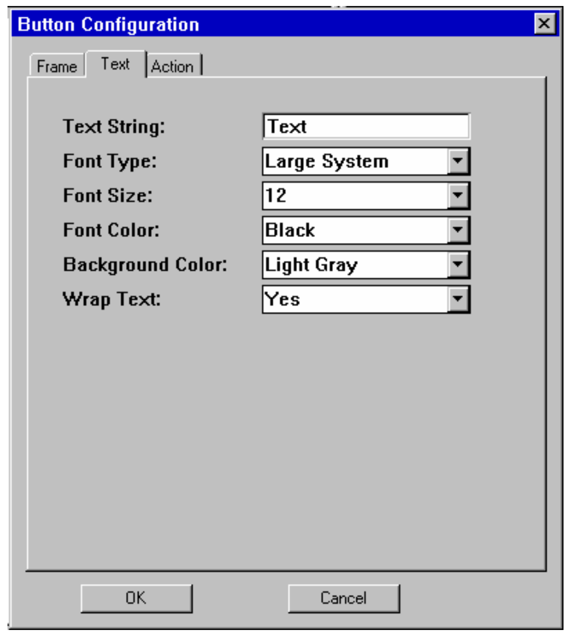


Figure 40: Frame and Text Tabs from the Button Configuration Parameters Box

The parameters set on the Frame and Text tabs will determine how the Button will appear on the layout screen and while SmartLink is in run mode. The Frame tab is configured as the other Frame tabs were configured. The Text tab specifies how the button will look. The Text String option allows you to give a name to the button, which will be displayed on the User Interface. For example, here you want to type “Start Inspections” or the equivalent so that the operator will know which button to press for a particular action to occur. The Font Type will enable you to manipulate the font of the text in which the Text String was written. Likewise, the Font Size, Font Color, and Background Color allow you to change the appearance of the button itself. If the text string associated with a particular button is a bit too long, you can Wrap Text so that it will wrap to the next line rather than try to write it on one line.

The Action tab enables you to associate an action with a button. Once you choose an action, the software will automatically bring up a connection for you to choose. This will enable you to apply the action you chose to a particular camera. Figure 41 illustrates a configured button.

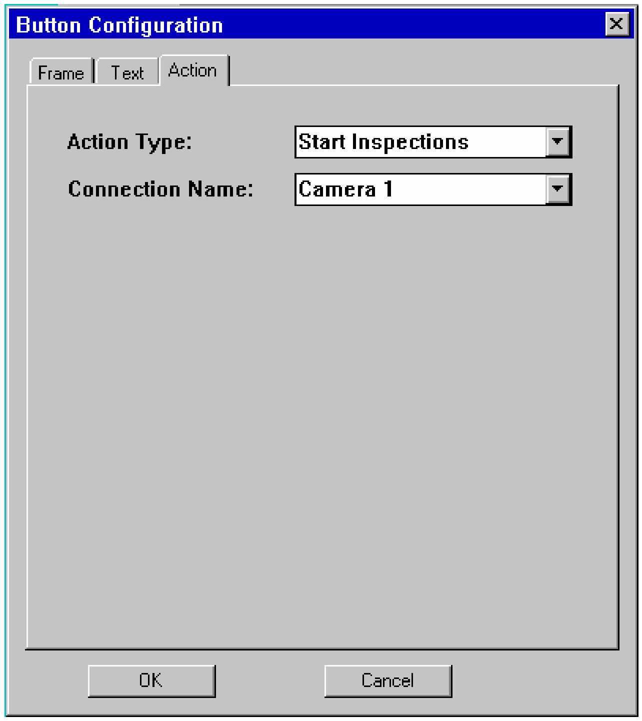


Figure 41: Action Tab of Button Configuration Parameters Box

There are several actions from which you can choose within the Action Type option. Each action will be discussed in detail.

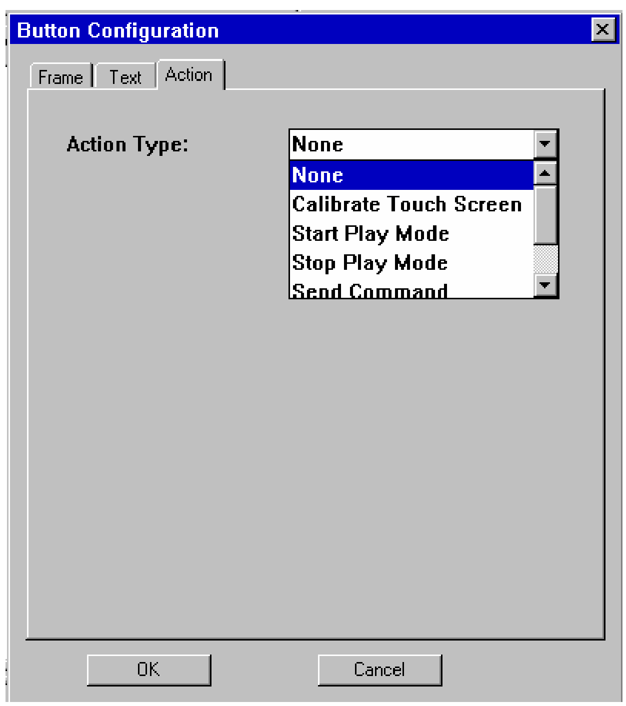


Figure 42: Some of the Available Actions

Calibrate Touch Screen

When a Button is configured to perform a 'Calibrate Touch Screen' Action it will start the calibration procedure for whatever touch screen is currently connected to the system (Figure 43). If a valid touch screen drive has not been configured for the serial port the calibration procedure will not start. Refer to Appendix B for further information on using Touch Screen monitors.

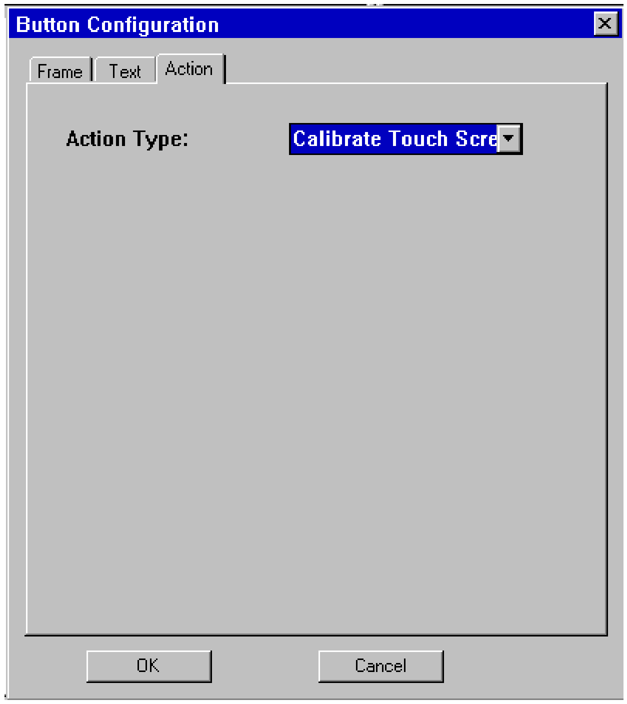


Figure 43: The Button Set to Calibrate Touch Screen Upon Selection

Start / Stop Play Mode

If either the Start Play Mode or the Stop Play Mode Action is selected the user will be asked to make the additional selection of a Connection Name (Figure 44). If configured to perform Start Play Mode Action, the Button will start the image and data playback for the Connection specified. If the Button is configured to Stop Play Mode, the image and data playback will stop for the specified Connection.

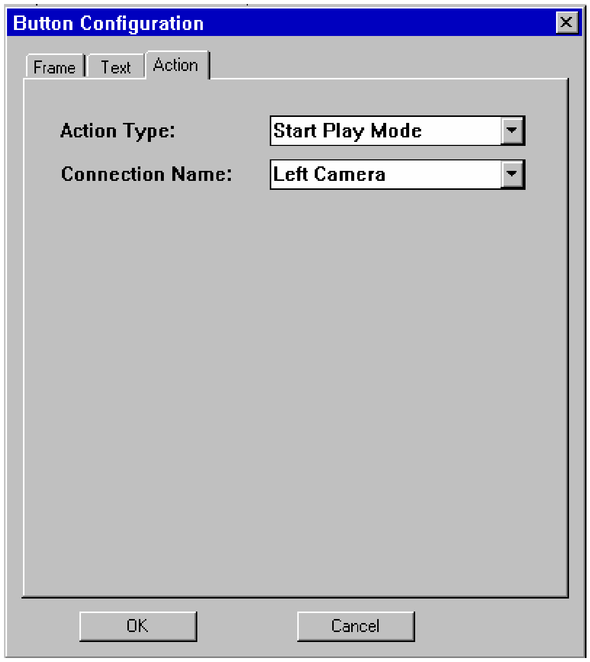


Figure 44: The Button Sets Play Mode for the Specified Connection

Start/Stop Inspections

If either the Start Inspections or the Stop Inspections Action is selected the user will be asked to make the additional selection of a 'Connection Name' (Figure 45). If configured to perform a Start Inspections Action, the Button will start the Inspections on the specified Connection as well as start playback of images and data from that Connection. If the Button is configured to 'Stop Inspections', Inspections will be stopped on the specified Connection as will the playback of images and data.

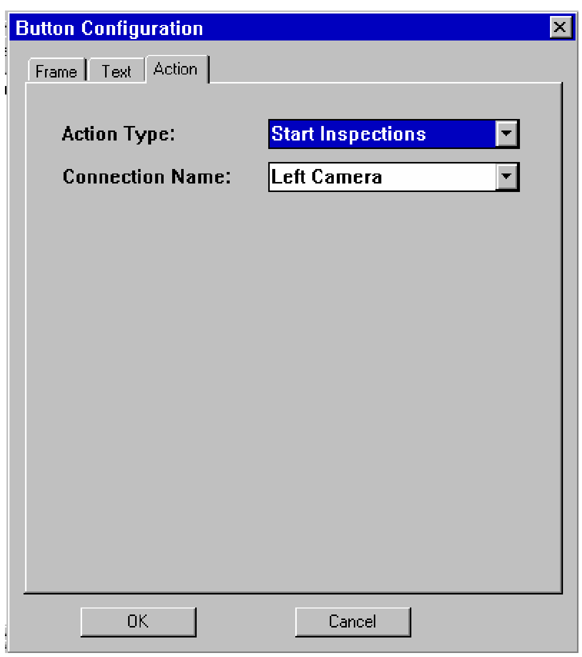


Figure 45: The Button Sets to Play Mode for the Specified Connection

Send Command

The Send Command Action will send the text command entered to the specified Connection. The only commands supported at this time are single line command (i.e. -- #YI, #PI2, etc). The command data type specifier (# for decimal in the example) should be entered prior to all commands just as would be done if the commands were entered over a HyperTerminal connection. For example, to configure a button to trigger the camera, enter #YI. This would allow an operator to step through inspections one by one. Using the #Rs command on the SmartLink and a script on the SmartImage camera allows the user to put values into registers and control most aspects of the inspection station without have to use Intellect. Refer to the Intellect Command List or Appendix A of the Intellect Users guide for available commands.

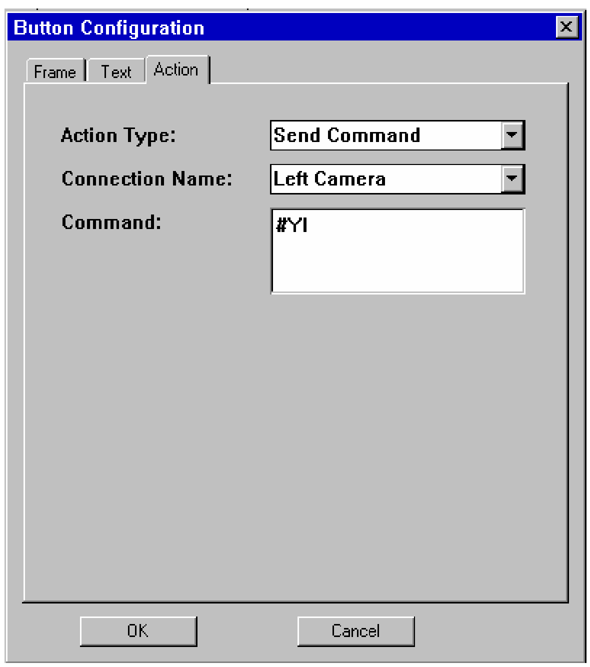


Figure 46: The Button is set to send the text command '#YI' to Camera 1

Chapter 5 - Configuring the Profibus Interface

Introduction

The SmartLink is a product that works in conjunction with DVT SmartImage cameras to expand their information sharing capabilities. It allows users to easily create a sophisticated monitoring system that will display inspection images and application specific information from several SmartImage cameras on a standard video monitor. In addition, certain versions of the SmartLink serve as communication bridges to industrial networks like Profibus or DeviceNet. The Profibus version of SmartLink is shown in Figure 47 below.

Figure 47: SmartLink with Profibus Connectivity

Data is transmitted cyclically from the SmartImage camera to the SmartLink unit using the Modbus TCP protocol. SmartLink then makes this data available to the Profibus network.

Profibus stands for Process Field Bus. Profibus is an international, vendor-independent, open Fieldbus standard. European standards EN 50170 and EN 50254 ensure vendor-independence and openness. The standard allows devices from multiple vendors to communicate without special interface adjustments. Profibus can be used for both complex communication tasks and high-speed time critical applications.

Profibus offers functionally graduated communication protocols (Communication Profiles): DP and FMS. Depending on the application, the transmission technologies (Physical Profiles) RS-485, IEC 1158-2 or fiber optics are available. Profibus Communication Profiles define how users transmit their data serially via the common transmission medium. DP is the most frequently used communication profile. It is optimized for speed, efficiency and low connection costs and is designed especially for communication between automation systems and distributed peripherals. DP is suitable as a replacement for conventional, parallel signal transmission with 24 volts in manufacturing automation as well as for analog signal transmission with 4 ... 20 mA or Hart in process automation. The Profibus SmartLink supports Profibus-DP with RS-485.

The media of this Fieldbus is a shielded copper cable composed of a twisted pair. The baud rate for the bus is between 9.6k baud to max. 12M baud. The Profibus network can consist of 126 nodes and the total amount of data for Profibus-DP is 244 Byte out and 244 Byte in per module.

Warning: Node 126 should not be used to exchange user data. This node is used for commissioning purposes

Development and administration of Profibus technology is handled by the User Organization known as the PTO in North America (http://www.profibus.com).

Installation and Configuration

Connectors

The SmartLink is connected to the Profibus network through the connector shown in Figure 48. The function of each pin is described in the table shown in Figure 49.

Figure 48: Profibus Connector (from left to right, pin 1 to 5 in bottom row and 6 to 9 in top row)

PIN	Name	Function
Housing	Shield	Connected to PE
1	Not Connected	-
2	Not Connected	-
3	B-Line	Positive RxD/TxD according to RS 485 specification
4	RTS	Request To Send *
5	GND Bus	Isolated GND from RS 485 side *
6	+5V Bus	Isolated +5V from RS 485 side *
7	Not Connected	-
8	A-Line	Negative RxD/TxD according to RS 485 specification
9	Not Connected	-

* +5V BUS and GND BUS are used for bus termination. Devices such as transceivers (RS485 to fiber optics) might require external power supply from these pins. RTS is used in some equipment to determine the direction of transmission. In normal applications only A-Line, B-Line and Shield are used.

Figure 49: Pin function of the Profibus connector (see Figure 48)

Node Address and Baud Rate

Each node in a Profibus network must have a distinct address. The node address in the SmartLink Profibus card is set using the rotary switches shown in Figure 50. Looking at the front of the SmartLink, the rightmost switch is used for the ten setting and the leftmost switch is used for setting the integers.

Example: Address = (Right Switch)*10 + (Left Switch)

The range for node address is between 1-99 and baud rate is 9.6 kbit/s-12Mbit/s.

The rotary switch is read only when power is connected to the SmartLink. Thus, you must cycle power to the SmartLink if the node address is changed.

Figure 50: Rotary Switches used to set node address.

The baud rate in a Profibus-DP network is selected during configuration of the master and only one baud rate is possible. The SmartLink has an auto baud rate detection function. A list of the supported baud rates is shown in Figure 51.

Baud rates supported by the SmartLink Profibus-DP
9.6 kbit/s
19.2 kbit/s
93.75 kbit/s
187.5 kbit/s
500 kbit/s
1.5 Mbit/s
3 Mbit/s
6 Mbit/s
12 Mbit/s

Figure 51: Supported Baud Rates

LED Indicators

The SmartLink is equipped with four LED's (shown in Figure 52) for debugging and monitoring purposes. LED #1 indicates the Off-Line Status, LED #2 the Fieldbus diagnostics, LED #3 the On-Line status and LED #4 is reserved for future use. The functions of the LED's are described in the table in Figure 53.

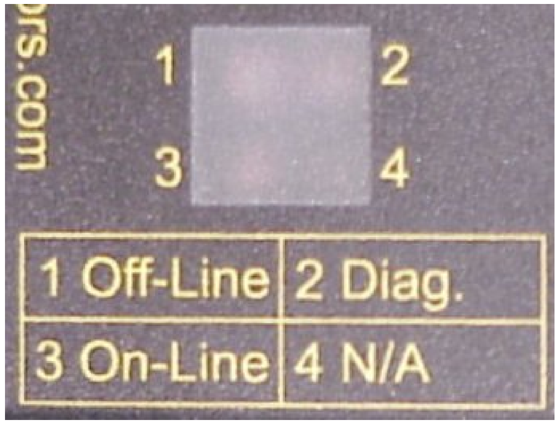


Figure 52: Profibus Status LED's

Name	Color	Function
Fielbus Diagnostics	RED	Indicates certain faults on the Fieldbus side. **Flashing Red 1 Hz** - Error in configuration: IN and /or OUT length set during initialization of the module is not equal to the length set during configuration of the network. **Flashing Red 2 Hz** - Error in User Parameter data: The length/contents of the User Parameter data set during configuration of the network. **Flashing Red 4Hz** - Error in initialization of the Profibus communication ASIC. **Turned Off** - No diagnostics present.
On-Line	Green	Indicates that the module is On-Line on the fieldbus. **Green** - Module is On-Line and data exchange is possible. **Turned Off** - Module is not On-Line
Off-Line	Red	Indicates that the module is Off-Line on the fieldbus. **Red -** Module is Off-Line and no data exchange is possible. **Turned Off** - Modules is Off-Line

Figure 53: Description of Profibus Status LED's Functions

Trunk Termination

The end nodes in a Profibus-DP network have to be terminated to avoid reflections on the bus line. A connector with a termination switch must be used if the SmartLink needs to be placed as an end node.

GSD file

Each type of device in a Profibus network is associated with a GSD file. This file contains all the necessary information about the device. This device is used by the network configuration application during initial set up and configuration.

Caution: It is important to make sure that the number of bytes to be transferred is the same in the SmartLink application (see Operation) and in the Profibus Master setup software

The latest version of the SmartLink GSD file can be downloaded from http://www.dvtsensors.com/support/DVTDownloads.htm.

DVT / Profibus Data Transfer Operation

The operation of the SmartLink Profibus interface is very simple. However, it is very important to understand how the data transfer is accomplished. Figure 54 describes this process.

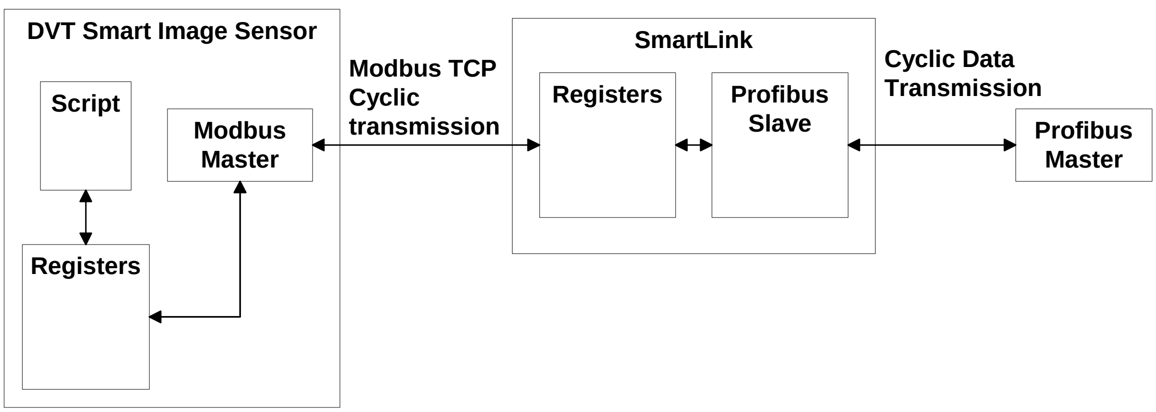


Figure 54: Data Transfer Process

A script running in the SmartImage camera writes data to registers in the sensor memory. This can happen at the same rate that the inspections are running (Script SoftSensor) or asynchronously (through a Background Script). Data from the SmartImage camera is transferred to the SmartLink via Modbus protocol. This is a cyclic process where the camera works as a master writing to and reading from the SmartLink. Finally, data in the SmartLink is exchanged with the Profibus network via cyclic data exchange.

Setting up the SmartLink to Exchange Data with the Profibus Network

The following sections describe the set up procedure for exchanging data with a Profibus Master using the SmartLink.

Writing Data to Registers

Script register management functions are used to write and read data in memory registers in the SmartImage camera. Examples of these functions are:

```
RegisterWriteByte( , );
RegisterReadByte( );
RegisterWriteShort( , );
```

```
RegisterReadShort( );
```

Care should be taken when using these functions. DVT registers are 1 byte long, but some of these functions use more than one byte depending on the format used to write/read the data (i.e. Byte, Short, etc).

Please refer to the Script Reference Manual for detail information on how to use these and any other necessary script commands.

In Intellect sensor, product and system data can be written to registers using Tags. This allows the user to setup an automatic transfer of results or system/product parameters to memory for use by other processes, such as Modbus communications. Setup of Sensor tags will be discussed below.

Setting up a Modbus Transfer

An example that is provided later in this document explains the steps to set the Modbus Master in the SmartImage camera in order to read/write to registers in the SmartLink. Data sent from the camera to the Profibus network is written to DVT registers 1000 through 1511 in SmartLink. Data received from the Profibus network has to be read by the camera from DVT registers 2000 through 2511 in the SmartLink. This is illustrated in Figure 55.

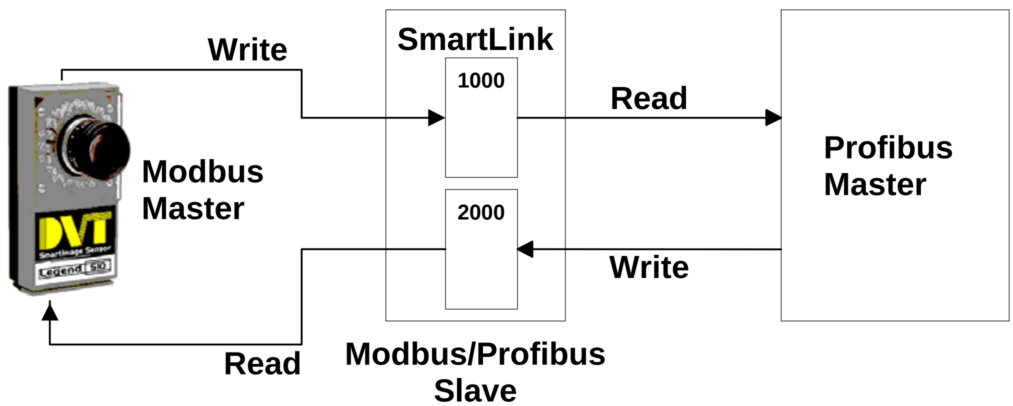


Figure 55: SmartLink Registers for Data Exchange with the Profibus Master

Exchanging Data with Profibus

Your Profibus network can be configured using any software utility designed for this purpose such as OpenControl for Profibus. As far as the Profibus network is concerned, the SmartLink is another node in the network.

SmartLink Project Configuration Specific to Profibus

One parameter needs to be set as part of the SmartLink project file. The Fieldbus tab under the Configure Local System dialog box is shown in Figure 56. This dialog box is found under the Project Parameters selection from the Configure menu in the SmartLink application. The Cyclic Data Input Length parameters define the number of bytes (not Modbus register which are two bytes long) that will be exchanged with the Profibus network. These numbers should match your network configuration.

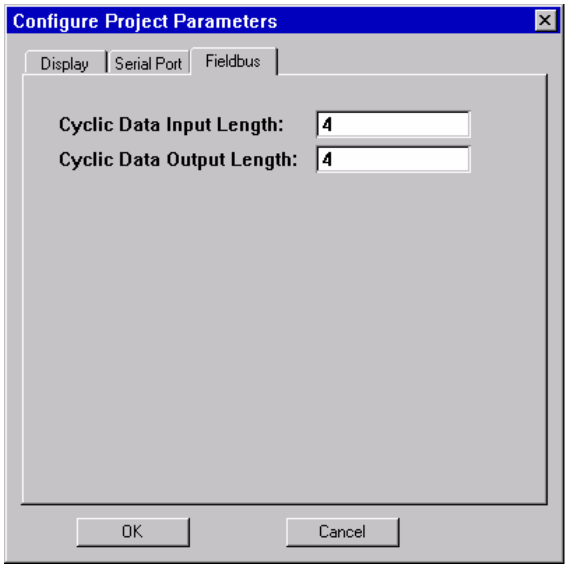


Figure 56: Configure local system dialogue box

The data is written to the SmartLink memory using the BigEndian convention. This should be taken into account when interpreting the data read from the SmartLink by other devices in the Profibus network.

Example

The following example shows the set up procedure for a Profibus application. It assumes that the SmartImage camera product file has a Count in Area sensor with Object Locate as the selected task. We will transfer the object count from a source camera's register to a SmartLink via Modbus, and consequently to the Profibus master.

Name	Method	Output
Image Acquisition		
ObjLocate	Counting: Area: Object ...	# Objects = 2, Match Score = 100...
Inspection Results		
Communications Ex...		

Figure 57: Object Locate sensor in the Intellect Result Table

In this example, the camera acts as the Modbus Master and the SmartLink as the Slave. The number of objects found must be sent to the Profibus Master and one byte of information will be received from it, thus the SmartLink acts as the middle man as the slave to both processes.

These following steps describe the procedure to set up this application:

Register Management using a Sensor Tag

Because Modbus data is transferred between registers on Master and Slave devices, a Modbus register tag is needed to assign the count output of the Object Locate sensor to one of the Modbus registers on the Master device, in this case the camera. This register will be used as the source data to be transferred to the SmartLink.

In FrameWork you needed to save this data to memory registers through script, as shown in the following script example. The first command in the simple Script shown below writes the ObjectCount data (to be sent to the Profibus Master) to Register 100 in the SmartImage camera memory. The second command reads the value in register 200 (data from the Profibus Master will be placed here), and assigns this value to the String output of the Script sensor. This value is then shown in the result table (see Figure 57).

```
RegisterWriteShort(100, ObjLocate.ObjectCount);
Script.String = "" + RegisterReadByte(200);
```

In FrameWork, Modbus and DVT memory is mixed together, so there is only one memory space to write to and read from in scripts and communications. In Intellect, because Modbus and DVT memory is separated into 8bit and 16bit memory, you must change this script so that the sensor data is written to and read from Modbus' 16 bit memory space. However, it is not necessary to use scripts at all to accomplish this in Intellect. Instead we can save sensor data to different areas of memory using tags.

Create a Sensor Tag by opening the Communications Explorer and browsing to the folder Modbus TCP>>Modbus Tags, and right clicking in the right pane of the window. Select Add>>Sensor Tag from the menu. Highlight the new tag and edit its properties as shown below.

General	
Type	Sensor Tag
Name	Object Count
Register Address	
Start	16
End	16
Length (Registers)	1
Error	
Register Data	
Data Type	Short
Format	Decimal
Size (Bytes)	2
Value	2
Source Expression	ObjLocate.ObjectCount

Figure 58: Modbus sensor tag properties

The Start register is the beginning register of the data value. The end and length are calculated and displayed automatically based on the Data Type entered below. Remember that Modbus memory uses 16bit registers instead of 8bit like DVT registers, so a 2 byte number like a data type short occupies only one register. Finally click the white area to the right of Source Expression to open the Sensor Parameter Browser. The Parameter Browser allows you to assign a sensor's output data to the Sensor Tag. Browse to the ObjLocate Sensor, select ObjectCount and press OK. Now this Sensor Tag, here named Object Count, automatically writes the low two bytes of ObjLocate.ObjectCount to the Modbus memory register 16 every time an inspection occurs. Note that ObjLocate.Objectcount returns an Integer which is 4 bytes, so we are truncating the two high bytes. Also note that no scripting has been used.

Modbus Master Setup

The next step is to set up the Modbus data exchange. This will allow the SmartImage camera to cyclically exchange data with the SmartLink and therefore with the Profibus Master. The SmartImage camera will be the Modbus Master controlling the data exchange with the Slave (the SmartLink). Highlight Modbus Masters in the Communications explorer and double click in the right window pane to create a new Modbus Master. Its properties are displayed in the properties window. The only value that needs to be changed for this example is the IP address. Set this to the address of the SmartLink. This is the IP address of the slave for all transfers made by the current Modbus Master. To initiate Modbus communications with another slave device, simply create another Modbus Master with the new slave's IP address.

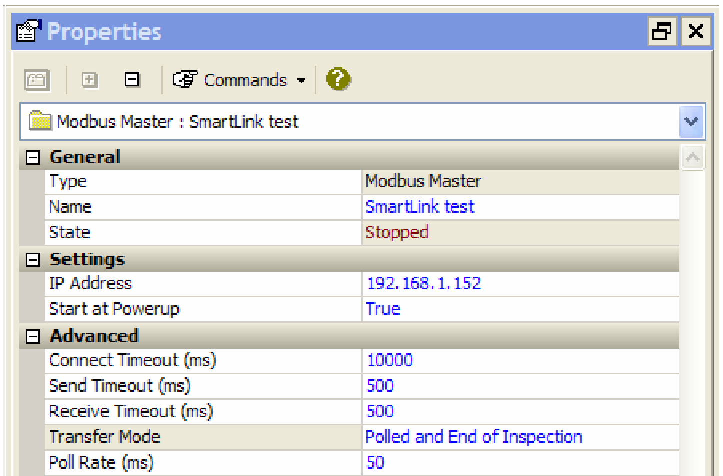


Figure 59: Modbus Master properties

Connection Time Out: This is a time in ms that releases a connection attempt to a Modbus Slave. It's useful when a connection cannot be established with the slave. Decrease this value if you are connecting to more than two slaves. Also, decrease this value if you decrease the polling rate.

Send/Receive Time Out: This is a time in ms after which the Master stops attempting to send or receive data over a good connection.

Transfer Mode: Determines whether data is transferred at a polling interval, when data changes, at the end of an inspection, or at a combination of those events.

Poll Rate: This is the rate in ms at which the cyclic data exchange between the SmartLink and the SmartImage camera takes place. This means that every 50 ms this Modbus Master (camera) will initiate all of its reads and writes to and from the Modbus Slave (Smartlink).

Modbus Master Transfer Setup

Next open the folder for the newly created Modbus Master and double click in the right window pane to create new read and write transfers. Again the properties for each created transfer are displayed in the properties window. Set the properties to the values shown in figure 60.

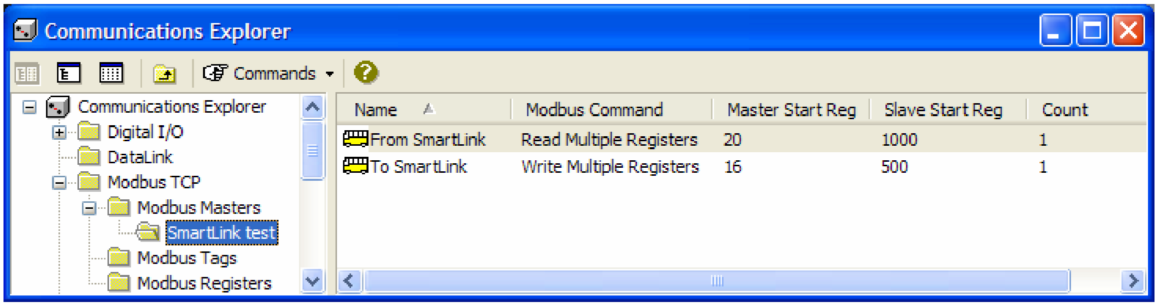


Figure 60: Modbus Master transfer configuration

The following parameters have to be set in the properties window:

Name: A name to identify this transfer.

Command: This pull down box has two options: Write Multiple Registers or Read Multiple Registers. Select Write Multiple Registers

Slave Register Start: That is the number of the register in SmartLink where the data will be written. As mentioned earlier, data that is going to the Profibus Master should be stored in DVT registers 1000 through 1511 in the SmartLink. The number in this box is 500 because DVT registers are 1 byte long, while Modbus Registers are 2 bytes long. The data source on the Modbus Master, the camera, is the 2 byte Modbus register space. The SmartLink however uses 1 byte DVT registers, so to write to the 1000th byte in the SmartLink's memory we must write to the equivalent Modbus register, register 500.

The Modbus protocol uses 16-bit registers to transfer data. This means that each Modbus register corresponds to two consecutive DVT registers as shown in **Error! Reference source not found.**. Modbus registers are BigEndian (high byte first).

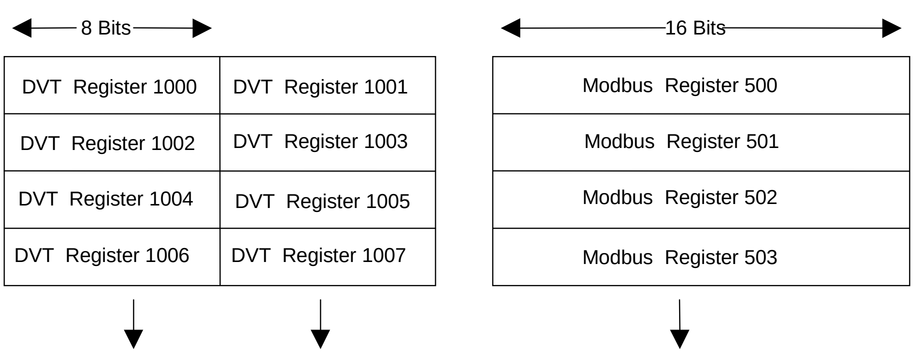


Figure 61: DVT and Modbus registers

Master Register Start: That is the number of the register in the SmartImage camera where the data was written by the Script SoftSensor. In this case, data was written to Modbus register 16.

Number of Registers: Data was written as type Short by the sensor tag, which is two bytes long. Therefore, 1 Modbus register is enough to transfer all the data. Care should be taken when reading/writing more than one byte with the Profibus Master from the SmartLink because currently the SmartLink uses the BigEndian convention to write and read data to memory. Using a Profibus master with LittleEndian convention to read data in formats that use more than one byte will result in the master reading garbage data. Manipulating the bytes with the Script Tools can change the byte order in the camera and will solve this problem.

The next step is to configure a new Modbus transfer for reading data coming from the Profibus Master. The procedure is similar, except this time a Read Multiple Registers command will be created and the registers will be different. The data from the Profibus Master will be available in DVT registers 2000 to 2511; therefore, the Slave Register Start in this case is 1000. Once the data is read to Modbus memory in the camera, in this case register 20 (see Figure: 62), it can be accessed from script or by a Monitor Tag similar to the Sensor Tag created earlier. Your read settings should be set as shown below.

Type	Modbus Transfer
Name	From SmartLink
⊟ **Settings**	
Modbus Command	Read Multiple Registers
Master Start Reg	20
Slave Start Reg	1000
Count	1

Figure 62: Modbus Read Multiple Registers

SmartLink Profibus Setup

Finally, one parameter needs to be set as part of the SmartLink project file. The dialog box shown in Figure 63 is found under the Project Parameters selection from the Configure drop down menu in the SmartLink application. The Cyclic Data Input Length parameters define the number of bytes that will be exchanged with the Profibus network. In this case, we are reading and writing 2 bytes. These numbers should match your network configuration.

Notice that Modbus is set to write/read two DVT registers (two bytes) and the SmartLink is only set to exchange one byte. This is done because the minimum amount of data transmitted with Modbus is one Modbus register, which is equivalent to two DVT registers. If our Sensor Tag was set to only write one byte of data, we would only need to make one byte available to the Profibus master and disregard the second byte transferred by Modbus.

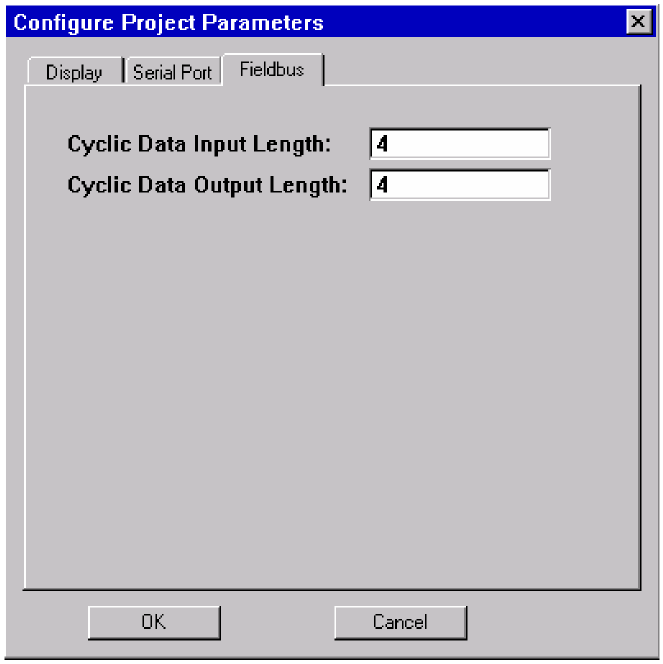


Figure 63: Cyclic Data Transfer Setup for Profibus

Troubleshooting the Application

When data is not being transferred from the Profibus network to the SmartImage camera (or vice versa), it's helpful to figure out what part of the communication is breaking down. It might be the Modbus transfer between the SmartImage camera and the SmartLink or it could be the card that's transferring the information between the Modbus registers and Profibus registers.

DVT recommends connecting to the SmartLink directly with telnet over Ethernet or HyperTerminal over the serial connection. If you're using a touch screen, you will have to use the Ethernet option since the serial port is tied up with the display. To connect over the serial line, start up HyperTerminal using the instructions on pages 16-17 of this manual. Instead of cycling power on the SmartLink, simply hit the Enter key and observe the question mark prompt.

To connect to the SmartLink via Ethernet, start your telnet program by going to the Start menu, selecting Run, and typing Telnet. Hitting the Okay button should bring up a telnet screen. On Windows 2000 and XP, you'll be looking at a command prompt called Microsoft Telnet. Type:

open <IP Address of SmartLink> 5100

and hit enter. For example: open 192.168.0.232 5100

You should be greeted with System Terminal Driver and a question mark prompt.

In either case, once you have the question mark prompt, you can enter a command to read the register out of the SmartLink. To verify the Modbus master is working, you may want to check the contents of register 1000 on the SmartLink. At the command prompt, type:

#Rq1000I

You should get a $ followed by the value in the register. On the next line should be a %0, meaning that the command was successfully executed. Note that the 'I' returns an integer which is 4 DVT registers or 2 Modbus registers. You'll have to figure out if the correct value is getting across. Checking that the SmartImage camera is reading the values correctly out of the SmartLink is done in a similar way. Since the camera is reading starting at SmartLink register 2000, set a value in the following way:

#Rs2000I1234

Setup a monitor tag of the appropriate data type to read Modbus register 20 to test whether the data written to SmartLink register 2000 was transferred back to the camera.

Chapter 6 - Configuring the DeviceNet Interface

Introduction

DeviceNet is an open, low-cost communications network based on a broadcast-oriented protocol, the reliable CAN (Control Area Network, an ISO standard for automotive networking). This technology is used to interconnect industrial devices such as limit switches, photoelectric sensors, process sensors, valves, motor starters, etc. via a single link. This results in significant savings due to the elimination of expensive and time consuming wiring. In addition, it provides reliable interchangeability of components from multiple vendors. The direct connectivity results in improved communication between devices as well as important device-level diagnostics not easily accessible or available through hardwired I/O interfaces.

Figure 64: SmartLink with DeviceNet Connectivity

The media of this Fieldbus is a shielded copper cable composed of one twisted pair for data transmission and two cables for the external power supply. The baud rate can be set to 125k, 250k or 500kbit/s.

DeviceNet is managed by ODVA, Open DeviceNet Vendor Association (ODVA). More information on DeviceNet can be found in http://www.odva.org.

Installation and Configuration

Connectors

The SmartLink is connected to the DeviceNet network through the connector shown in Figure 65. The function of each pin is described in the table shown in Figure 66.

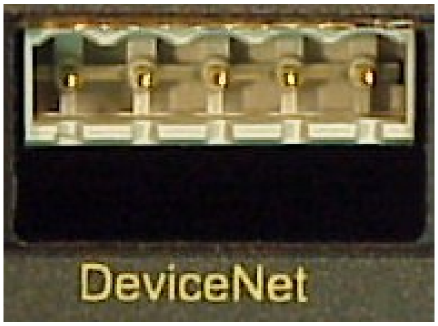


Figure 65: DeviceNet Connector (pin 1 to 5 from left to right)

Pluggable Control	Screw Terminal	Description
1	1	V+
2	2	CAN_H
3	3	SHIELD
4	4	CAN_L
5	5	V-

Figure 66: Pin function of the DeviceNet connector (see Figure 65)

MAC Address and Baud Rate

The MacID in a DeviceNet network is the node address. Each device in the network must have a different one. The MacID in the SmartLink DeviceNet card is set using the DipSwitch shown in Figure 67. This DipSwitch is also used to set the baud rate as described in Figure 69.

The range for MacID is between 0-63 and baud rate is between 0-2 (0=125kb, 1=250kb and 2=500kb).

The DipSwitch is read only when power is connected to the SmartLink. Thus, you must cycle power to the SmartLink if the MacID or baud rate is changed.

Figure 67: DipSwitch used to set MacID and baud rate

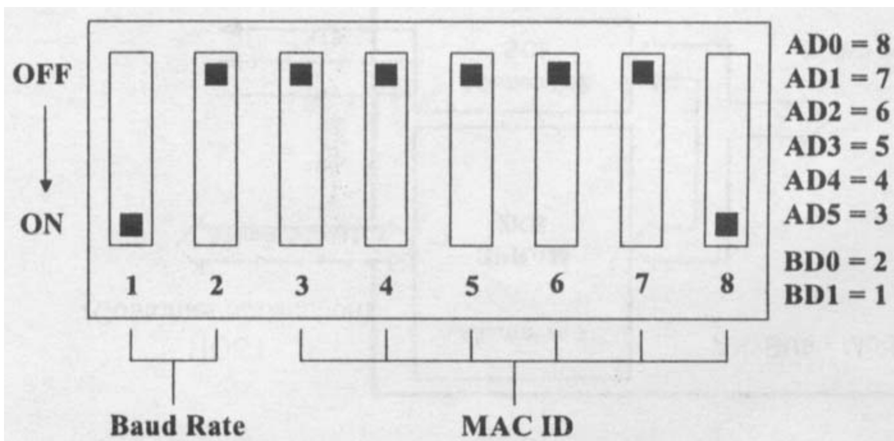


Figure 68: DipSwitch Assignment

ADDRESS	DIP 3-8
0	000000
1	000001
2	000010
3	000011
...	...
62	111110
63	111111

BAUD RATE	DIP 1-2
125K	00
250K	01
500K	10
Reserved	11

Figure 69: DipSwitch Settings for Address and Baud Rate

LED Indicators

The SmartLink is equipped with four LED's (shown in Figure 70) for debugging and monitoring purposes. LED #1 indicates the Module Status, LED #3 the network status and LED's #2 and #4 are reserved for future use. The functions of the LED's are described in the table in Figure 71.

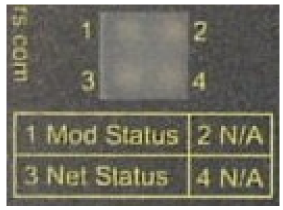


Figure 70: DeviceNet status LED's

LED'S	Description
Module_Status, steady off	No Power
Module_Status, steady red	Unrecoverable Fault
Module_Status, steady green	Device Operational
Module_Status, flashing red	Minor fault
NetWork_Status, steady off	Not Powered/Not online
Net_Work_Status, steady green	Link OK on line, Connected
NetWork_Status, steady red	Critical Link failure
NetWork_Status, flashing green	On line not connected
NetWork_Status, flashing red	Connection Time Out

Figure 71: Description of DeviceNet Status LED's Functions

Trunk Termination

A terminating resistor is needed if the SmartLink is the last node along the DeviceNet trunk. Use a 121 Ohm resistor between pins 2 and 4 in the DeviceNet connector.

EDS file

Each type of device in a DeviceNet network is associated with an EDS file. This file contains all the necessary information about the device. This device is used by the network configuration application during initial set up and configuration.

The latest version of the SmartLink EDS file can be downloaded from http://www.dvtsensors.com/support/DVTDownloads.htm.

DVT/DeviceNet Data Transfer Operation

The operation of the SmartLink DeviceNet interface is very simple. However, it is very important to understand how the data transfer is accomplished. Figure 72 describes this process.

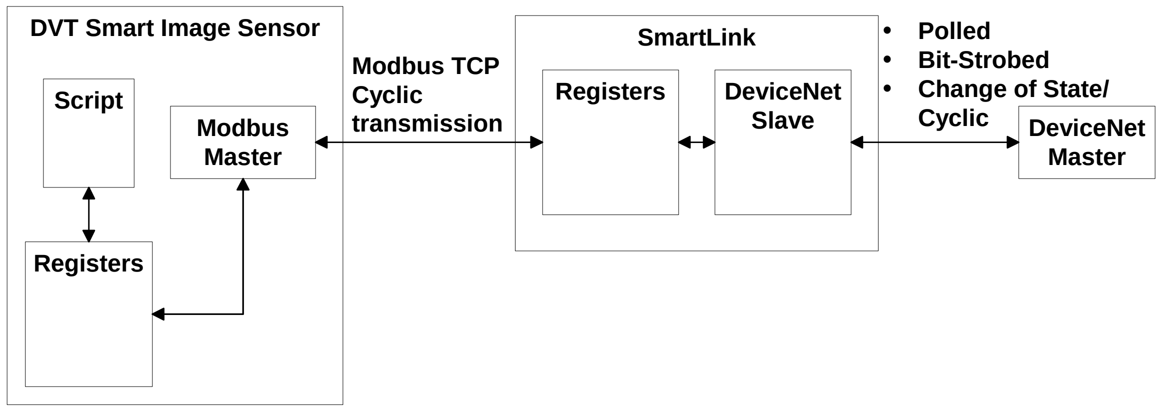


Figure 72: Data Transfer Process

A script running in the SmartImage camera writes data to registers in the sensor memory. This can happen at the same rate that the inspections are running (Script SoftSensor) or asynchronously (Background Script). Data from the SmartImage camera is transferred to the SmartLink via Modbus protocol. This is a cyclic process where the camera works as a master writing to and reading from the SmartLink. Finally, data in the SmartLink is exchanged with the DeviceNet network via any of the three connections supported: Polled, Bit-strobed or Change of State/Cyclic.

Setting up the SmartLink to exchange Data with the DeviceNet Network

The following sections describe the set up procedure for exchanging data with a DeviceNet master using the SmartLink:

Writing Data to Registers

Script register management functions are used to write and read data in memory registers in the SmartImage camera. Examples of these functions are:

```
RegisterWriteByte( , );
RegisterReadByte( );
RegisterWriteShort( , );
RegisterReadShort( );
```

Care should be taken when using these functions. DVT registers are 1 byte long, but some of these functions use more than one byte depending on the format used to write/read the data (i.e. Byte, Short, etc).

Please refer to the Script Reference Manual for detail information on how to use these and any other necessary script commands.

In Intellect sensor, product and system data can be written to registers using Tags. This allows the user to setup an automatic transfer of results or system/product parameters to memory for use by other processes, such as Modbus

Setting up a Modbus Transfer

An example that is provided later in this document explains the steps to set the Modbus Master in the SmartImage camera in order to read/write to registers in the SmartLink. Data sent from the camera to the DeviceNet network is written to DVT registers 1000 through 1511 in the

SmartLink. Data received from the DeviceNet network has to be read by the camera from DVT registers 2000 through 2511 in the SmartLink. This is illustrated in Figure 73.

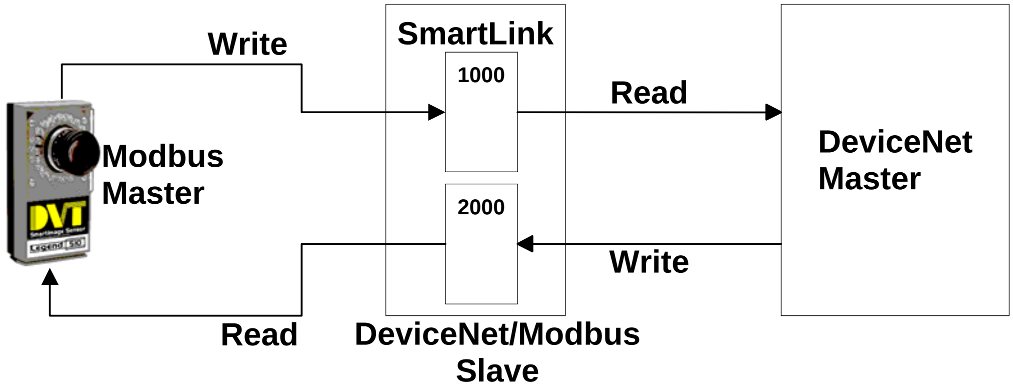


Figure 73: SmartLink Registers for Data Exchange with the DeviceNet Master

Exchanging Data with DeviceNet

Your DeviceNet network can be configured using any software utility designed for this purpose such as RSNetworx for DeviceNet from Allen-Bradley. As far as the DeviceNet network is concerned, the SmartLink is another node in the network.

SmartLink Project Configuration Specific to DeviceNet

One parameter needs to be set as part of the SmartLink project file. The Fieldbus tab under the Configure Local System dialog box is shown in Figure 74. This dialog box is found under the Project Parameters selection from the Configure menu in the SmartLink application. The Cyclic Data Input Length parameters define the number of bytes (not Modbus register which are two bytes long) that will be exchanged with the DeviceNet network. These numbers should match your network configuration.

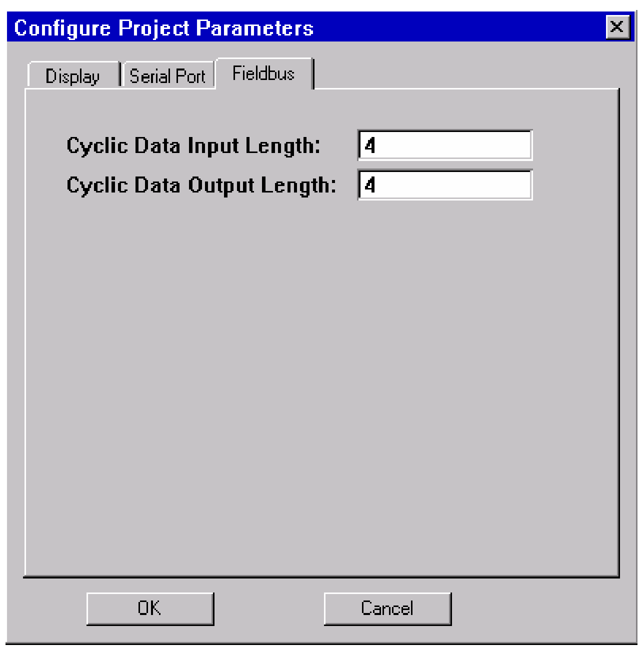


Figure 74: Configure Local System Dialog Box

The data is written to the SmartLink memory using the BigEndian convention. This should be taken into account when interpreting the data read from the SmartLink by other devices in the DeviceNet network.

Example

The following example shows the set up procedure for a DeviceNet application. It assumes that the SmartImage camera product file has a Count in Area sensor with Object Locate as the selected task. We will transfer the object count from a source camera's register to a SmartLink via Modbus, and consequently to the DeviceNet master.

Name	Method	Output
Image Acquisition		
ObjLocate	Counting: Area: Object ...	# Objects = 2, Match Score = 100...
Inspection Results		
Communications Ex...		

Figure 75: Object Locate sensor in the Intellect Result Table

In this example, the camera acts as the Modbus Master and the SmartLink as the Slave. The number of objects found must be sent to the DeviceNet Master and one byte of information will be received from it, thus the SmartLink acts as the middle man as the slave to both processes.

These following steps describe the procedure to set up this application:

Register Management using a Sensor Tag

Because Modbus data is transferred between registers on Master and Slave devices, a Modbus register tag is needed to assign the count output of the Object Locate sensor to one of the Modbus registers on the Master device, in this case the camera. This register will be used as the source data to be transferred to the SmartLink.

In FrameWork you needed to save this data to memory registers through script, as shown in the following script example. The first command in the simple Script shown below writes the ObjectCount data (to be sent to the DeviceNet Master) to Register 100 in the SmartImage camera memory. The second command reads the value in register 200 (data from the DeviceNet Master will be placed here), and assigns this value to the String output of the Script sensor. This value is then shown in the result table (see Figure 57).

```
RegisterWriteShort(100, ObjLocate.ObjectCount);
Script.String = "" + RegisterReadByte(200);
```

In FrameWork, Modbus and DVT memory is mixed together, so there is only one memory space to write to and read from in scripts and communications. In Intellect, because Modbus and DVT memory is separated into 8bit and 16bit memory, you must change this script so that the sensor data is written to and read from Modbus' 16 bit memory space. However, it is not necessary to use scripts at all to accomplish this in Intellect. Instead we can save sensor data to different areas of memory using tags.

Create a Sensor Tag by opening the Communications Explorer and browsing to the folder Modbus TCP>>Modbus Tags, and right clicking in the right pane of the window. Select Add>>Sensor Tag from the menu. Highlight the new tag and edit its properties as shown below.

General	
Type	Sensor Tag
Name	Object Count
Register Address	
Start	16
End	16
Length (Registers)	1
Error	
Register Data	
Data Type	Short
Format	Decimal
Size (Bytes)	2
Value	2
Source Expression	ObjLocate.ObjectCount

Figure 76: Modbus sensor tag properties

The Start register is the beginning register of the data value. The end and length are calculated and displayed automatically based on the Data Type entered below. Remember that Modbus memory uses 16bit registers instead of 8bit like DVT registers, so a 2 byte number like a data type short occupies only one register. Finally click the white area to the right of Source Expression to open the Sensor Parameter Browser. The Parameter Browser allows you to assign a sensor's output data to the Sensor Tag. Browse to the ObjLocate Sensor, select ObjectCount and press OK. Now this Sensor Tag, here named Object Count, automatically writes the low two bytes of ObjLocate.ObjectCount to the Modbus memory register 16 every time an inspection occurs. Note that ObjLocate.Objectcount returns an Integer which is 4 bytes, so we are truncating the two high bytes. Also note that no scripting has been used.

Modbus Master Setup

The next step is to set up the Modbus data exchange. This will allow the SmartImage camera to cyclically exchange data with the SmartLink and therefore with the DeviceNet Master. The SmartImage camera will be the Modbus Master controlling the data exchange with the Slave (the SmartLink). Highlight Modbus Masters in the Communications explorer and double click in the right window pane to create a new Modbus Master. Its properties are displayed in the properties window. The only value that needs to be changed for this example is the IP address. Set this to the address of the SmartLink. This is the IP address of the slave for all transfers made by the current Modbus Master. To initiate Modbus communications with another slave device, simply create another Modbus Master with the new slave's IP address.

Properties

Commands

Modbus Master : SmartLink test

General	
Type	Modbus Master
Name	SmartLink test
State	Stopped
Settings	
IP Address	192.168.1.152
Start at Powerup	True
Advanced	
Connect Timeout (ms)	10000
Send Timeout (ms)	500
Receive Timeout (ms)	500
Transfer Mode	Polled and End of Inspection
Poll Rate (ms)	50

Figure 77: Modbus Master properties

Connection Time Out: This is a time in ms that releases a connection attempt to a Modbus Slave. It's useful when a connection cannot be established with the slave. Decrease this value if you are connecting to more than two slaves. Also, decrease this value if you decrease the polling rate.

Send/Receive Time Out: This is a time in ms after which the Master stops attempting to send or receive data over a good connection.

Transfer Mode: Determines whether data is transferred at a polling interval, when data changes, at the end of an inspection, or at a combination of those events.

Poll Rate: This is the rate in ms at which the cyclic data exchange between the SmartLink and the SmartImage camera takes place. This means that every 50 ms this Modbus Master (camera) will initiate all of its reads and writes to and from the Modbus Slave (Smartlink).

Modbus Master Transfer Setup

Next open the folder for the newly created Modbus Master and double click in the right window pane to create new read and write transfers. Again the properties for each created transfer are displayed in the properties window. Set the properties to the values shown in figure 60.

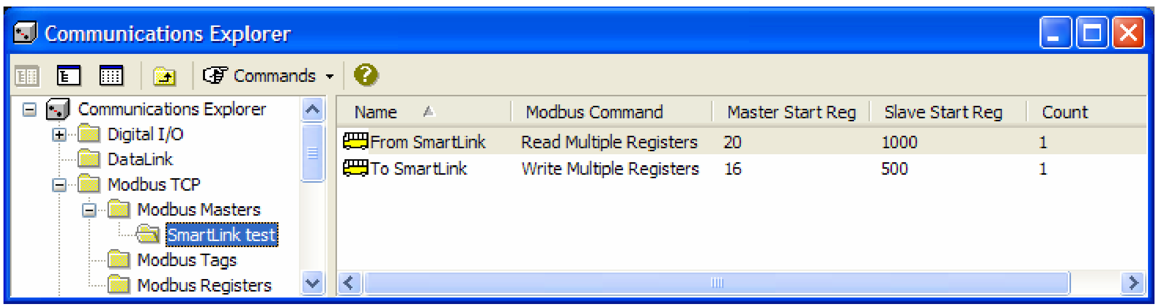


Figure 78: Modbus Master transfer configuration

The following parameters have to be set in the properties window:

Name: A name to identify this transfer.

Command: This pull down box has two options: Write Multiple Registers or Read Multiple Registers. Select Write Multiple Registers

Slave Register Start: That is the number of the register in SmartLink where the data will be written. As mentioned earlier, data that is going to the DeviceNet Master should be stored in DVT registers 1000 through 1511 in the SmartLink. The number in this box is 500 because DVT registers are 1 byte long, while Modbus Registers are 2 bytes long. The data source on the Modbus Master, the camera, is the 2 byte Modbus register space. The SmartLink however uses 1 byte DVT registers, so to write to the 1000^{th} byte in the SmartLink's memory we must write to the equivalent Modbus register, register 500.

The Modbus protocol uses 16-bit registers to transfer data. This means that each Modbus register corresponds to two consecutive DVT registers as shown in **Error! Reference source not found.**. Modbus registers are BigEndian (high byte first).

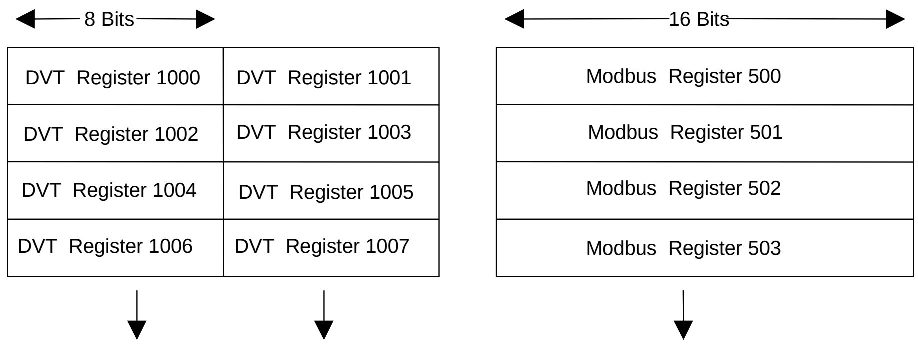
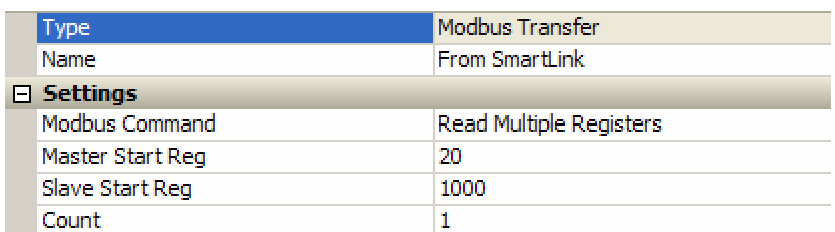


Figure 79: DVT and Modbus registers

Master Register Start: That is the number of the register in the SmartImage camera where the data was written by the Script SoftSensor. In this case, data was written to Modbus register 16.

Number of Registers: Data was written as type Short by the sensor tag, which is two bytes long. Therefore, 1 Modbus register is enough to transfer all the data. Care should be taken when reading/writing more than one byte with the DeviceNet Master from the SmartLink because currently the SmartLink uses the BigEndian convention to write and read data to memory. Using a DeviceNet master with LittleEndian convention to read data in formats that use more than one byte will result in the master reading garbage data. Manipulating the bytes with the Script Tools can change the byte order in the camera and will solve this problem.

The next step is to configure a new Modbus transfer for reading data coming from the DeviceNet Master. The procedure is similar, except this time a Read Multiple Registers command will be created and the registers will be different. The data from the DeviceNet Master will be available in DVT registers 2000 to 2511; therefore, the Slave Register Start in this case is 1000. Once the data is read to Modbus memory in the camera, in this case register 20 (see Figure: 62), it can be accessed from script or by a Monitor Tag similar to the Sensor Tag created earlier. Your read settings should be set as shown below.



Type	Modbus Transfer
Name	From SmartLink
Settings	
Modbus Command	Read Multiple Registers
Master Start Reg	20
Slave Start Reg	1000
Count	1

Figure 80: Modbus Read Multiple Registers

SmartLink DeviceNet Setup

Finally, one parameter needs to be set as part of the SmartLink project file. The dialog box shown in Figure 63 is found under the Project Parameters selection from the Configure drop down menu in the SmartLink application. The Cyclic Data Input Length parameters define the number of bytes that will be exchanged with the DeviceNet network. In this case, we are reading and writing 2 bytes. These numbers should match your network configuration.

Notice that Modbus is set to write/read two DVT registers (two bytes) and the SmartLink is only set to exchange one byte. This is done because the minimum amount of data transmitted with Modbus is one Modbus register, which is equivalent to two DVT registers. If our Sensor Tag was set to only write one byte of data, we would only need to make one byte available to the DeviceNet master and disregard the second byte transferred by Modbus.

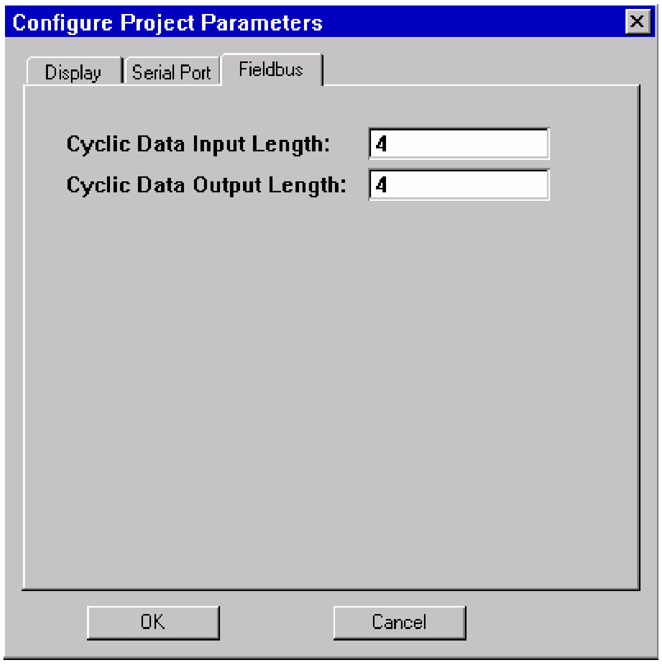


Figure 81: Cyclic Data Transfer Setup for DeviceNet

Troubleshooting the Application

When data is not being transferred from the DeviceNet network to the SmartImage camera (or vice versa), it's helpful to figure out what part of the communication is breaking down. It might be the Modbus transfer between the SmartImage camera and the SmartLink or it could be the card that's transferring the information between the Modbus registers and DeviceNet registers.

DVT recommends connecting to the SmartLink directly with telnet over Ethernet or HyperTerminal over the serial connection. If you're using a touch screen, you will have to use the Ethernet option since the serial port is tied up with the display. To connect over the serial line, start up HyperTerminal using the instructions on pages 16-17 of this manual. Instead of cycling power on the SmartLink, simply hit the Enter key and observe the question mark prompt.

To connect to the SmartLink via Ethernet, start your telnet program by going to the Start menu, selecting Run, and typing Telnet. Hitting the Okay button should bring up a telnet screen. On Windows 2000 and XP, you'll be looking at a command prompt called Microsoft Telnet. Type:

open <IP Address of SmartLink> 5100

and hit enter. For example: open 192.168.0.232 5100

You should be greeted with System Terminal Driver and a question mark prompt.

In either case, once you have the question mark prompt, you can enter a command to read the register out of the SmartLink. To verify the Modbus master is working, you may want to check the contents of register 1000 on the SmartLink. At the command prompt, type:

#Rq1000I

You should get a $ followed by the value in the register. On the next line should be a %0, meaning that the command was successfully executed. Note that the 'I' returns an integer which is 4 DVT registers or 2 Modbus registers. You'll have to figure out if the correct value is getting across. Checking that the SmartImage camera is reading the values correctly out of the SmartLink is done in a similar way. Since the camera is reading starting at SmartLink register 2000, set a value in the following way:

#Rs2000I1234

Setup a monitor tag of the appropriate data type to read Modbus register 20 to test whether the data written to SmartLink register 2000 was transferred back to the camera.

Chapter 7 - Using the VDX Protocol with SmartLink

Introduction

The VDX driver is used for communications between SmartImage cameras and PROFIBUS or DeviceNet enabled SmartLink devices. The SmartLink device, as an intermediary, will exchange information with a PROFIBUS or DeviceNet master through these networks and will use the VDX driver to relay the information to the SmartImage camera.

The Protocol

The VDX driver's protocol consists of input and output telegrams (from the point of view of the VDX server) exchanged between the server (SmartImage camera) and the client (e.g. SmartLink unit) over TCP/IP on port 7500 in a cyclic fashion. The telegrams are 64 bytes long although only 32 bytes are used to communicate with a SmartLink device. The server would wait for a client to connect and after that it would not accept any more connections from other clients. When the SmartImage camera is running inspections, the VDX server will send an output telegram after every inspection. At this point the server will also process any input telegram available. The input telegrams will also be read every 100 milliseconds between inspections.

There are two parts to each input and output telegram: certain I/O functions are mapped to the first four bytes of these packets; the other 28 bytes can hold any value of any of four different data types to be used for multi-purpose data transfer. The Script Reference manual and online help files contain more information on the data structure and the script syntax to take advantage of VDX.

Setting up the SmartLink

As mentioned before, the VDX driver is an alternative to Modbus communications when exchanging information between a SmartImage camera and a PROFIBUS or DeviceNet enabled SmartLink. In this section we will cover how to setup the SmartLink so that it can use the VDX protocol serving as a link between the PROFIBUS or DeviceNet network and the SmartImage camera. The next paragraph gives an overall picture of what happens in the communication process.

A script running in the SmartImage camera writes data to registers in the Sensor memory. This can happen at the same rate that the inspections are running (Script SoftSensor) or asynchronously (Background Script). Data from the SmartImage camera is transferred to the SmartLink via the VDX driver. This is a cyclic process where the camera works as a server writing to and reading from the SmartLink's internal registers after every inspection (the reading also occurs in intervals of 100 ms between inspections). Finally, data in the SmartLink is exchanged with the DeviceNet or PROFIBUS network. This process is illustrated in Figure 82.

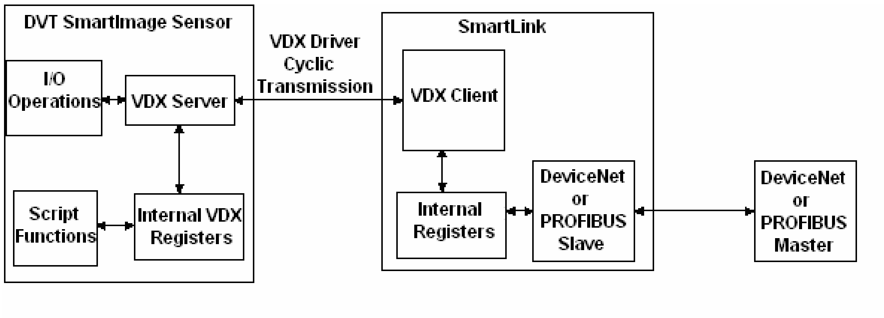


Figure 82: VDX Data Flow Process

On the DeviceNet or PROFIBUS side, the SmartLink needs to be configured by specifying the length for input and output fieldbus packets. These packets are not related to the VDX driver telegrams. They depend on the data being exchanged between the SmartLink and the DeviceNet or PROFIBUS masters. The master can use all 32 bytes of information contained in each VDX telegram for each SmartImage camera or just use what is needed (it can even exchange 256 bytes of data which corresponds to 32 bytes for 8 Sensors as we will see later). Figure 83 shows how to configure the size of these packets in the SmartLink device. This dialog box is found under the "Project Parameters" selection from the "Configure" menu in the SmartLink application.

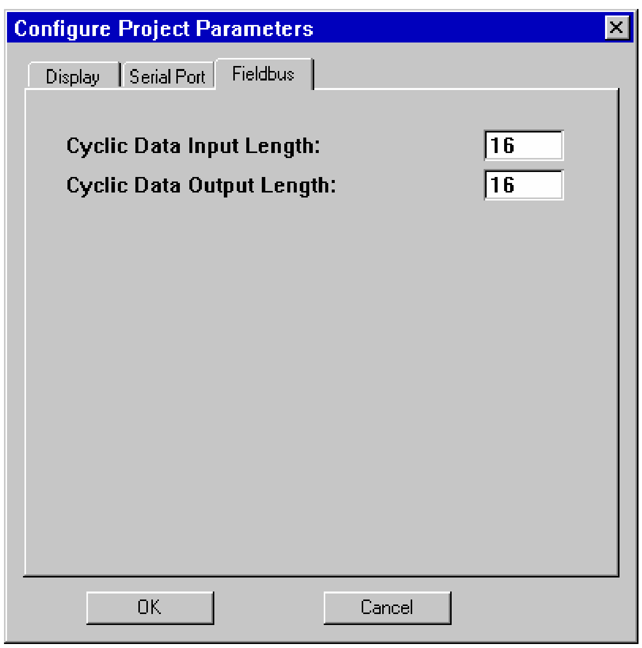


Figure 83: Fieldbus Parameters Dialog Box

The SmartLink device can be configured to receive VDX packets from up to eight SmartImage cameras. Each SmartImage camera is assigned a block of registers within the internal memory in the SmartLink reserved for fieldbus data exchange. The block assignment for each SmartImage camera is performed on the SmartLink project when creating the "Connection" component under the "Add" menu in the SmartLink application. Figure 84 shows the dialog box configuring a connection to a specific SmartImage camera where the VDX driver has been enabled for that connection. An update rate of 200 ms will be used to update information on the VDX client. Block 1 out of eight possible ones has been selected for this SmartImage camera connection.

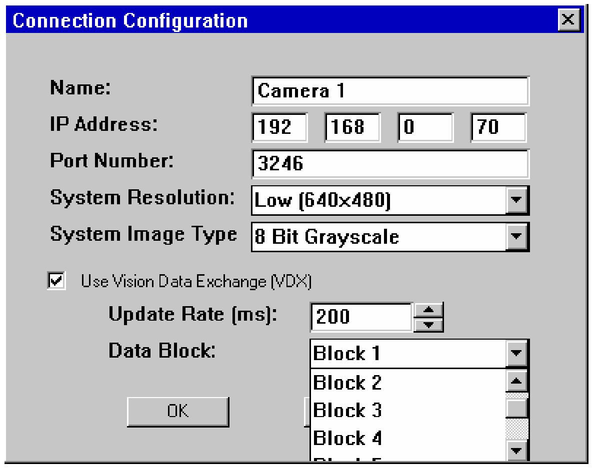


Figure 84: Connection Configuration Dialog Box

In this case the data in the PROFIBUS and DeviceNet packets is directly mapped to the VDX telegrams (e.g. byte 0 of the PROFIBUS output packet will be equivalent to byte 0 of the VDX input telegram). This is because information from this SmartImage camera will be located in Block 1 of the internal SmartLink space dedicated for fieldbus data. Figure 86 shows where in the SmartLink memory this section resides.

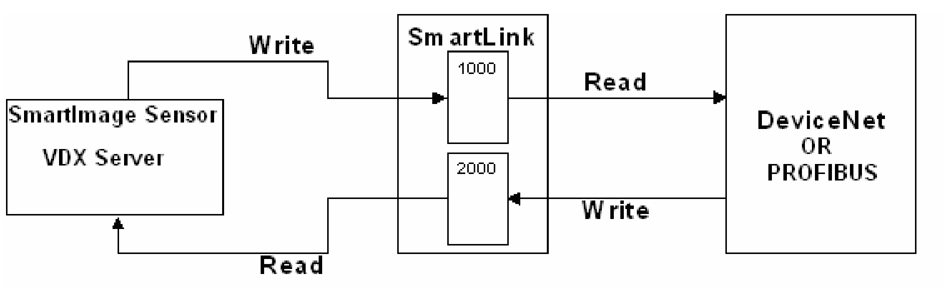


Figure 85: SmartLink Memory Map

As we can see the fieldbus data is mapped to SmartLink's internal registers beginning in register 1000 for fielbus output packets and register 2000 for fielbus input packets.

Since the SmartLink can be configured to communicate with up to eight SmartImage cameras through the VDX driver's protocol, a PROFIBUS or DeviceNet master can exchange information with all eight Sensors at once. The blocks in the SmartLink assigned to each Sensor are stacked back to back in memory as the next figure illustrates:

Fieldbus Data	SmartLink Block	VDX Packets	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0	Block 1	Byte 0	I/O data for SmartImage camera 1							
Byte 1		Byte 1	I/O data for SmartImage camera 1							
Byte 2		Byte 2	I/O data for SmartImage camera 1							
Byte 3		Byte 3	I/O data for SmartImage camera 1							
Byte 4		Byte 4	Multi-purpose data for SmartImage camera 1: Index 0							
Byte 5		Byte 5	Multi-purpose data for SmartImage camera 1: Index 1							
...		...	...							
Byte 31		Byte 31	Multi-purpose data for SmartImage camera 1: Index 27							
Byte 32	Block 2	Byte 0	I/O data for SmartImage camera 2							
Byte 33		Byte 1	I/O data for SmartImage camera 2							
Byte 34		Byte 2	I/O data for SmartImage camera 2							
Byte 35		Byte 3	I/O data for SmartImage camera 2							
Byte 36		Byte 4	Multi-purpose data for SmartImage camera 2: Index 0							
Byte 37		Byte 5	Multi-purpose data for SmartImage camera 2: Index 1							
...		...	...							
Byte 63		Byte 31	Multi-purpose data for SmartImage camera 2: Index 27							
...	...	...	...							
Byte 224	Block 8	Byte 0	I/O data for SmartImage camera 8							
Byte 225		Byte 1	I/O data for SmartImage camera 8							
Byte 226		Byte 2	I/O data for SmartImage camera 8							
Byte 227		Byte 3	I/O data for SmartImage camera 8							
Byte 228		Byte 4	Multi-purpose data for SmartImage camera 8: Index 0							
Byte 229		Byte 5	Multi-purpose data for SmartImage camera 8: Index 1							
...		...	...							
Byte 255		Byte 31	Multi-purpose data for SmartImage camera 8: Index 27							

Figure 86: SmartLink reserved VDX memory blocks

As an example, if we need to trigger an inspection for SmartImage camera assigned to block 3 in the SmartLink, the PROFIBUS or DeviceNet master will need to write a 0 and then a 1 (rising edge trigger) to byte 64 of its output data packet. This will cause bit 0 of byte 0 of block 3 to activate, which will be reflected on bit 0 of byte 0 for the VDX input telegram of that particular SmartImage camera.

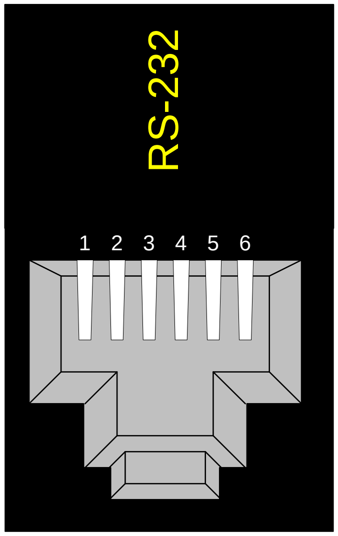
Finally, note that most I/O functions (like “Trigger” in the example above) work on the rising edge of that signal (change of bit value from 0 to 1). However other signals like “Trigger Mode” and “Run Mode” depend on the state of the bit at each moment. For this reason, these bits need to be maintained, that is, written to on each fieldbus data packet to keep the SmartImage camera in the desired mode. For instance, if it is desired for the SmartImage camera to be always in external trigger mode and running inspections, the PROFIBUS or DeviceNet master needs to write a value of 1 on each output packet to bits 5 and 6 of byte 0 for the VDX input telegram of that Sensor.

Summary

In this chapter we have seen how the new VDX driver works. We described the specifications for the protocol. We illustrated how to access the VDX individual bytes of information from the SmartImage camera through provided script functions. From the SmartLink point of view, we saw how to enable the VDX transmissions for a particular SmartImage camera through the “Connection” component in the SmartLink application. We described how the VDX packets from eight different SmartImage cameras are stacked in memory and can be accessed by the DeviceNet or PROFIBUS device.

Appendix A: Pin Outs

RS 232 serial port (RJ-12 connector)

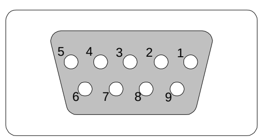


PIn No.	Signal
1	NC
2	Tx
3	GND
4	Rx
5	GND
6	NC

View of Connector Socket on the SmartLink Unit

Figure 87: SmartLink RS-232 Pin Out

RJ-12 to DB-9 Serial Adapter:



PIn No.	Signal
1	NC
2	Tx
3	Rx
4	NC
5	GND
6	NC
7	NC
8	NC
9	NC

Figure 88: Serial Adapter Pin Out

Warning: The pin out shown above corresponds to the SmartLink's serial pin functions. Do not confuse this with the PC's COM port pin assignment. With respect to the pin out above, the Tx and Rx lines are crossed on the PC's COM port. This allows the PC's Tx to go into the SmartLink's Rx and vice versa.

Appendix B: Calibrating a Touch Screen

These calibration instructions are intended for the Microtouch-3M ClearTek MT3000, ELO SmartSet, and Dynapro SC3 touchscreens only.

Configuring the Driver

The driver for the touch screen is configured from the 'Configure...Project Parameters' dialog. A new tab has been added there for configuring the serial port on the SmartLink hardware.

Note: This will only affect the serial port on the hardware, not on the PC when running as an Emulator

The configuration tab, shown in Figure 89, is much like the dialog in Intellect. It contains a drop list to select the driver to use along with controls for configuring the baud rate, the data bit, the stop bit, the parity, the flow control, and the echo.

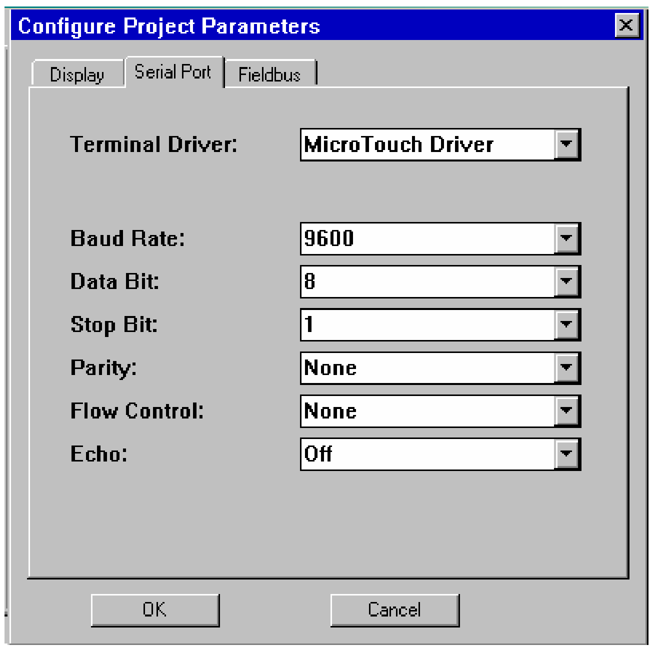


Figure 89: Serial Port Configuration Tab

For the touch screen tested the default settings are:

Driver	Microtouch	Elo	Dynapro SC3
Baud Rate	9600	9600	2400
Data Bits	8	8	8
Stop Bits	1	1	1
Parity	None	None	None
Flow Control	None	None	None
Echo	Off	Off	Off

The serial port is changed to the user settings either when a project is downloaded from the PC or when a project is loaded from flash memory upon power-up. These are the only ways the user can change the serial port settings.

Note: Depending on the driver currently running on the serial port when changes are made, the time it takes to change to the new settings will vary. Because of this, it is best to leave the touch screen unconnected to the SmartLink hardware serial port for a minute or two after the change has been made.

Note: Since the desired baud rate for the MicroTouch screens (9600) is different than the baud rate used by the SmartLink emulator for downloading new firmware, it will not be able to automatically place the hardware in Diagnostics Mode. To download new firmware, the user will need to manually place the system in Diagnostics Mode.

To place the System in Diagnostics Mode:

- Connect to the hardware over a Ethernet terminal (at port 5100) and enter the command #YR3742.
- Connect to the hardware over a serial terminal at the same settings as the touch screen driver. Enter *T- to temporarily change to the system driver. Enter #YR3742.
- Connect to the hardware over a serial terminal at the default system settings (38400, 8, 1, None, None). Hold down the + key while cycling power on the hardware.
- Download a new project with the serial port configured to System and 38400, 8, 1, None, None. After this loads the emulator will be able to automatically place the system in diagnostic mode and download new firmware as usual.

Connecting to the Touch Screen

The physical connection between the SmartLink hardware and the touch screen may require the use of a Null Modem and/or other gender and plug adaptors. The connection necessary for the monitor tested is shown in Figure 90.

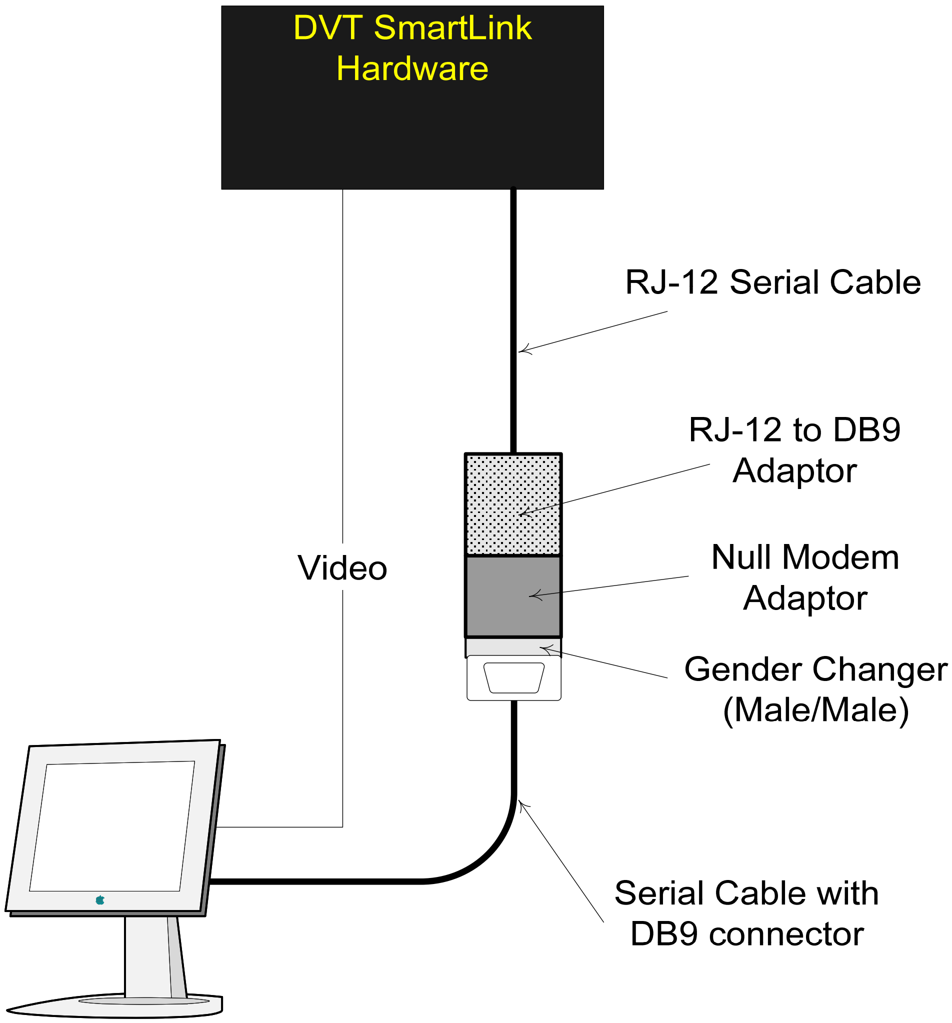


Figure 90: Serial Connection Series for Connecting Between the Touch Screen Monitor and the DVT SmartLink

Calibration

The touch screen will need to be recalibrated every so often during normal use. This may be done through the SmartLink hardware by placing a button on the screen configured to start calibration when pressed. See the document on using the Button Controls for more information.

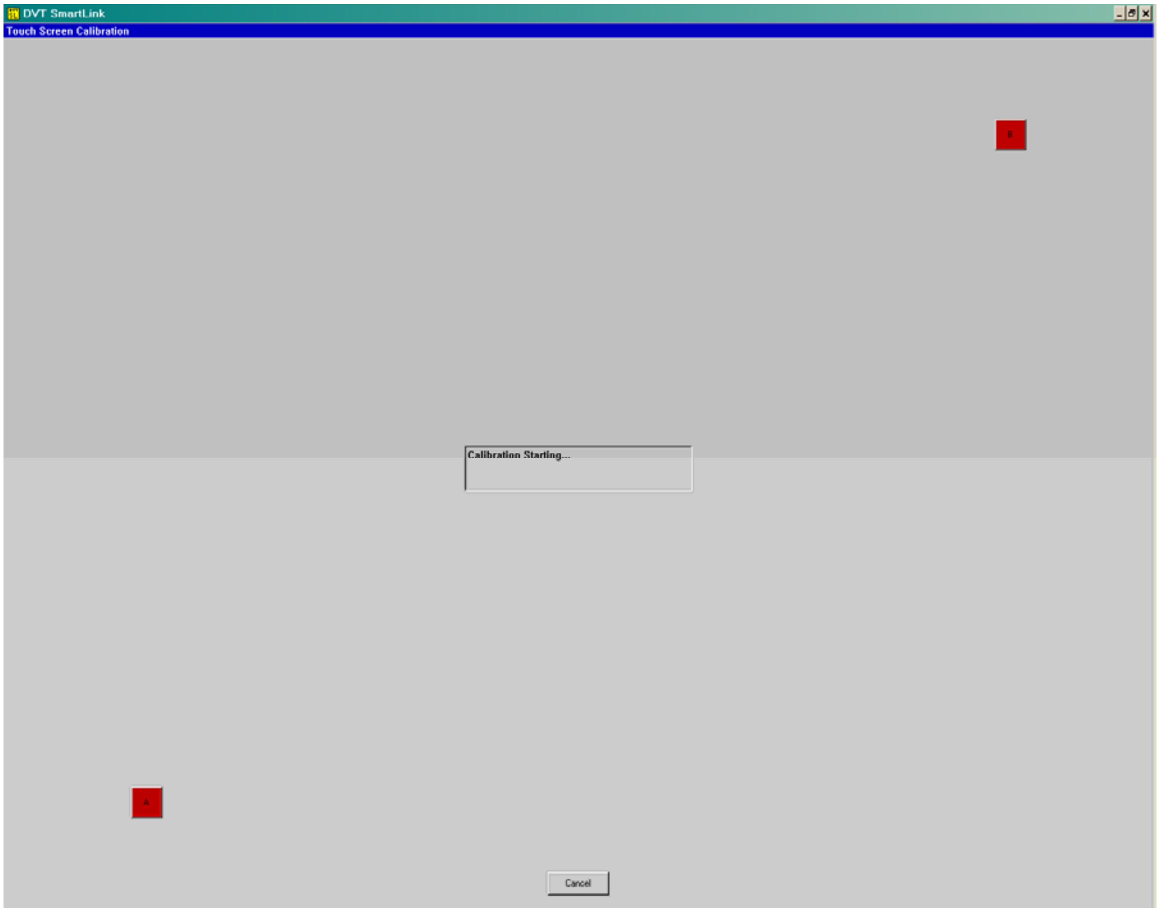


Figure 91: Touch screen calibration window. Contains two calibration targets, a message box, and a cancel button.

Touch screen calibration starts by showing a target window, Figure 91. This window consists of two colored targets, A and B; a message box; and a cancel button. The message box will provide feedback to the user about the progress of the calibration process.

To calibrate the touch screen:

- Press the button configured to start the calibration process.
- Once the dialog in Figure 91 appears wait for the startup procedure to finish. This will be indicated by a message of Press Button A appearing in the message box.
- Press button A using the method recommended in the Touch Screen manual:
- Place the finger on the screen somewhere away from the target.
- Drag the finger to the middle of the target and hold for a second.
- Release the finger.

Perform the same process for button B when the message Press Button B appears.

To cancel the calibration procedure at any time, press the Cancel button.

If an invalid input is entered the calibration process will fail and exit returning the monitor to using the previous calibration values.

Note: On the Microtouch and ELO products, a separate SmartLink project is recommend for calibration. This avoids the possibility of inadvertently performing a calibration during normal run conditions.

Warning: The calibration information for the DynaPro/CTC touch screens is stored in the SmartLink. As a result, the touch screen MUST be powered up before the SmartLink. If the SmartLink is powered up first, the controller will use default values and not produce the desired results.

Note: It is recommended that a very large calibration button be created for calibration. This will prevent the button from being missed if the initial calibration is very far off.

Appendix C: Mouse Driver Configuration

A mouse pointer device can be connected directly to a SmartLink to provide utilization of the buttons and container functions within the SmartLink. In order to provide mouse pointing functionality two things must be configured: the SmartLink serial port and the mouse hardware.

In order for a mouse to communicate with a SmartLink, the mouse must provide power to the SmartLink. Supplying power to the SmartLink serial port is accomplished though a PS/2 to Serial powered converter. This powered PS/2 to serial converter (model number: VIP-327-PS) can be purchased from Vetra systems (www.vetra.com). Other devices that are needed to make the connection to the SmartLink are a PS/2 mouse, a null modem connector, and a gender changer.

On the software side, the SmartLink's serial port must be configured as a mouse driver. To do this, go to the Configure|Project Parameters Menu and select the Serial Port Tab. The settings necessary for the recommended powered mouse converter are Mouse Driver, 1200 Baud Rate, 7 Data Bits, 1 Stop Bit, No Parity, No Flow Control, and Echo Off.

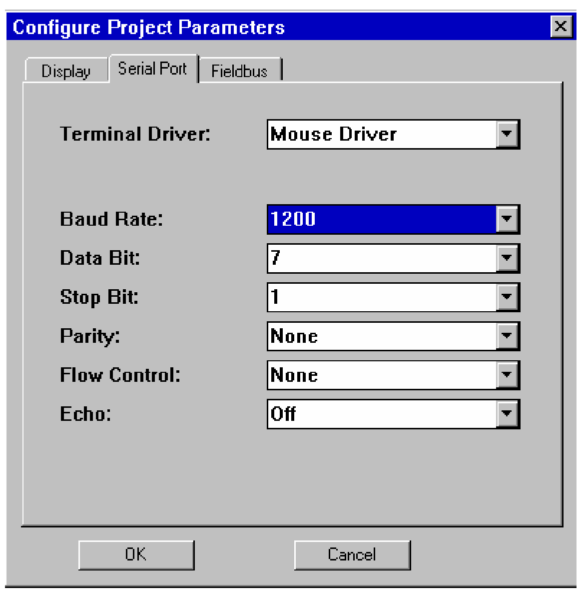


Figure 92: SmartLink serial port settings for the recommended powered mouse converter.