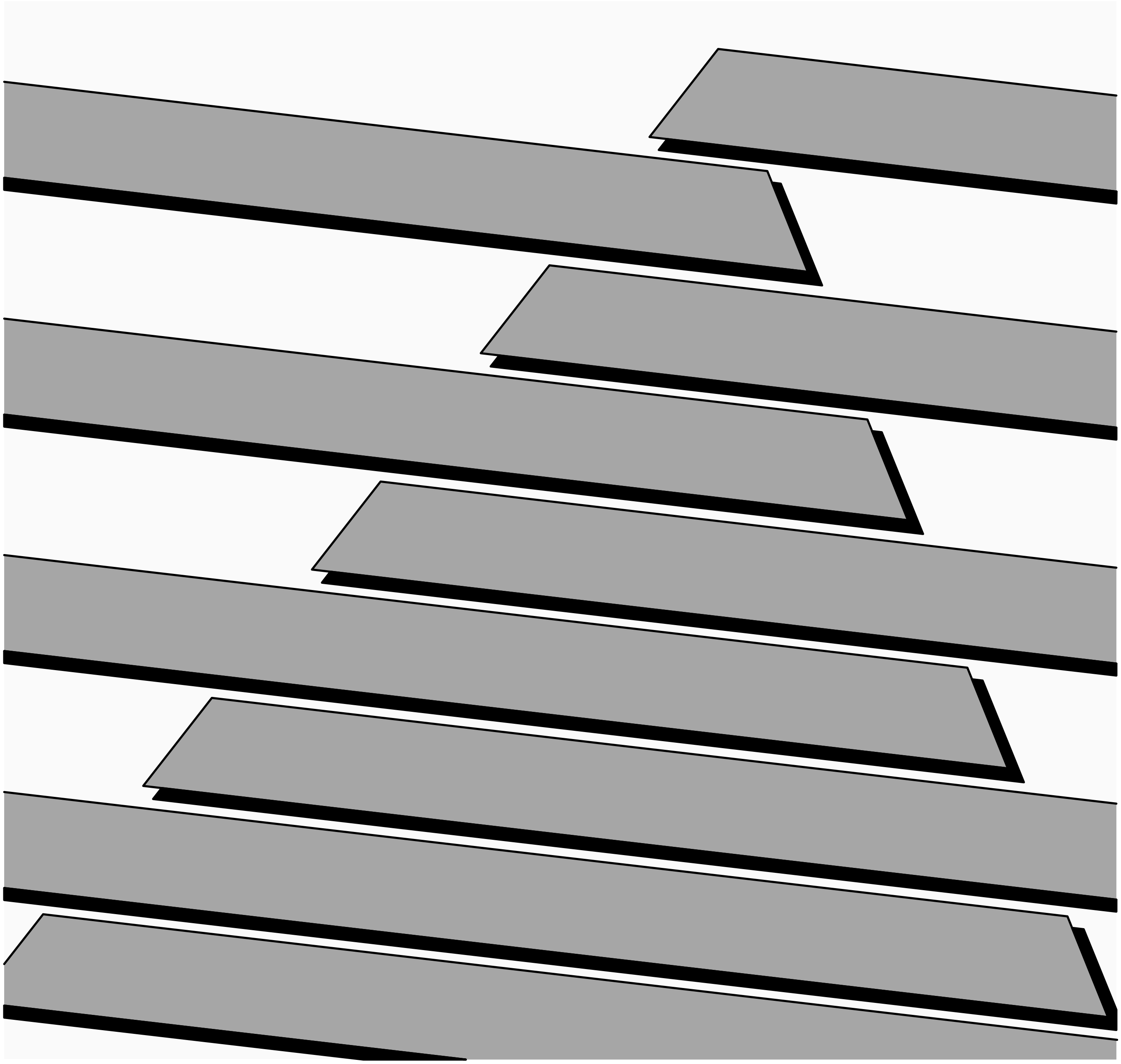
ALLEN-BRADLEY

Bulletin 5370 Color CVIM™ Configurable Vision Input Module

Communications Manual



Important User Information

Solid state equipment has operational characteristics differing from those of electromechanical equipment. “Safety Guidelines for the Application, Installation and Maintenance of Solid State Controls” (Publication SGI-1.1) describes some important differences between solid state equipment and hard–wired electromechanical devices. Because of this difference, and also because of the wide variety of uses for solid state equipment, all persons responsible for applying this equipment must satisfy themselves that each intended application of this equipment is acceptable.

In no event will the Allen-Bradley Company be responsible or liable for indirect or consequential damages resulting from the use or application of this equipment.

The examples and diagrams in this manual are included solely for illustrative purposes. Because of the many variables and requirements associated with any particular installation, the Allen-Bradley Company cannot assume responsibility or liability for actual use based on the examples and diagrams.

No patent liability is assumed by Allen-Bradley Company with respect to use of information, circuits, equipment, or software described in this manual.

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

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

Attentions help you:

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

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

Using This Manual

Chapter 1

Chapter Objectives 1–1
Software Revision 1–1
Overview of this Manual 1–1
Intended Audience 1–2
Related Publications 1–3
How to Use this Manual 1–4
Nomenclature 1–4
Trademarks 1–4

Introduction

Chapter 2

Chapter Objectives 2–1
How is Data Stored in the Color CVIM Module? 2–1
How Does the Host DeviceRead Configuration/Results Information? . 2–1
Remote I/O (Node Adapter) 2–3
RS–232 Ports 2–3
Local I/O 2–3
Pyramid Integrator Backplane 2–3
What Types of Information Can Be Communicated? 2–4
Discrete Bit Information 2–4
Results Blocks 2–5
Configuration Blocks 2–5
Communications Cables 2–5
Memory Addressing 2–5
Host Designation 2–7

Using Local I/O

Chapter 3

Chapter Objectives 3–1
Equipment Connections 3–1
Planning OutputLine Assignments 3–2
Using the Output LinePlanning Sheet 3–3
Using Output Signal Timing Data 3–7
Planning Output Line Connections 3–11
Connections to RS–232 Ports (2801–N27 Interface Box) 3–11
Connections to RS–232 Port (2801–N21 Interface Box) 3–12
Color CVIM Module I/O Interface Box Connections 3–13
Connections to 2801–JMB Interface 3–14

Using the Remote I/O Link

Chapter 4

Chapter Objectives 4–1
Remote I/O Communications 4–1
What Functions can be Performed over the Remote I/O Network? . . . 4–6
Obtaining Inspection Result Information 4–7

Using the Remote I/O Link

Chapter 4 *(continued)*

Color CVIM Module Configuration Instructions 4–7
Accessing Discrete Bit Information 4–8
Note to PLC–2 Users: 4–8
Example Program for Accessing/Setting Discrete Bit Data 4–12
Example Program For Accessing/Setting Discrete Bit Data (cont'd) . . 4–13
Example Program For Accessing/Setting Discrete Bit Data (Cont'd) . 4–14
Accessing Results and Configuration Information 4–15
Transferring Results Blocks 4–15
Configuring Results Block 6 and Statistics Block Formats 4–17
Converting Results Data 4–18
Transferring Configuration Blocks 4–20
Example Program for Accessing Results Data 4–21
Example Program for Accessing Results Data, Cont'd 4–22
Example Program for Accessing Results Data, Cont'd 4–23
Example Program For Accessing Configuration Data 4–24
Example 6008–SI Program 4–27
Example 6008–SI Program (cont'd) 4–28

Using the RS–232 Ports

Chapter 5

Chapter Objectives 5–1
RS–232 Communications 5–1
ASCII and DF1 Protocols 5–2
Equipment Connections 5–3
What Functions can be performed over the RS–232 Interfaces? 5–4
Color CVIM Module Configuration Instructions 5–5
ASCII Protocol 5–6
Overview 5–6
ASCII Character Set 5–7
Command Structure 5–7
XON/OFF Flow Control 5–8
Deactivate Forces 5–8
Echoing Data 5–9
Enable/Disable Outputs 5–9
Forcing Local I/O 5–10
Loading Configurations 5–11
Lock Command 5–12
Unlock Command 5–13
Read Output Status 5–13
Read Configuration Blocks 5–14
Read Inspection Results 5–16
Save Configuration 5–19
Select Image Displayed 5–21
Set Configurable Results 5–22

Using the RS–232 Ports

Chapter 5 ***(continued)***

Set/Read Configurable Statistics 5–23
Trigger Operation 5–25
Write Configuration (W) Write Configuration (WC) 5–25
Command Summary 5–27
Explanation of ASCII Programming Example 5–29
ASCII Programming Example 5–31
DF1 Protocol 5–33
What is DF1? 5–33
DF1 Character Set 5–33
Command Structure 5–34
Command Structure (cont'd) 5–35
ACK/NAK, BCC Characters 5–35
Block Check Character 5–36
Deactivate Forces 5–37
Echoing Data 5–38
Enable/Disable Outputs 5–39
Forcing Local I/O 5–39
Loading Configurations 5–41
Lock Command 5–42
Read Output Status 5–42
Read Configuration Block Command 5–43
Read Results Command 5–47
Save Command 5–50
Select Image Displayed 5–51
Set Configurable Results 5–53
Set/Read Configurable Statistics 5–54
Trigger Operation Command 5–56
Unlock Command 5–56
Write Configuration Blocks 5–57
Command Summary 5–60
DF1 Programming Example 5–62

Using the Pyramid Integrator Backplane

Chapter 6

Chapter Objectives 6–1
What Information can be Accessed? 6–1
Host Designation 6–1
What Functions can be Performed Over the Backplane? 6–2
Color CVIM Module Configuration Instructions 6–2
Obtaining Inspection Result Information Using a PLC–5/250 6–3
Manipulating Configuration Data Using a PLC–5/250 6–4
Sample PLC–5/250 Program 6–6
Obtaining Inspection Result Information Using a MicroVAX
Information Processor 6–13

Results/Configuration Data Overview

Appendix A

Introduction A–1
Overview A–1
Fractional Notation A–2
32 Bit Integer Format A–4

Discret Bit Results Bits (Host Input and Output Bits)

Appendix B

Introduction B–1
Discrete Bit Inputs B–1
Discrete Bit Outputs B–19

Numerical Results Data

Appendix C

Results Block Overview C–1
Block Transfer Signature C–2
Results Block #1 . . . . C–3
Results Block #2 . . . . C–6
Results Block #3 . . . . C–8
Results Block #4 . . . . C–11
Results Block #5 . . . . C–13
Results Block #6 . . . . C–15
Statistics Block C–20

Configuration Data

Appendix D

Configuration Block Overview D–1
Configuration Block #1 . . . . D–2
Configuration Blocks 2 & 3 . . . . D–4
Configuration Blocks 4–35 . . . . D–6
Configuration Block 36 . . . . D–7
Configuration Blocks 37–39 . . . . D–8
Configuration Blocks 40–45 . . . . D–9
Configuration Blocks 46–109 . . . . D–11
Configuration Blocks 110–205 . . . . D–13
Configuration Blocks 206–213 . . . . D–15

ASCII Conversion Table

Appendix E

Tables

Table 1.A Related Publications 1–3
Table 2.A Example of Addressing Results Block 1 2–6
Table 3.A Example Color CVIM Output Line Planning Sheet
Output Line Functions and Tool Assignments 3–3
Table 3.B Color CVIM Module I/0 Connector 3–13
Table 3.C I/O Interface Box (Catalog No. 2801–N21):
RS–232 Connector with Series A Color CVIM Module 3–13
Table 3.D I/O Interface Box (Catalog No. 2801–N27):
RS–232 Port A Connector with Color CVIM Module 3–14
Table 3.E I/O Interface Box (Catalog No. 2801–N27):
RS–232 Port B Connector with Color CVIM Module 3–14
Table 3.F Color CVIM I/O Output Numbering 3–16
Table 3.G Color CVIM Output Line Planning Sheet
Output Line Functions and Assignments 3–17
Table 4.A Color CVIM Module Remote I/O Inputs
(Color CVIM Module to PLC) if Color CVIM Module
is Rack 02 (First Discrete Bits Results) 4–9
Table 4.B Color CVIM Module Remote I/O Inputs
(Color CVIM Module to PLC) if Color CVIM Module
is Rack 02 (Second Discrete Bits Results) 4–10
Table 4.C Color CVIM Module Remote I/O Outputs (PLC to Color
CVIM Module) if Color CVIM Module is Rack 02 4–11
Table 5.A ASCII Command Summary 5–27
Table 5.B DF1 Command Summary 5–60
Table 6.A Shared Memory Overview 6–3
Table B.1 Discrete Bit Inputs B–2
Table B.2 Backplane Handshake Bits B–19
Table B.3 Discrete Bit Outputs B–20
Table C.1 Numerical Results Data–Results Block 1 . . . . C–3
Table C.2 Numerical Results Data–Results Block 2 . . . . C–6
Table C.3 Numerical Results Data– Results Block 3 . . . . C–8
Table C.4 Numerical Results Data–Results Block 4 . . . . C–11
Table C.5 Numerical Results Data–Results Block 5 . . . . C–13
Table C.6 Results Block #6 . . . . C–15
Table C.7 Programmable Results/Statistics Block Configuration . . . C–16
Table C.8 Statistics Block C–20
Table D.1 Configuration Block #1– System Environment D–2
Table D.2 Configuration Block #2 – Camera Definition D–4
Table D.3 Configuration Block #3 – Camera Definition D–6
Table D.4 Configuration Blocks #4–35 – Color Definition D–6
Table D.5 Configuration Block 36 – Gray Scale Definition D–7
Table D.6 Configuration Blocks #37–39 – Reference Lines 1–3 . . . D–8
Table D.7 Configuration Blocks #40, 42, and 44 –
Reference Windows 1–3 . . . . D–9

Tables *(continued)*

Table D.8 Configuration Blocks #41, 43, & 45 – Reference Windows 1–3 . . . . D–10
Table D.9 Configuration Blocks #46–109 – Gages 1–64 . . . . D–11
Table D.10 Configuration Blocks #110, 112 . . . & 204 – Windows 1–48 . . . . D–13
Table D.11 Configuration Blocks #110, 112 . . . & 204 – Windows 1–48 . . . . D–14
Table D.12.Configuration Block #111, 113 . . . & 205 – Windows 1–48 . . . . D–14
Table D.13 PoIygon Configuration Words D–15

Figures

Figure 2.1 Color CVIM Module Communications Ports 2–2
Figure 2.2 Bit Numbering 2–6
Figure 3.1 Local I/O Equipment Connections 3–2
Figure 3.2 Timing Diagram — Pulsed I/O 3–7
Figure 3.3 Timing Diagram — Trigger #2 During Data Valid, Pulsed I/O 3–8
Figure 3.4 Timing Diagram — Non–Pulsed I/O 3–9
Figure 3.5 Timing Diagram– Missed Trigger 3–10
Figure 3.6 Pinouts– Catalog No. 2801–N27 I/O Interface Box 3–11
Figure 3.7 Pinouts– Catalog No. 2801–N21 I/O Interface Box 3–12
Figure 3.8 Local I/O Board (Catalog No. 2801–JMB). 3–15
Figure 4.1 PLC–5 to Color CVIM Module– Remote I/O Link 4–2
Figure 4.2 6008 SI IBM PC/AT Scanner to Color CVIM Module– Remote I/O Link 4–2
Figure 4.3 6008 SV VME Scanner to Color CVIM Module– Remote I/O Link 4–2
Figure 4.4 6008 SQH1/2 Q–BUS Scanner to Color CVIM Module– Remote I/O Link 4–3
Figure 4.5 Mini PLC–2 to Color CVIM Module– Remote I/O Link 4–3
Figure 4.6 PLC–2 to Color CVIM Module – Remote I/O Link 4–3
Figure 4.7 PLC–3 to Color CVIM Module – Remote I/O Link 4–4
Figure 4.8 PLC–5/250 to Color CVIM Module– Remote I/O Link . 4–4
Figure 4.9 Typical Hardware Layout for Remote I/O 4–5
Figure 5.1 RS–232 Equipment Connections. 5–3
Figure 5.2 RS–232 Cabling. 5–4
Figure 5.3 Memory Requirements for Images and Configurations . . 5–12
Figure 5.4 Configuration Block Returned Data Format* 5–15
Figure 5.5 Numerical Results Block Returned Data Format 5–18
Figure 5.3 Discrete Bit Results Returned Data Format 5–18
Figure 5.6 Memory Requirements for Images and Configurations . . 5–20
Figure 5.7 Statistics Block Returned Data Format 5–24

Figures *(continued)*

Figure 5.8 Read Configuration Block Command Bytes 3–29 5–44
Figure 5.9 Configuration Block Returned Data Format 5–46
Figure 5.10 Results Block Returned Data Format 5–49
Figure 5.11 Discrete Bit Results Returned Data Format 5–49
Figure 5.12 Statistics Block Returned Data Format 5–55
Figure 5.13 Write Configuration Block Command Bytes 3–29 5–58
Figure 6.1 32 Bit Long Integer Files 6–5
Figure A.1 Fractional Notation A–3

Chapter 1

Using this Manual

Chapter Objectives

Read this chapter to familiarize yourself with the rest of the manual. You will learn about:

- Contents of the manual.
- Intended audience.
- How to use the manual.

Software Revision

This manual describes how to communicate with a Color CVIM™ module (Catalog No. 5370–CVIMC) with at least firmware revision A01.

Overview of this Manual

This manual explains how to communicate with the Bulletin 5370 Color Configurable Vision Input Module (Color CVIM) module using a variety of peripheral devices.

Chapter	Title	Purpose
1	Using This Manual	Provides an overview.
2	Introduction	Describes the basic options available for communications with the Color CVIM module.
3	Using Local I/O	Describes how to use the discrete I/O module (Catalog No. 2801–JMB).
4	Using the Remote I/O Link (Node Adapter)	Describes how to access data through the remote I/O port with a PLC–2®, or PLC–3®, PLC–5™. Includes sample programs.
5	Using the RS–232 Ports	Describes how to access data through the RS–232 interfaces using ASCII and DF1 formatted commands. Provides sample programs.
6	Using the Pyramid Integrator™ Backplane	Describes how to access shared memory through the Pyramid Integrator™ information processor and/or MicroVAX®. Provides sample programs.

Overview of this Manual (cont'd)

Chapter	Title	Purpose
Appendix A	Results/Configuration Data Overview	Provides an overview of the configuration and results data. Describes data formats.
Appendix B	Discrete I/O Results Bits	Provides a description of the 256 discrete input bits and 256 discrete output bits.
Appendix C	Numerical Results Data	Provides a description of inspection results data.
Appendix D	Configuration Data	Provides information on the configuration blocks.
Appendix E	ASCII Conversion Chart	Provides equivalent values for the ASCII character set.
	Glossary	
	Index	

Intended Audience

This manual was written for an experienced PLC® user or computer programmer. The user of this manual should:

- Know how to program the host device being used to communicate with the Color CVIM module. For example, if you are using a PLC–5 to communicate with the Color CVIM module, you must have a background in programming a PLC–5.
- Know terms common to the computer and programmable controller industries.
- Understand how to operate and configure the Color CVIM module before using this manual. You may not understand many of the terms being used unless you have read the Color CVIM User's Reference Manual, Catalog No. 5370–ND009.

Related Publications

Table 1.A lists related publications that you may require:

Table 1.A
Related Publications

Publication No.	Title	Purpose of Publication
Catalog No. 5370–ND011	Color CVIM Getting Started Manual	Describes the basics of the Color CVIM user interface.
Catalog No. 5370–ND009	Color CVIM User's Reference Manual	Provides step–by–step procedures for the installation, configuration and operation of the Color CVIM module.
1772–6.8.1	PLC–2/20 Programming and Operations Manual	Provides instructions on how to program a PLC–2/20® programmable controller.
1772–6.8.3	PLC–2/30 Programming and Operations Manual	Provides instructions on how to program a PLC–2/30® programmable controller.
1772–6.8.6	Mini–PLC–2/05 Programming and Operations Manual	Provides instructions on how to program a Mini–PLC–2/05® programmable controller.
1772–6.8.2	Mini–PLC–2/15 Programming and Operations Manual	Provides instructions on how to program a Mini–PLC–2/15® programmable controller.
1775–6.7.1	PLC–3 Controller Installation and Operations Manual	Provides instructions on how to program a PLC–3 programmable controller.
1785–6.8.2	PLC–5 Family Processor Manual	Provides instructions on how to program a PLC–5 programmable controller.
5000–2.3	Allen–Bradley Pyramid Integrator Technical Overview	Provides an overview of the Pyramid Integrator.
5000–2.17	Allen–Bradley Pyramid Integrator Technical Description	Provides a technical description of the Pyramid Integrator.
5000–2.20	MicroVAX Information Processor Technical Description	Provides a technical description of the MicroVAX Information Processor.
5000–6.2.10	Allen–Bradley Pyramid Integrator Installation Manual	Provides instructions on installing Pyramid Integrator devices.
5000–6.2.10	Allen–Bradley Pyramid Integrator Start–up and Integration Manual	Provides instructions on how to use Pyramid Integrator devices.

How to Use this Manual

When using this manual, we recommend that you do the following.

1. Become familiar with the Color CVIM module by reading the User's Reference Manual, Catalog No. 5370–ND009. If possible, use the Color CVIM module to become familiar with its operation. Only with a thorough understanding of the Color CVIM module will you be able to interpret the data that is stored in its memory.
2. Read Chapters 1 and 2 of this manual. After reading these introductory chapters, you will be able to determine which of the remaining chapters, some or all, you will need to read. See note below.

   Important Note: This manual is divided into chapters. It is not necessary to read all of the information contained in this manual. Chapters 1 and 2 are mandatory. You can read the remaining chapters on a "need to know basis" depending upon the information you want to read or write and the type of host device you are using.

3. Use the programming examples provided in each section as a guide to create your own programs. In some applications, you may be able to simply modify the example provided.

   These examples are included solely for illustrative purposes. Because the many variables and special requirements associated with any particular installation, Allen–Bradley Company cannot assume responsibility or liability for their applicability to your own situation.

Nomenclature

In this Chapter and in subsequent chapters we refer to the Bulletin 5370 Configurable Vision Input Module as the Color CVIM module. In some tables we use the abbreviation "PI" to indicate the PLC–5/250 Pyramid Integrator. We have also provided a glossary in the back of this manual. Use this glossary whenever you are unsure of the meaning of a word.

Trademarks

In this manual, we use the following trademarks:

CVIM™ is a trademark of Allen–Bradley
PLC®, PLC–2®, PLC–2/20®, PLC–2/30®, PLC–2/05®,
PLC–2/15®, and PLC–3® are registered trademarks of Allen–Bradley
PLC– 5™, PLC–5/250™ are trademarks of Allen–Bradley
Pyramid Integrator™ is a trademark of Allen–Bradley
Dataliner™ is a trademark of Allen–Bradley
RediPANEL™ is a trademark of Allen–Bradley
DATAMYTE® is a registered trademark of Allen–Bradley

Microsoft® is a registered trademark of Microsoft Corporation
MicroVAX® is a registered trademark of Digital Equipment Corporation
GW BASIC™ is a trademark of Microsoft Corporation

Chapter 2

Introduction

Chapter Objectives

In this chapter we provide you with an overview of the options for communicating with the Color CVIM module. We also describe the types of data that can be accessed or manipulated. The descriptions in this chapter will enable you to determine the type of communications most suitable for your application. You then can proceed to the chapter of this manual that describes the selected option.

How is Data Stored in the Color CVIM Module?

The result and command data that you can access with a host device is stored in an area of Random Access Memory (RAM) inside the Color CVIM module. Configuration data which controls the operating instructions for the Color CVIM module is located in a separate area of memory which can be also be accessed through a host device. Refer to Appendix A for an overview of configuration/results memory. Appendix B, C, and D contain tables listing the information stored in results and configuration memory locations.

How Does the Host Device Read Configuration/Results Information?

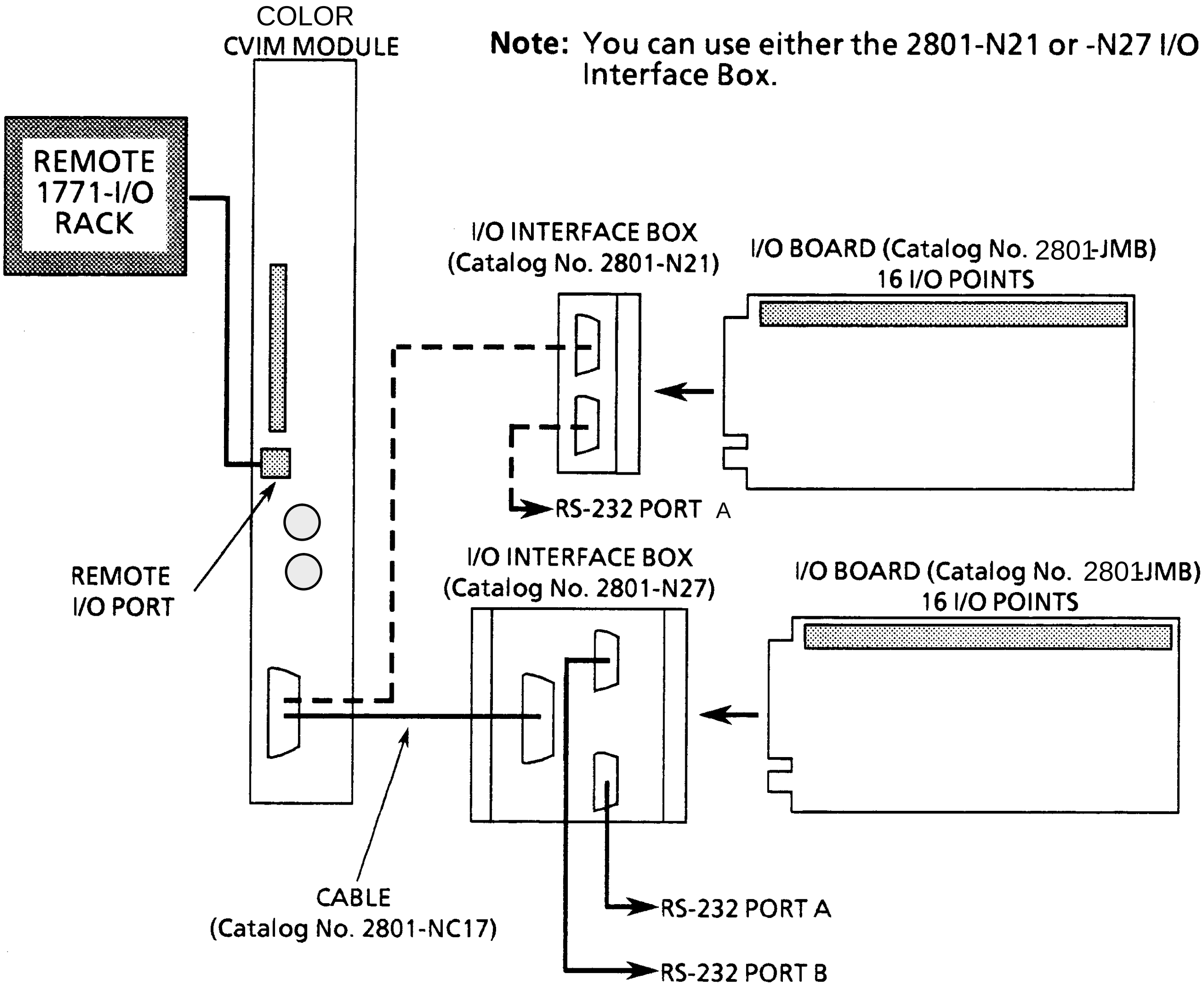
The remainder of this chapter describes the various options you have for accessing this information. Refer to Figure 2.1. In summary, your host device will be linked to the Color CVIM module through one of the following ports:

- Remote I/O (Node Adapter)
- RS–232 Interface(s)
- Pyramid Integrator Backplane
- Local I/O Board

Note: The local I/O board has sixteen discrete I/O lines. Fourteen of these lines are outputs only. One of the remaining lines is for input, and can be connected to a presence–sensing device to trigger an inspection process. The other line is not used.

How Does the Host Device Read Configuration/Results Information? (cont'd)

Figure 2.1
Color CVIM Module Communications Ports



Remote I/O (Node Adapter)

The remote I/O port (RIO) is located on the front of the Color CVIM module as shown in Figure 2.1. Using the remote I/O port, you can connect the following types of devices:

- Allen–Bradley Programmable Controllers (PLC–2, –3, and –5).
- Host Computers which have the Allen–Bradley IBM Bus Scanner (Catalog No. 6008–SI). The 6008–SI bus scanner is compatible with the A–B 6121/22 Industrial Computer, Industrial Terminal (Catalog Nos. 1784– T50, 1784–T35), or other IBM PC/AT compatible devices.

RS–232 Ports

As shown in Figure 2.1, the RS–232 ports are located on the I/O Interface Boxes (Catalog No. 2801–N21, –N27). The I/O Interface Box is connected to the MODULE I/O port on the front of the Color CVIM module. Using the RS–232 interface(s) you can connect a variety of devices which use the RS–232 standard:

- Computers
- Operator Interfaces such as Allen–Bradley Industrial Computers and Terminals with serial ports.
- I/O modules such as the Flexible Interface Module (Catalog No. 2760–RB) or ASCII module (Catalog No. 1771–DA).

Local I/O

As shown in Figure 2.1, the local I/O consists of an I/O Board (Catalog No. 2801–JMB), I/O Interface Box (Catalog No. 2801–N21, –N27), an input and up to 14 output modules as configured by the user. The Catalog No. 2801–NC17 cable connects the I/O interface box to the Color CVIM module.

Pyramid Integrator Backplane

Using the Pyramid Integrator backplane, you can directly communicate data between the Color CVIM module and other devices installed in the Pyramid Integrator chassis:

- Allen–Bradley PLC–5/250
- MicroVAX Information Processor

What Types of Information can be Communicated?

Depending upon the type of interface in use, you can access some or all of the information listed below:

- Warning and Pass/Fail data.
- Numerical inspection results.
- Configuration data.

Discrete Bit Information

With each inspection that the Color CVIM module performs, individual bits are set. There are 256 bits that can be read as inputs to a host device. These bits (part of the inspection results) indicate:

- Master fault.
- Mastership.
- Configuration fault.
- Module Busy flag.
- Missed Trigger flag.
- Results Valid flag.
- Inspection Tool Pass/Fail/Warning flags.

There are 128 bits that can be set as outputs by a host device to control the operation of the Color CVIM module. These bits control:

- Monitor display.
- Toolset selection.
- Enable/disable and force discrete I/O.
- Selection of operation after reject.
- Memory storage location. RAM, EEPROM, RAM Card, or external host memory.

Note: For more information on the 256 discrete input and 128 discrete output bits refer to Appendix B.

Results Blocks

The results data for each inspection are stored in Random Access Memory (RAM) and overwrite the results of the previous inspection. The data stored in results blocks contain information regarding reference windows, inspection gages, inspection windows, etc. For a complete description of the results blocks, refer to Appendix C.

Configuration Blocks

The user developed inspection parameters of the Color CVIM module are stored in the Color CVIM module's memory as configuration blocks. This area of memory can be read or manipulated through the Remote I/O port, RS–232 ports (A & B) or Pyramid Integrator backplane. Refer to Appendix D for a complete description of the configuration blocks and their contents.

Communications Cables

If you are not using the Pyramid Integrator backplane for communications, you will have to physically link the Color CVIM module to the host device. If you need to create a communications cable, refer to the chapter that describes the communications port you are using.

Memory Addressing

Depending upon how you access the Color CVIM module results and configuration memory, you will have to address the data differently. If you refer to Appendix A, B, and C you will notice that separate columns are provided for Backplane, RS–232, and Remote I/O communications:

Note: The RS–232 protocols (ASCII and DF1) do not access data using word and bit addresses. Data is read/written in blocks. We have grouped the RS–232 and Remote I/O ports together in Appendix B, C, and D (where appropriate) for your convenience. You can ignore word and bit addresses if you are using the RS–232 ports (A & B).

Memory Addressing (cont'd)

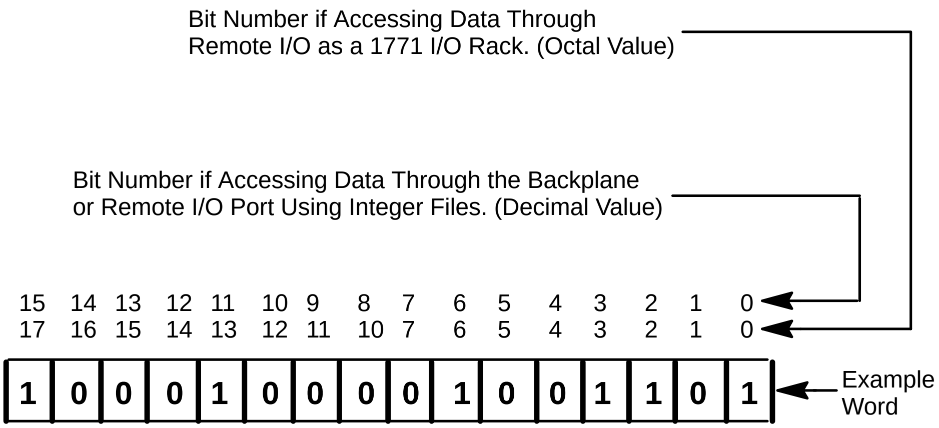
When you communicate through the Pyramid Integrator backplane all of the data words are numbered consecutively and grouped in blocks. When you use the Remote I/O port, you select a specific block and the first word in each block is word #0.

Table 2.A
Example of Addressing Results Block 1

Word Number	
Pyramid Integrator Backplane	RS–232 and Remote I/O
32–95	**0–63**

In addition, PLC I/O bit numbers are entered in octal format when referencing 1771 I/O, while PLC files and backplane communications specify a decimal bit number. Figure 2.2 illustrates how bits are numbered.

Figure 2.2
Bit Numbering


Bit Number if Accessing Data Through
Remote I/O as a 1771 I/O Rack. (Octal Value)
Bit Number if Accessing Data Through the Backplane
or Remote I/O Port Using Integer Files. (Decimal Value)
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
17 16 15 14 13 12 11 10 7 6 5 4 3 2 1 0
1 0 0 0 1 0 0 0 0 1 0 0 1 1 0 1
Example Word

Host Designation

There are four communications ports which you can use simultaneously to access Color CVIM module data (Remote I/O, RS–232 Ports A & B, and Backplane). Only the host can issue commands to control the operation of the Color CVIM module. You can read discrete bits and numerical results information through any of the four communications ports, even through non–host devices.

The Color CVIM can operate with multiple hosts. You can select one host to perform Color CVIM module/host configuration transfers, and another host to perform all other Color CVIM/host operations. The first of these two hosts is referred to as the configuration host (CFG) because it is the sole source of configuration information. The second, referred to as the system host (SYS), can tell the system when to trigger and control the user interface. A system host *cannot* supply a new configuration.

An example of using multiple hosts is to select RS–232 A as the CFG host, and Remote I/O as the SYS host. If you select a single device as both CFG and SYS hosts, then that device must be labeled as such.

Note: Any Color CVIM communications port can be used for reading results block data regardless of whether or not the device connected to the port is selected as a host.

Note: You can select the same host (Stand Alone, Pyramid, Remote I/O, RS–232 A or B) as both the configuration host and the system host.

Chapter 3

Using Local I/O

Chapter Objectives

The objectives of this chapter are to help you plan:

- The *number* of discrete output lines (up to 14) that your application will require.
- The *function* that each output line will perform in your application.
- The *assignment* of analysis tool “results” to output lines.
- The *assignment* of status signals to output lines.
- The *electrical and mechanical connections* of the trigger (input) and output lines to your production equipment.

Equipment Connections

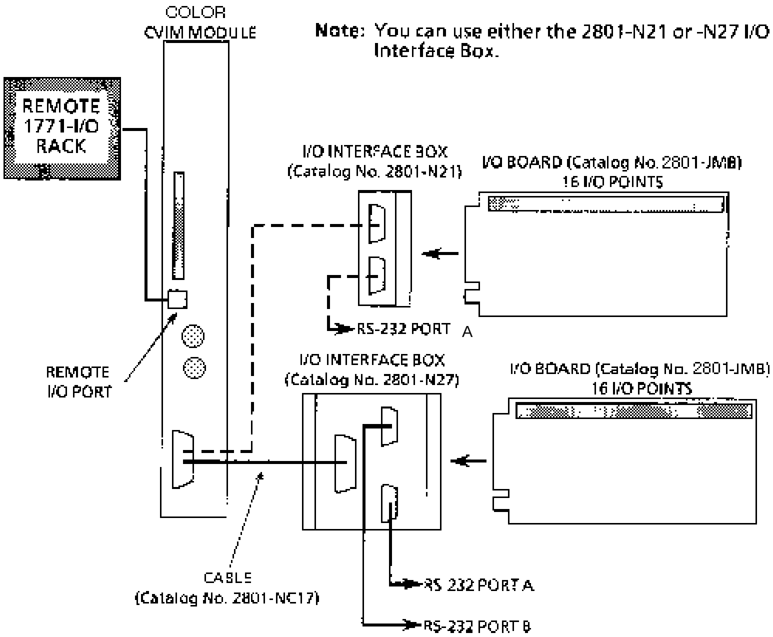
The local I/O consists of:

- I/O Interface Box (Catalog No. 2801–N21, –N27)
- I/O Board (Catalog No. 2801–JMB)
- User specified I/O modules (plug into I/O board)
- Communications Cable (2801–NC17)

As shown in Figure 3.1, the communications cable (Catalog No. 2801–NC17) is connected to the MODULE I/O port on the front of the Color CVIM module and the connector on the I/O Interface box. The I/O board connector slides into the connector slot on the I/O Interface Box.

Equipment Connections (cont'd)

Figure 3.1
Local I/O Equipment Connections



COLOR
CVIM MODULE
Note: You can use either the 2801-N21 or -N27 I/O Interface Box.
REMOTE 1771-I/O RACK
I/O INTERFACE BOX (Catalog No. 2801-N21)
I/O BOARD (Catalog No. 2801-JMB) 16 I/O POINTS
RS-232 PORT A
REMOTE I/O PORT
I/O INTERFACE BOX (Catalog No. 2801-N27)
I/O BOARD (Catalog No. 2801-JMB) 16 I/O POINTS
CABLE (Catalog No. 2801-NC17)
RS 232 PORT A
RS-232 PORT B

Planning Output Line Assignments

This section provides a planning sheet that you can use to lay out the *function* and *tool* assignments for output lines.

The term “function assignment” refers to the type of signal information that you want an output line to carry to your production equipment.

The term “tool assignment” refers to the tool(s) that you assign to an output line.

Note: Tools can be assigned *only* to output lines that you have assigned a “results” *function.* These output lines will carry the “pass/fail” *results* signals from the tools during each inspection.

The next section, *Planning Output Line Connections*, provides electrical and timing diagrams and data. You will need to use these diagrams to correctly identify and connect the output lines to your production equipment.

Using the Output Line Planning Sheet

The Output Line Planning Sheet is a form on which you can lay out your plans for each output line. On this form you can account for:

- The 14 output lines.
- The six output line functions.
- The 64 gages and their warning and fault outputs.
- The 48 windows and their warning and fault outputs.
- The 6 reference tools and their “pass/fail” outputs.
- The light probe with its separate red, green, and blue warning and fault outputs.

Here is an example of how an Output Line Planning Sheet could be filled out:

Table 3.A
Example Color CVIM Output Line Planning Sheet
Output Line Functions and Tool Assignments

Line No.	Output Line Function	Gage				Window				Reference Tool		Light Probe		
		No.	Rng.	No.	Rng.	No.	Rng.	No.	Rng.	Line	Win.	Red	Green	Blue
1	Results	1	W	2	W	1	W	2	W					
″	″	3	W	4	W									
2	Results	1	F	2	F	1	F	2	F					
″	″	3	F	4	F									
3	Results									1	1			
4	Results											W	W	W
5	Results											F	F	F
6	Results	1	W	1	F									
″	″	2	W	2	F									
7	Strobe													
8	Trig. Nak													
9	Master Fault													
10	Data Valid													
11	Module Busy													
12	Not Used													
13	Not Used													
14	Not Used													

Using the Output Line Planning Sheet (cont'd)

The entries for the output lines have the following meanings:

- **Output Line 1:** The Results function is assigned to line 1. The Warning Range results (W) for gages 1–4 and windows 1 and 2 are assigned to output line 1.
- **Output Line 2:** The Results function is assigned to line 2. The Fault Range results (F) for gages 1–4 and windows 1 and 2 are assigned to output line 2.
- **Output Line 3:** The Results function is assigned to line 3. The "pass/fail" results for reference line 1 and reference window 1 are assigned to line 3.
- **Output Line 4:** The Results function is assigned to line 4. The Warning Range result from the red, blue and green light probe is assigned to line 4.
- **Output Line 5:** The Results function is assigned to line 5. The Fault Range result from the red, blue and green light probe is assigned to line 5.
- **Output Line 6:** The Results function is assigned to line 6. The Warning *and* Fault Range results for gages 1 and 2 are assigned to line 6.
- **Output Line 7:** The Strobe function is assigned to line 7.
- **Output Line 8:** The Trigger NAK function is assigned to line 8.
- **Output Line 9:** The Master Fault function is assigned to line 9.
- **Output Line 10:** The Data Valid function is assigned to line 10.
- **Output Line 11:** The Module Busy function is assigned to line 11.
- **Output Lines 12–14:** These lines are not used.

Note: Output lines 1–6 are assigned the Results function. These lines will carry "pass/fail" results from the analysis tools to your production equipment. Lines 7– 11 are assigned other functions. Lines 12–14 are not used.

Here is a brief explanation of the signal functions that you can assign to the output lines:

- **Module Busy:** This signal goes *high* when the Color CVIM system enters the configuration mode and during a configuration download operation. Module Busy goes *low* when the system enters the run mode (whether or not triggers are present).

  You can assign the Module Busy function to only *one* output line.

Using the Output Line Planning Sheet (cont'd)

Note: All of the remaining signal functions (except Strobe, Module Busy, and Trigger NAKs) can be configured to produce a *pulse* whose duration depends on the number of milliseconds that you assign to the Duration parameter.

- **Results:** This signal occurs when the results of a tool inspection exceed the warning and/or fault limits. (The tool must be assigned to an output line that has already been assigned the Results function.)

  You can assign the Results signal function to *any unassigned* output line.

As noted above, the Results signal function must be assigned to an output line *before* any tool can be assigned to that line. Thus, if you wanted inspection results from Ref. Line # 2 to be assigned to output line #10, you would *first* have to assign the Results signal function to output line #10.

Note: You can assign the inspection results from *any* tool to an output line to which you have already assigned the Results signal function.

- **Data Valid:** This signal occurs when the Color CVIM system has *completed* an inspection. Data Valid *signals* (the "data") are stable on all output lines assigned to the Results signal function. Data Valid goes low after the next trigger or the selected Duration time.

  Note: Data Valid does *not* indicate whether an inspection has passed or failed. That is the task of the output lines assigned to the Results signal function.

  You can assign the Data Valid function to only *one* output line.

- **Trigger NAK:** This signal occurs when the Color CVIM system receives a trigger input signal, but cannot process that trigger. The signal goes low upon the next "accepted trigger".

  You can assign the Trigger NAK function to only *one* output line.

- **Master Fault:** This signal occurs when *any (one or more)* analysis tools in the Color CVIM system detects a Fail condition.

  You can assign the Master Fault function to only *one* output line.

Using the Output Line Planning Sheet (cont'd)

- **Strobe:** This signal is used to trigger the strobe flash unit (if used). The signal occurs within 1 ms after the Color CVIM system receives a trigger input signal.

You can assign the Strobe function to only *one* output.

- **Duration (n)ms:** From 1msec to 2000msec. This value determines the pulse duration, in milliseconds (and with an accuracy of ±5 milliseconds), of *all* pulse–type signals. A setting of zero means the signal will remain in its present state until updated by a subsequent inspection.

Note: The output duration may vary if subsequent inspections occur before the specified output duration has elapsed.

In *your* application, the function and tool assignment(s) for each output line will of course depend on the specific requirements of your production equipment.

You will find a full–page, blank copy of the planning sheet on the last page of this chapter. We suggest that you do not mark that page, but use it instead as a copy master, and use the copies to prepare your output line plans.

Keep in mind that a *completed* planning sheet can serve also as a *record* of your output line usage. You may find it desirable to store your filled–out planning sheets in a file folder or loose leaf binder.

Using Output Signal Timing Data

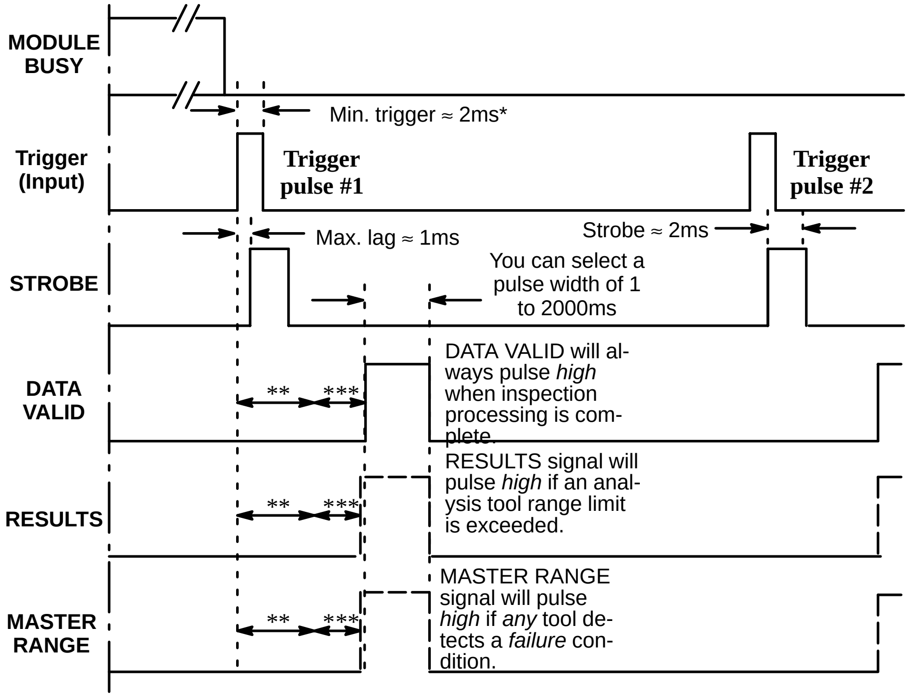
To make proper use of the signal data available to the output lines, you must first understand the timing relationships that exist between the trigger *input* signal (which *starts* each inspection cycle) and the *output* signals.

Knowing these signal timing relationships enables you to accurately *synchronize* the inspection cycles with your production equipment.

Timing charts (Figures 3.2, 3.3, and 3.4) show the timing relationships in various circumstances.

Figure 3.2 shows the relationship between the trigger leading edge and the Strobe, Data Valid, and Results signals, where the last three appear as *pulses* whose duration *you* determine during configuration.

Figure 3.2
Timing Diagram — Pulsed I/O

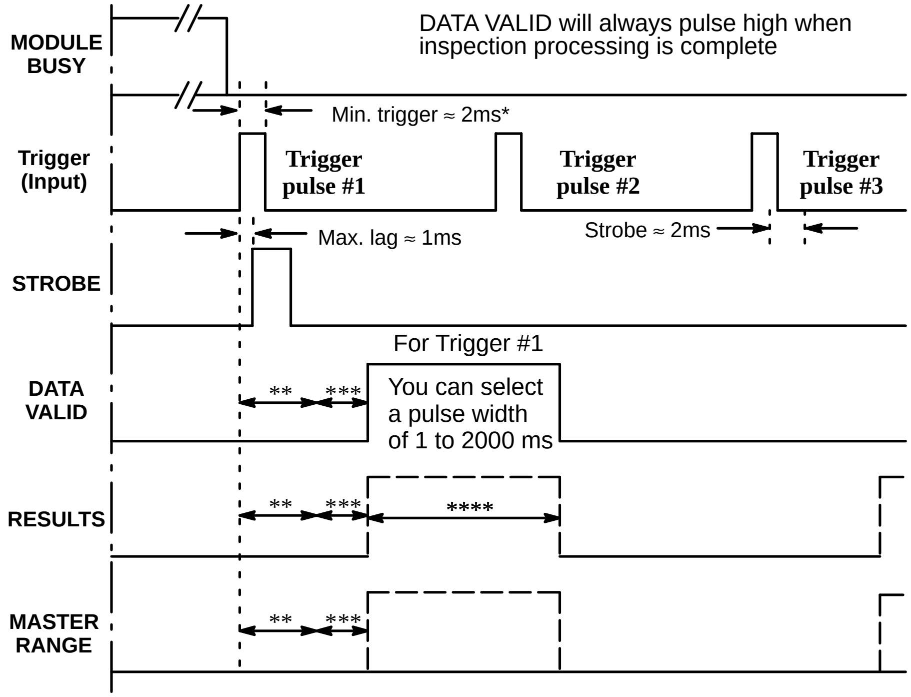

MODULE BUSY
Min. trigger ≈ 2ms*
Trigger (Input)
Trigger pulse #1
Trigger pulse #2
Max. lag ≈ 1ms
Strobe ≈ 2ms
STROBE
You can select a pulse width of 1 to 2000ms
DATA VALID
**
***
DATA VALID will always pulse *high* when inspection processing is complete.
RESULTS
RESULTS signal will pulse *high* if an analysis tool range limit is exceeded.
MASTER RANGE
MASTER RANGE signal will pulse *high* if *any* tool detects a *failure* condition.

* As seen by the Color CVIM when the trigger is directly connected through the JMB board, and not provided by a host device.
** Minimum acquisition time: 17ms for 256x256 and 512x256 Res; 34 ms for 512x512 res.
*** Analysis time (variable).

Using Output Signal Timing Data (cont'd)

In Figure 3.3, trigger pulse #2 occurs before the Color CVIM module has finished the pulsed output duration of the outputs from trigger #1.

Figure 3.3
Timing Diagram — Trigger #2 During Data Valid, Pulsed I/O



* As seen by the Color CVIM when the trigger is directly connected through the JMB board, and not provided by a host device.

** Minimum acquisition time: 17ms for 256x256 and 512x256 Res; 34 ms for 512x512 res.

*** Analysis time (variable).

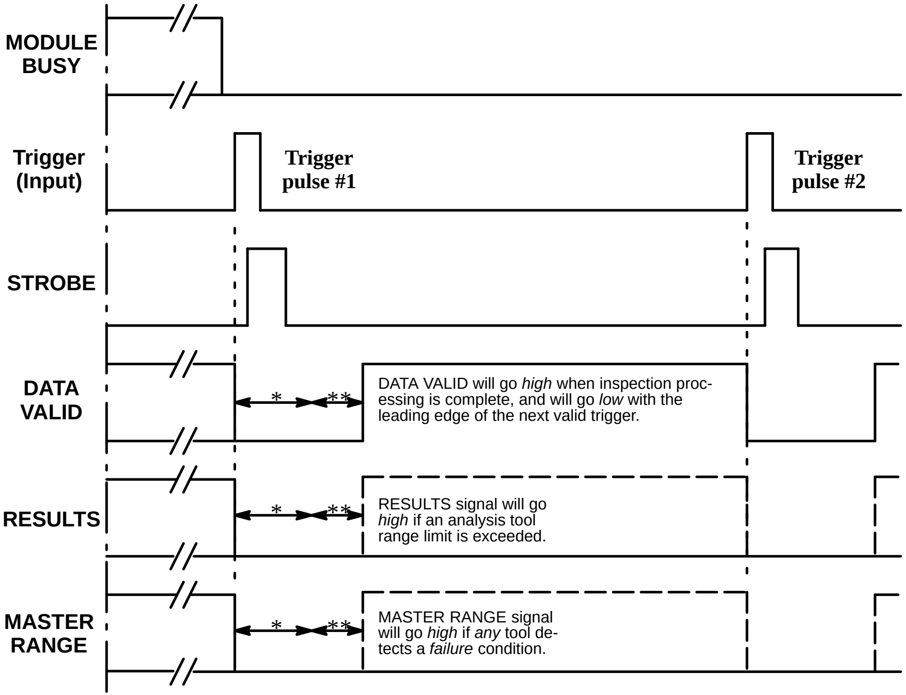
****RESULTS will pulse high if an analysis tool range is exceeded.

Using Output Signal Timing Data (cont'd)

Whenever these signals go *high*, they will go *low* again at the *end* of the specified pulse duration (1 to 2000ms).

In Figure 3.4, the Data Valid, and Results signals appear as *changes in signal levels.* This will occur if, during system configuration, you select a pulse "duration" of 0 (zero) milliseconds. Data Valid will *stay* high until the leading edge of the next valid trigger signal (Trigger Pulse #2). Results stay in their current state until the leading edge of the next Trigger pulse, then change depending upon the results.

Figure 3.4
Timing Diagram — Non–Pulsed I/O



DATA VALID will go *high* when inspection processing is complete, and will go *low* with the leading edge of the next valid trigger.

RESULTS signal will go *high* if an analysis tool range limit is exceeded.

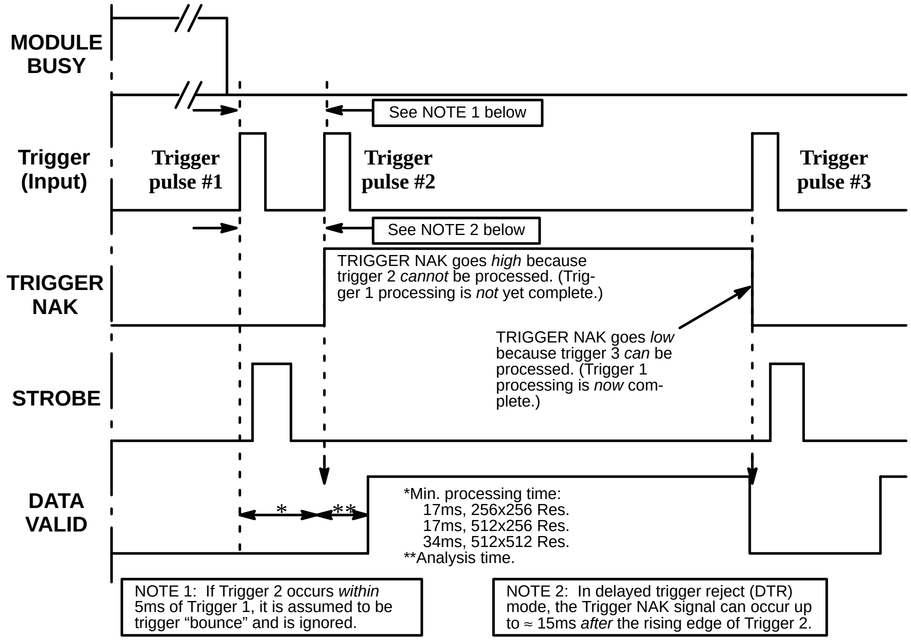
MASTER RANGE signal will go *high* if *any* tool detects a *failure* condition.

*Minimum acquisition time: 17ms for 256x256 Res.; 34ms for 512x512 Res.
**Analysis time.

Using Output Signal Timing Data (cont'd)

In Figure 3.5, trigger pulse #2 occurs *before* the Color CVIM system is finished processing the inspection cycle started by trigger pulse #1. This causes the Trigger NAK signal to go *high.* Trigger NAK will *stay* high until leading edge of the next *valid* trigger pulse (trigger pulse #3).

Figure 3.5
Timing Diagram– Missed Trigger


MODULE BUSY
See NOTE 1 below
Trigger (Input)
Trigger pulse #1
Trigger pulse #2
Trigger pulse #3
See NOTE 2 below
TRIGGER NAK
TRIGGER NAK goes *high* because trigger 2 *cannot* be processed. (Trigger 1 processing is *not* yet complete.)
TRIGGER NAK goes *low* because trigger 3 *can* be processed. (Trigger 1 processing is *now* complete.)
STROBE
DATA VALID
*
**
*Min. processing time:
17ms, 256x256 Res.
17ms, 512x256 Res.
34ms, 512x512 Res.
**Analysis time.
NOTE 1: If Trigger 2 occurs *within* 5ms of Trigger 1, it is assumed to be trigger "bounce" and is ignored.
NOTE 2: In delayed trigger reject (DTR) mode, the Trigger NAK signal can occur up to ≈ 15ms *after* the rising edge of Trigger 2.

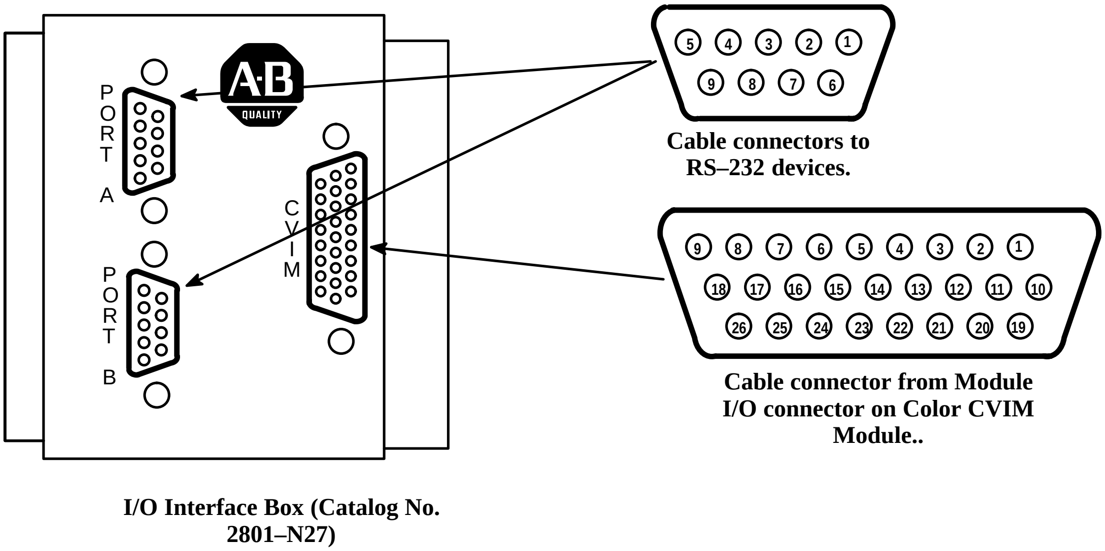
Planning Output Line Connections

This section provides diagrams of electrical connections for correctly connecting your production equipment to the Color CVIM module's discrete output and RS–232 lines.

Connections to RS–232 Ports (2801–N27 Interface Box)

Figure 3.6 shows the cable connectors and their pin numbers on the Catalog No. 2801–N27 I/O Interface Box.

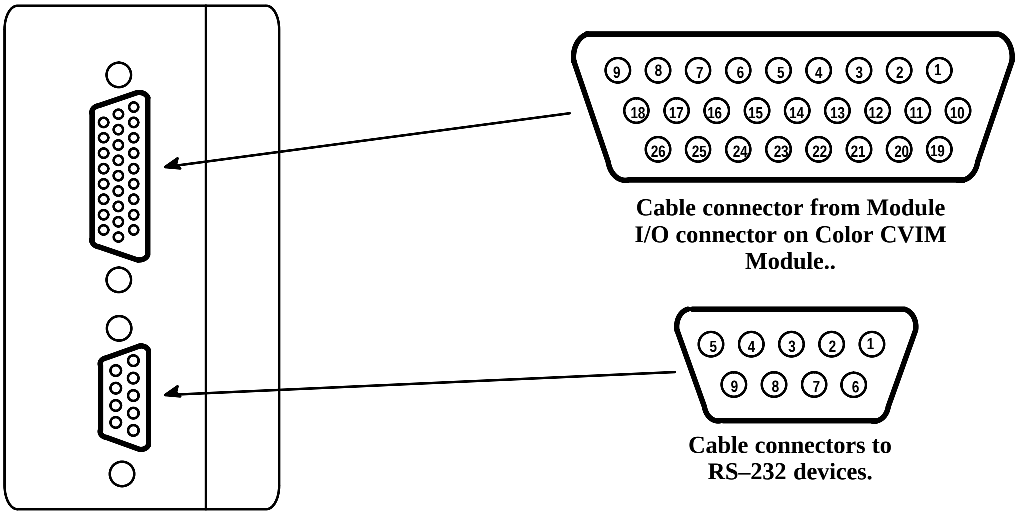
Figure 3.6
Pinouts– Catalog No. 2801–N27 I/O Interface Box


PORT A
PORT B
CVIM
5 4 3 2 1
9 8 7 6
Cable connectors to RS–232 devices.
9 8 7 6 5 4 3 2 1
18 17 16 15 14 13 12 11 10
26 25 24 23 22 21 20 19
Cable connector from Module I/O connector on Color CVIM Module..
I/O Interface Box (Catalog No. 2801–N27)

Connections to RS–232 Port (2801–N21 Interface Box)

Figure 3.7 shows the cable connectors and their pin numbers on the Catalog No. 2801–N21 I/O Interface Box.

Figure 3.7
Pinouts– Catalog No. 2801–N21 I/O Interface Box



Cable connector from Module I/O connector on Color CVIM Module..

Cable connectors to RS–232 devices.

I/O Interface Box (Catalog No. 2801–N21)

Color CVIM Module I/O Interface Box Connections

Tables 3.B through 3.E show the connector pin assignments for the Color CVIM module connected to Catalog No. 2801–N21, –N27 I/O Interface Box.

⚠ **ATTENTION:**
Making discrete I/O connections directly to the front of the Color CVIM module will result in damage to the equipment and void your warranty.

Table 3.B
Color CVIM Module I/0 Connector

Pin Number	Function	Pin Number	Function
1	Trigger Input Line #1	14	Output Line #12
2	Not Used	15	Not Used
3	Output Line #1	16	Output Line #14
4	Output Line #2	17	Reserved
5	Output Line #3	18	Reserved
6	Output Line #4	19	Ground (Power)
7	Output Line #5	20	Ground (Power)
8	Output Line #6	21	Ground (Chassis)
9	Output Line #7	22	Ground (Signal)
10	Output Line #8	23	TXD (Transmit Data: RS–232 A)
11	Output Line #9	24	TXD (Transmit Data: RS–232B)
12	Output Line #10	25	RXD (Receive Data: RS–232 A)
13	Output Line #11	26	RXD (Receive Data: RS–232 B)

Table 3.C
I/O Interface Box (Catalog No. 2801–N21):
RS–232 Connector with Series A Color CVIM Module

Pin Number	Function	Pin Number	Function
1	No Connection	6	No Connection
2	RXD (Receive Data: RS–232A)	7	TXD (Transmit Data: RS–232 B)
3	TXD (Transmit Data: RS–232A)	8	RXD (Receive Data: RS–232 B)
4	Ground (Chassis)	9	No Connection
5	Ground (Signal)		

Color CVIM Module I/O Interface Box Connections (cont'd)

Table 3.D
I/O Interface Box (Catalog No. 2801–N27):
RS–232 Port A Connector with Color CVIM Module

Pin Number	Function	Pin Number	Function
1	No Connection	6	No Connection
2	RXD (Receive Data: RS–232A)	7	+ 5V DC*
3	TXD (Transmit Data: RS–232A)	8	No Connection
4	+ 5V DC*	9	No Connection
5	Ground (Signal)		

*Not to be used to power external devices.

Table 3.E
I/O Interface Box (Catalog No. 2801–N27):
RS–232 Port B Connector with Color CVIM Module

Pin Number	Function	Pin Number	Function
1	No Connection	6	No Connection
2	RXD (Receive Data: RS–232B)	7	+ 10V DC*
3	TXD (Transmit Data: RS–232B)	8	No Connection
4	+ 10V DC*	9	No Connection
5	Ground (Signal)		

*Not to be used to power external devices.

Connections to 2801–JMB Interface

The 2801–JMB interface board is designed for direct edge connection to the I/O Interface Box, Catalog Nos. 2801–N21, –N27.

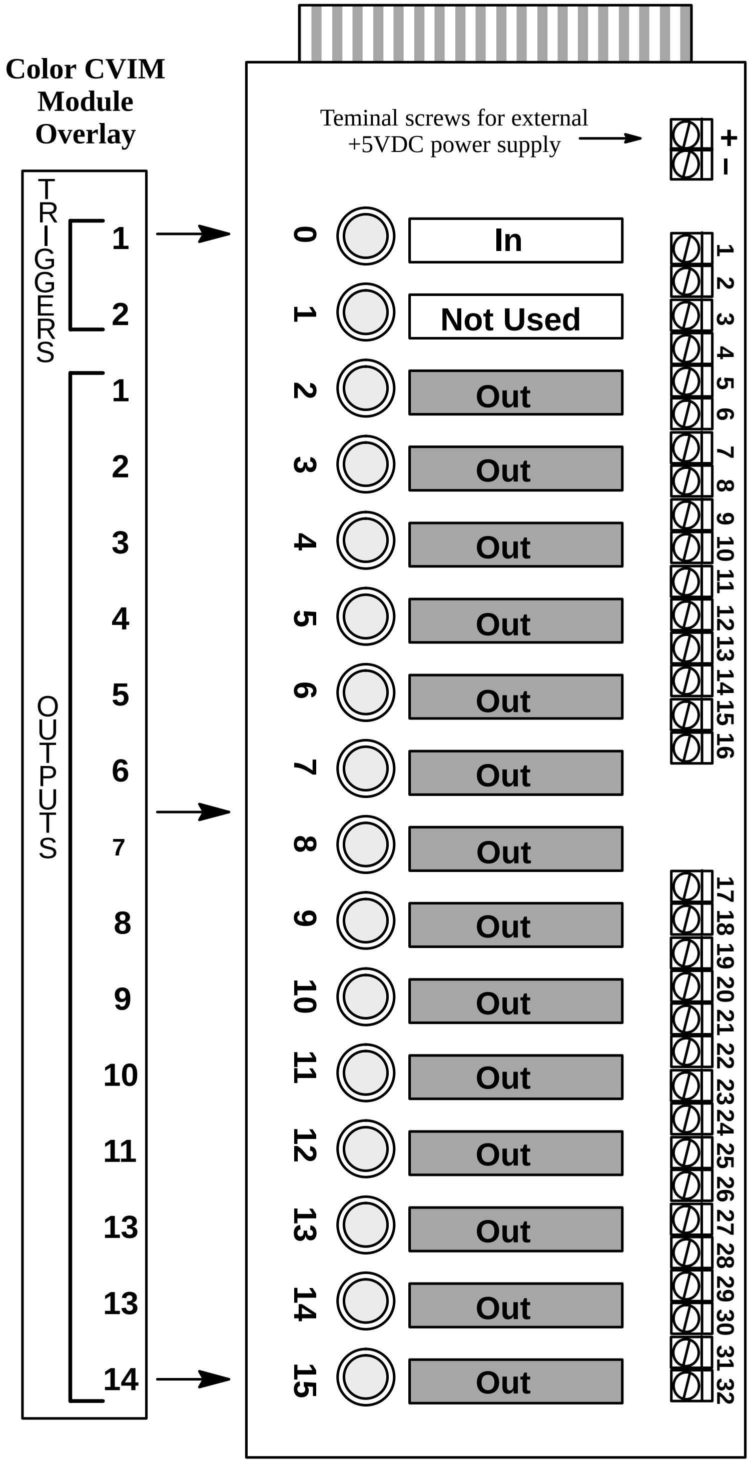
If you intend to use the 2801–JMB board and the I/O Interface Box, you will need to know the relationship between the discrete I/O line numbers and the LED numbers, the optic–isolator type, and the terminal block screws numbers on the 2801–JMB board. These are shown in the figure and table that follows.

ATTENTION:
To power the JMB logic components, you must connect an *external* +5VDC power supply to the (+) and (–) terminal screws shown on the board layout that follows.

Connections to 2801–JMB Interface (cont'd)

Figure 3.8 shows the layout of the 2801–JMB interface board and the adhesive–backed overlay.

Figure 3.8
Local I/O Board (Catalog No. 2801–JMB).



Connections to 2801–JMB Interface (cont'd)

Table 3.F shows the relationship between the I/O line and optic–isolator numbers shown in Figure 3.8.

Table 3.F
Color CVIM I/O Output Numbering

Discrete I/O Line Number		LED and I/O Module Number	Terminal Screw and Polarity	
Input	Output		+	–
1		0	1	2
2*		NA	3/NC** this input is not used by Color CVIM.	4/NC
	1	2	5	6
	2	3	7	8
	3	4	9	10
	4	5	11	12
	5	6	13	14
	6	7	15	16
	7	8	17	18
	8	9	19	20
	9	10	21	22
	10	11	23	24
	11	12	25	26
	12	13	27	28
	13	14	29	30
	14	15	31	32

Note: A self–adhesive decal is provided with the 2801–JMB Local I/O board. This decal identifies the I/O lines. Use this chart if the decal is not in place.

* This input is not used by Color CVIM.

** Not Connected

Connections to 2801–JMB Interface (cont'd)

Table 3.G
Color CVIM Output Line Planning Sheet
Output Line Functions and Assignments

Line No.	Output Line Function	Gage				Window				Reference Tool		Light Probe		
		No.	Rng.	No.	Rng.	No.	Rng.	No.	Rng.	Line	Win.	Red	Green	Blue

Chapter 4

Using the Remote I/O Link

Chapter Objectives

In this chapter we provide:

- Basic description of Remote I/O communications.
- Connection diagrams.
- Description of Color CVIM module setup requirements.
- Three example PLC programs for accessing Color CVIM module data.
- An example 6008–SI program.

Remote I/O Communications

As stated earlier, the Remote I/O port is located on the front of the Color CVIM module and is labeled RIO. This port allows the Color CVIM module to become a link in an Allen–Bradley Remote I/O network which can be up to 10,000 feet long. Data on the network can be transmitted at baud rates as high as 230K.

Maximum Link Length (Feet)	Baud Rate
10,000	57.6K
5,000	115.2K
2,500	230.4K

Use twin–axial cable (Catalog No. 1770–CD) to connect the Color CVIM module to other devices. This cable connects to the Remote I/O port (labeled RIO) and the next device on the network. Refer to Figures 4.1 through 4.8 for connection diagrams.

Remote I/O Communications (cont'd)

Figure 4.1
PLC–5 to Color CVIM Module– Remote I/O Link

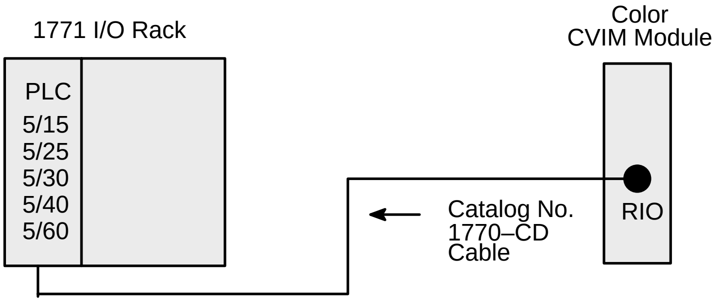


Figure 4.2
6008 SI IBM PC/AT Scanner to Color CVIM Module– Remote I/O Link

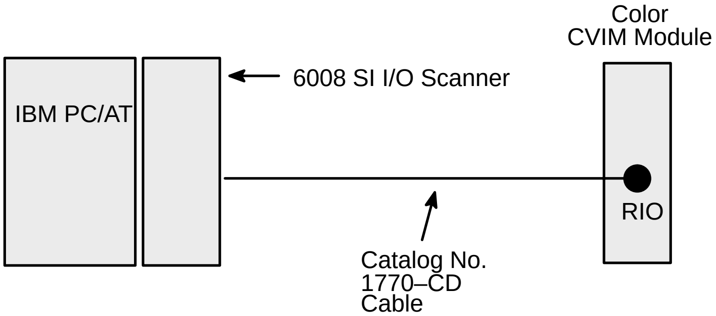
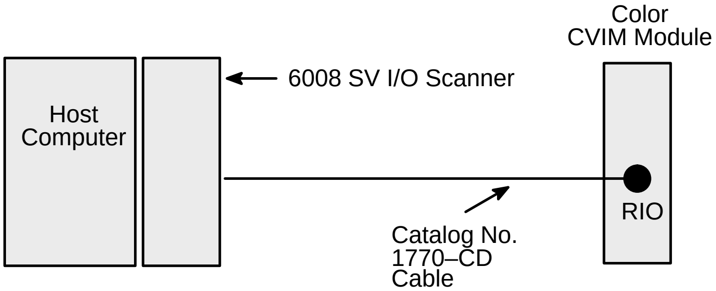


Figure 4.3
6008 SV VME Scanner to Color CVIM Module– Remote I/O Link



Remote I/O Communications (cont'd)

Figure 4.4
6008 SQH1/2 Q–BUS Scanner to Color CVIM Module–Remote I/O Link

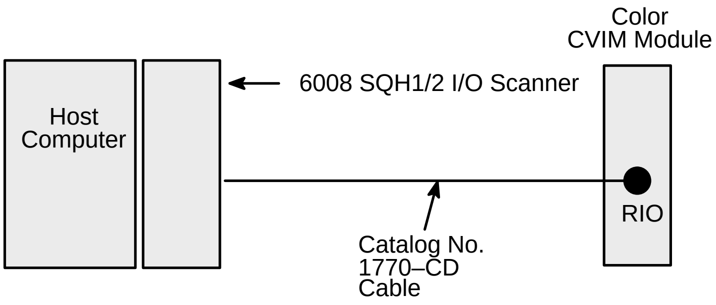


Figure 4.5
Mini PLC–2 to Color CVIM Module– Remote I/O Link

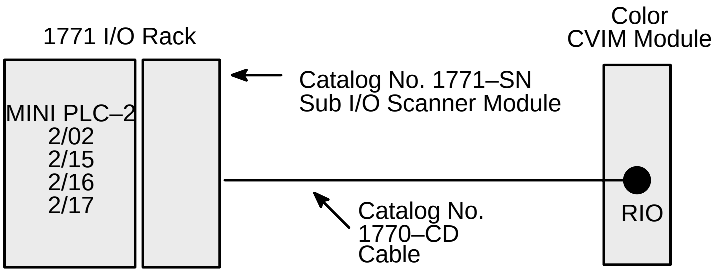
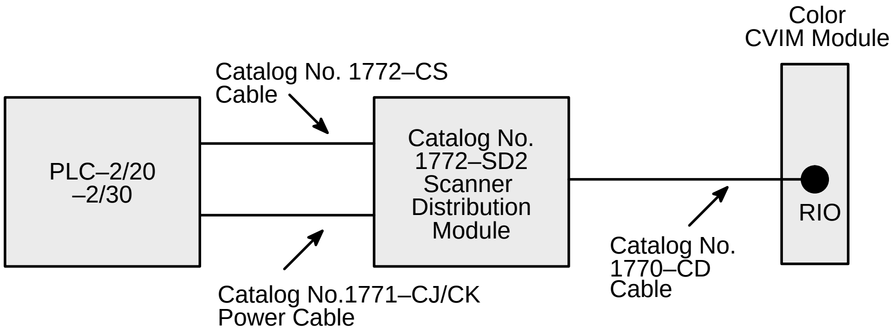


Figure 4.6
PLC–2 to Color CVIM Module– Remote I/O Link



Remote I/O Communications (cont'd)

Figure 4.7
PLC–3 to Color CVIM Module–Remote I/O Link

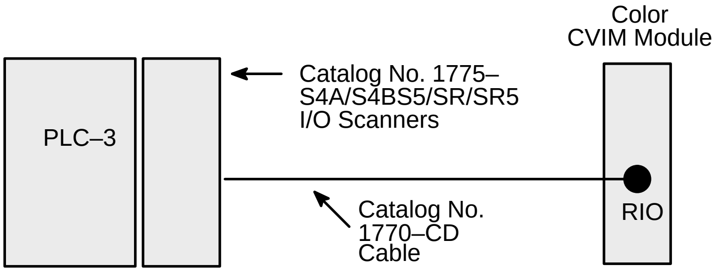
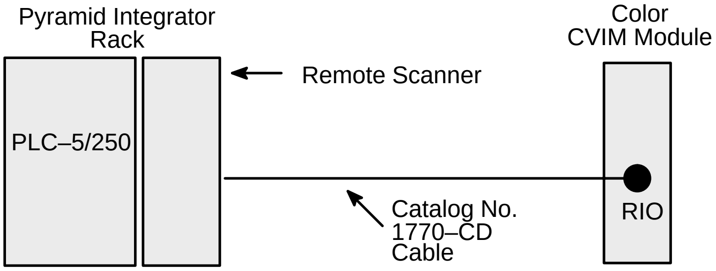
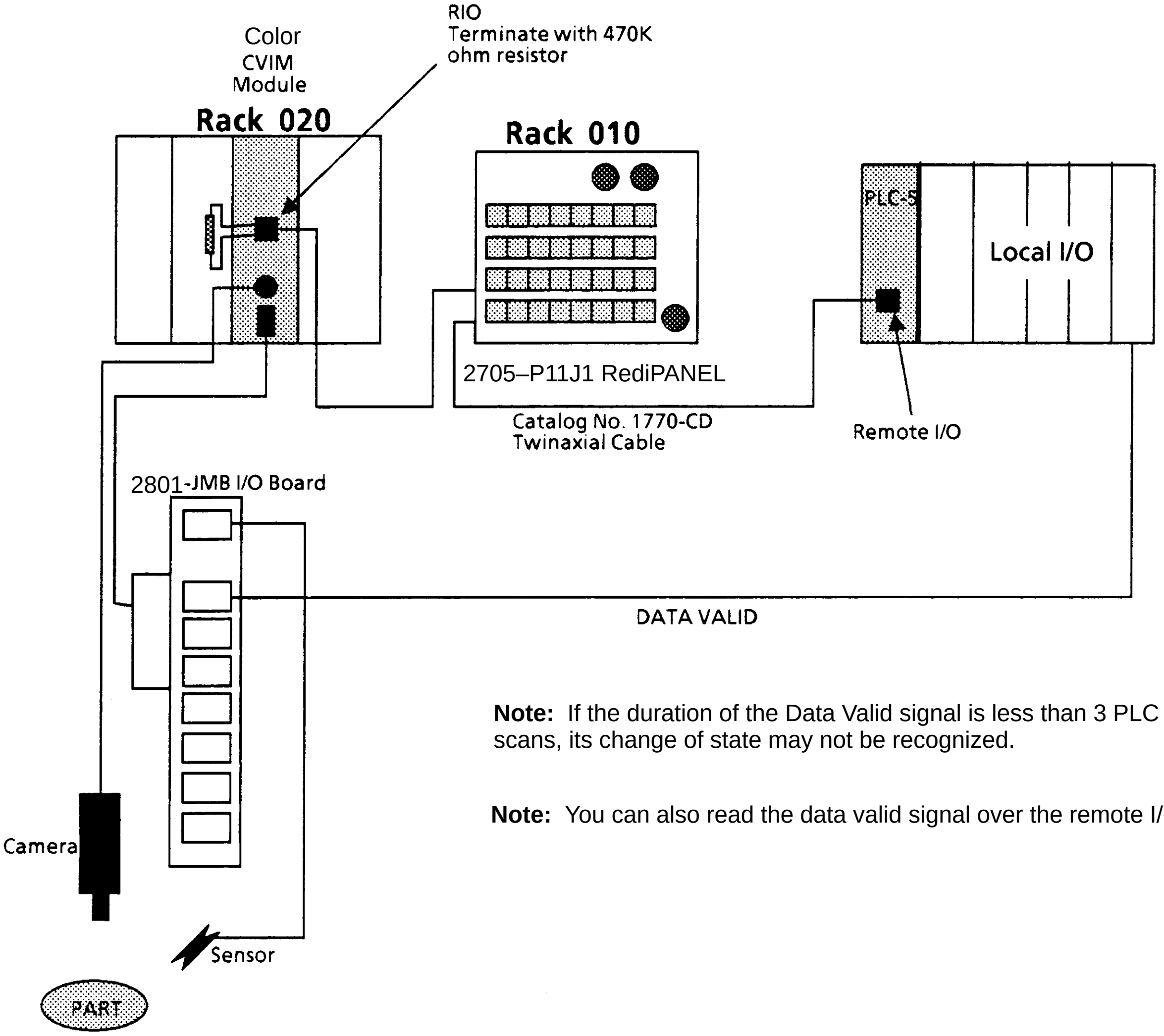


Figure 4.8
PLC–5/250 to Color CVIM Module– Remote I/O Link



Remote I/O Communications (cont'd)

Figure 4.9
Typical Hardware Layout for Remote I/O



Note: If the duration of the Data Valid signal is less than 3 PLC scans, its change of state may not be recognized.

Note: You can also read the data valid signal over the remote I/O link.

Remote I/O Communications (cont'd)

When installed on a Remote I/O network, the Color CVIM module acts as a slave device. Another device such as a PLC or computer will act as a host device. This means that the Color CVIM module will not initiate the sending of any data until a request is made by the host. To a host device, the Color CVIM module will appear simultaneously as both a full I/O rack on the network (256 input bits and 128 output bits) and as an intelligent module with block transfer capability in group 0, slot 0 in the same rack. Refer to Appendix B for a description of discrete bit data.

Note: If the Color CVIM module is the last node on a network, you must terminate the communication line (refer to Figure 4.9 for an example).

What Functions can be Performed over the Remote I/O Network?

Any Color CVIM communications port can be used by an external device to read inspection result information. (Refer to Appendices B and C for more information.) In addition, any device selected as a system (SYS) host can:

Change run–time display menus.

Enable/Disable local I/O board.

Force local I/O On or Off

A host device selected as a configuration host can also:

Upload or download Color CVIM module configurations for inspections. Refer to Appendix D.

Issue Configuration Read/Write commands between the following Color CVIM module memory locations:

Obtaining Inspection Result Information

You can obtain inspection result information for each of the inspection tools over the Remote I/O link. There are two levels of access to this information:

Discrete Bits. These bits indicate pass/fail/warning data. Note that there are 256 input bits from the Color CVIM to the PLC. That input consists of two pages of 128 bits each. The first 128 bits refer to the first half of the toolset, and the second 128 bits refer to the second half of the toolset.

Result Data Words. These words contain actual inspection result data such as measured lengths, number of black pixels, etc.

Note: Refer to Appendix B for a description of the discrete bit results and Appendix C for a description of numerical results data blocks.

Color CVIM Module Configuration Instructions

If you are using the Remote I/O link to communicate with a PLC–2, –3, or –5 (or PLC–5/250 in another rack), you must configure the Color CVIM module as follows:

Select the Remote I/O port for communications:

Note: This step is not required if you are only reading results.

1. Select the setup menu <Setup>.
2. Select the environment menu <Envir/Cam>.
3. Select the system menu <System>.
4. Select a Host menu <CFG Host> or <SYS Host>.
5. Select remote I/O option <Remote I/O>.

Note: Unless a separate configuration host is being used, set both the CFG Host & SYS Host for Remote I/O.

Configure Color CVIM module I/O parameters:

6. Select the I/O menu <I/O>.
7. Select <1771 Remote I/O> option.
8. Enable the Remote I/O port by selecting <Enabled>.
9. Select the rack address (octal) using the keypad.
10. Select the baud rate <57.6Kbaud> or other options.

Select the Color CVIM module trigger source:

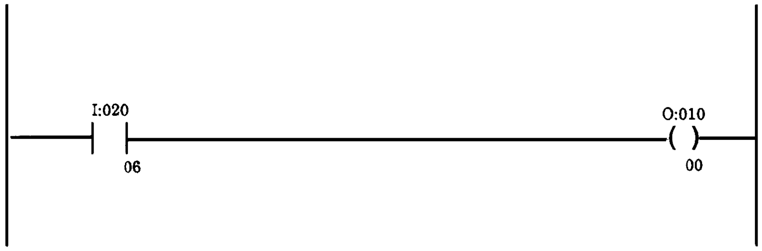
11. Select the trigger source menu <Trigger Source>.
12. Select either <I/O>, <Hosted>, or <Internally Triggered> trigger sources.

Note: The example connection diagram shown on Figure 4.9 shows a trigger using the local I/O board.

Accessing Discrete Bit Information

A PLC can directly access discrete bit information using a simple ladder program. For example:

You can use the following rung to examine the data valid bit and energize an output if the data is valid. Refer to Chapter 3 for a description of the local I/O. This example assumes that the Color CVIM module is in Rack 02 and the output device is in Rack 01.



Although the same basic information is provided in Appendix B, Tables 4.A and 4.B illustrate the word and bit locations of the discrete bits that can be read or manipulated using simple ladder programs. We have organized the data so that it is formatted similar to a PLC setup screen. Table 4.A shows the first part of discrete bits results (Color CVIM module to PLC). Table 4.B shows the second part. You can tell the difference by examining word 0, bit 5, which has a value of 0 for the first part and 1 for the second part. Table 4.C shows the Color CVIM remote outputs (PLC to Color CVIM module).

Important Note: To read results data, you must set one of the following bits (assuming Color CVIM module is rack 02):

O:22/00 (Post First Part of Results to Remote I/O)

O:22/01 (Post Second Part of Results to Remote I/O)

Note to PLC–2 Users:

When you use any PLC–2 family processor with the Color CVIM module, you should understand the operation of the PLC Block Transfer Done bits for Read and Write instructions. PLC–2 family processors use the input image table for these bits, all other PLCs can specify integer files for this function. This means that a PLC–2 user must use proper programming techniques to avoid confusion between the following bits:

Color CVIM module discrete I/O input word 0, bit 6 (data valid) and bit 7 (not used).

PLC–2 family input image table word 0, bit 6 (BTW done bit) and bit 7 (BTR done bit).

Remember that word 27 is a reserved word in the PLC–2 data table, and should not be used for any other purpose.

Accessing Discrete Bit Information (cont'd)

Table 4.A
Color CVIM Module Remote I/O Inputs (Color CVIM Module to PLC) if Color CVIM Module is Rack 02 (First Discrete Bits Results)

07	06	05	04	03	02	01	00	BIT
17	16	15	14	13	12	11	10	WORD
(Not used)	1=Data Valid	0=First Bits Results	1 = Trigger Missed	1 = Module Busy	1 = PLC is Master	1 = Config. Error	(Not Used)	20
1 = Master Fault	1 = Light Probe Failed	1 = Reference Window 3 Failed	1 = Reference Window 2 Failed	1 = Reference Window 1 Failed	1 = Reference Line 3 Failed	1 = Reference Line 2 Failed	1 = Reference Line 1 Failed	
1 = Window 4 Fault	1 = Window 4 Warning	1 = Window 3 Fault	1 = Window 3 Warning	1 = Window 2 Fault	1 = Window 2 Warning	1 = Window 1 Fault	1 = Window 1 Warning	21
1 = Window 8 Fault	1 = Window 8 Warning	1 = Window 7 Fault	1 = Window 7 Warning	1 = Window 6 Fault	1 = Window 6 Warning	1 = Window 5 Fault	1 = Window 5 Warning	
1 = Window 12 Fault	1 = Window 12 Warning	1 = Window 11 Fault	1 = Window 11 Warning	1 = Window 10 Fault	1 = Window 10 Warning	1 = Window 9 Fault	1 = Window 9 Warning	22
1 = Window 16 Fault	1 = Window 16 Warning	1 = Window 15 Fault	1 = Window 15 Warning	1 = Warning 14 Fault	1 = Window 14 Warning	1 = Window 13 Fault	1 = Window 13 Warning	
1 = Window 20 Fault	1 = Window 20 Warning	1 = Window 19 Fault	1 = Window 19 Warning	1 = Window 18 Fault	1 = Window 18 Warning	1 = Window 17 Fault	1 = Window 17 Warning	23
1 = Window 24 Fault	1 = Window 24 Warning	1 = Window 23 Fault	1 = Window 23 Warning	1 = Window 22 Fault	1 = Window 22 Warning	1 = Window 21 Fault	1 = Window 21 Warning	
1 = Gage 4 Fault	1 = Gage 4 Warning	1 = Gage 3 Fault	1 = Gage 3 Warning	1 = Gage 2 Fault	1 = Gage 2 Warning	1 = Gage 1 Fault	1 = Gage 1 Warning	24
1 = Gage 8 Fault	1 = Gage 8 Warning	1 = Gage 7 Fault	1 = Gage 7 Warning	1 = Gage 6 Fault	1 = Gage 6 Warning	1 = Gage 5 Fault	1 = Gage 5 Warning	
1 = Gage 12 Fault	1 = Gage 12 Warning	1 = Gage 11 Fault	1 = Gage 11 Warning	1 = Gage 10 Fault	1 = Gage 10 Warning	1 = Gage 9 Fault	1 = Gage 9 Warning	25
1 = Gage 16 Fault	1 = Gage 16 Warning	1 = Gage 15 Fault	1 = Gage 15 Warning	1 = Gage 14 Fault	1 = Gage 14 Warning	1 = Gage 13 Fault	1 = Gage 13 Warning	
1 = Gage 20 Fault	1 = Gage 20 Warning	1 = Gage 19 Fault	1 = Gage 19 Warning	1 = Gage 18 Fault	1 = Gage 18 Warning	1 = Gage 17 Fault	1 = Gage 17 Warning	26
1 = Gage 24 Fault	1 = Gage 24 Warning	1 = Gage 23 Fault	1 = Gage 23 Warning	1 = Gage 22 Fault	1 = Gage 22 Warning	1 = Gage 21 Fault	1 = Gage 21 Warning	
1 = Gage 28 Fault	1 = Gage 28 Warning	1 = Gage 27 Fault	1 = Gage 27 Warning	1 = Gage 26 Fault	1 = Gage 26 Warning	1 = Gage 25 Fault	1 = Gage 25 Warning	27
1 = Gage 32 Fault	1 = Gage 32 Warning	1 = Gage 31 Fault	1 = Gage 31 Warning	1 = Gage 30 Fault	1 = Gage 30 Warning	1 = Gage 29 Fault	1 = Gage 29 Warning	

Accessing Discrete Bit Information (cont'd)

Table 4.B
Color CVIM Module Remote I/O Inputs (Color CVIM Module to PLC) if Color CVIM Module is Rack 02 (Second Discrete Bits Results)

BIT

07	06	05	04	03	02	01	00	WORD
17	16	15	14	13	12	11	10	
(Not used)	1=Data Valid	1=Second Bits Results	1 = Trigger Missed	1 = Module Busy	1 = PLC is Master	1 = Config. Error	(Not Used)	20
1 = Master Fault	1 = Light Probe Failed	1 = Reference Window 3 Failed	1 = Reference Window 2 Failed	1 = Reference Window 1 Failed	1 = Reference Line 3 Failed	1 = Reference Line 2 Failed	1 = Reference Line 1 Failed	
1 = Window 28 Fault	1 = Window 28 Warning	1 = Window 27 Fault	1 = Window 27 Warning	1 = Window 26 Fault	1 = Window 26 Warning	1 = Window 25 Fault	1 = Window 25 Warning	21
1 = Window 32 Fault	1 = Window 32 Warning	1 = Window 31 Fault	1 = Window 31 Warning	1 = Window 30 Fault	1 = Window 30 Warning	1 = Window 29 Fault	1 = Window 29 Warning	
1 = Window 36 Fault	1 = Window 36 Warning	1 = Window 35 Fault	1 = Window 35 Warning	1 = Window 34 Fault	1 = Window 34 Warning	1 = Window 33 Fault	1 = Window 33 Warning	22
1 = Window 40 Fault	1 = Window 40 Warning	1 = Window 39 Fault	1 = Window 39 Warning	1 = Warning 38 Fault	1 = Window 38 Warning	1 = Window 37 Fault	1 = Window 37 Warning	
1 = Window 44 Fault	1 = Window 44 Warning	1 = Window 43 Fault	1 = Window 43 Warning	1 = Window 42 Fault	1 = Window 42 Warning	1 = Window 41 Fault	1 = Window 41 Warning	23
1 = Window 48 Fault	1 = Window 48 Warning	1 = Window 47 Fault	1 = Window 47 Warning	1 = Window 46 Fault	1 = Window 46 Warning	1 = Window 45 Fault	1 = Window 45 Warning	
1 = Gage 36 Fault	1 = Gage 36 Warning	1 = Gage 35 Fault	1 = Gage 35 Warning	1 = Gage 34 Fault	1 = Gage 34 Warning	1 = Gage 33 Fault	1 = Gage 33 Warning	24
1 = Gage 40 Fault	1 = Gage 40 Warning	1 = Gage 39 Fault	1 = Gage 39 Warning	1 = Gage 38 Fault	1 = Gage 38 Warning	1 = Gage 37 Fault	1 = Gage 37 Warning	
1 = Gage 44 Fault	1 = Gage 44 Warning	1 = Gage 43 Fault	1 = Gage 43 Warning	1 = Gage 42 Fault	1 = Gage 42 Warning	1 = Gage 41 Fault	1 = Gage 41 Warning	25
1 = Gage 48 Fault	1 = Gage 48 Warning	1 = Gage 47 Fault	1 = Gage 47 Warning	1 = Gage 46 Fault	1 = Gage 46 Warning	1 = Gage 45 Fault	1 = Gage 45 Warning	
1 = Gage 52 Fault	1 = Gage 52 Warning	1 = Gage 51 Fault	1 = Gage 51 Warning	1 = Gage 50 Fault	1 = Gage 50 Warning	1 = Gage 49 Fault	1 = Gage 49 Warning	26
1 = Gage 56 Fault	1 = Gage 56 Warning	1 = Gage 55 Fault	1 = Gage 55 Warning	1 = Gage 54 Fault	1 = Gage 54 Warning	1 = Gage 53 Fault	1 = Gage 53 Warning	
1 = Gage 60 Fault	1 = Gage 60 Warning	1 = Gage 59 Fault	1 = Gage 59 Warning	1 = Gage 58 Fault	1 = Gage 58 Warning	1 = Gage 57 Fault	1 = Gage 57 Warning	27
1 = Gage 64 Fault	1 = Gage 64 Warning	1 = Gage 63 Fault	1 = Gage 63 Warning	1 = Gage 62 Fault	1 = Gage 62 Warning	1 = Gage 61 Fault	1 = Gage 61 Warning	

Accessing Discrete Bit Information (cont'd)

Table 4.C
Color CVIM Module Remote I/O Outputs (PLC to Color CVIM Module) if Color CVIM Module is Rack 02

BIT

07 17	06 16	05 15	04 14	03 13	02 12	01 11	00 10	WORD
(Reserved)***	(Reserved)***	(Not Used)	(Not Used)	(Not Used)	(Not Used)	(Not Used)	(Not Used)	20
(Not Used)	1 = Config Transfer	1 = I/O Request	1 = Light pen Request	(Not Used)	1 = Trigger	1 = Unlock Setup	1 = Lock Setup	
(Not Used)	1 = Display Stat 2 Page	1 = Display Stat 1 Page	1 = Display Results Page	1 = Display I/O Page	1 = Display All Tools	1 = Display Failed Tools	1 = Display Image Only	21
(Not Used)	(Not Used)	(Not Used)	Halt on Reject	1 = Freeze Next Image	1 = Freeze All Rejects	1 = Freeze First Reject	Go on reject	
1 = Enable JMB Forces	1 = Disable JMB Forces	1 = Enable JMB Outputs	1 = Disable JMB Outputs	(Not Used)	(Not Used)	1 = Post 2nd Half Results to Remote I/O	1=Post 1st Half Results to Remote I/O	22
1 = Credit Card Config. (8's bit)****	1 = Credit Card Config. (4's bit)****	1 = Credit Card Config. (2's bit)****	1 = Credit Card Config. (1's bit)****	1 = RAM to Credit Card	1 = Credit Card to RAM	1 = RAM to EEPROM	1 = EEPROM to RAM	
Toolset Active (Normally Not Used *****)	Toolset Active (Normally Not Used *****)	Toolset Active (Normally Not Used *****)	Toolset Active (Normally Not Used *****)	1 = Last Block (write Only)	Block Transfer Type *	Block Transfer Type *	Block Transfer Type *	23
1 = Block Trnsfer Block No. (128's bit)	1 = Block Trnsfer Block No. (64's bit)	1 = Block Trnsfer Block No. (32's bit)	1 = Block Trnsfer Block No. (16's bit)	1 = Block Trnsfer Block No. (8's bit)	1 = Block Trnsfer Block No. (4's bit)	1 = Block Trnsfer Block No. (2's bit)	1 = Block Trnsfer Block No. (1's bit)	
1 = Force JMB Output 8 ON**	1 = Force JMB Output 7 ON**	1 = Force JMB Output 6 ON**	1 = Force JMB Output 5 ON**	1 = ForceJMB Output 4 ON**	1 = Force JMB Output 3 ON**	1=Force JMB Output 2 ON**	1=Force JMB Output 1 ON**	24
(Not Used)	(Not Used)	1 = Force JMB Output 14 ON**	1 = Force JMB Output 13 ON**	1 = Force JMB Output 12 ON**	1 = Force JMB Output 11 ON**	1 = Force JMB Output 10 ON**	1 = Force JMB Output 9 ON**	
1 = Force JMB Output 8 OFF**	1 = Force JMB Output 7 OFF**	1 = Force JMB Output 6 OFF**	1 = Force JMB Output 5 OFF**	1 = Force JMB Output 4 OFF**	1 = Force JMB Output 3 OFF**	1 = Force JMB Output 2 OFF**	1 = Force JMB Output 1 OFF**	25
(Not Used)	(Not Used)	1 = Force JMB Output 14 OFF**	1 = Force JMB Output 13 OFF**	1 = Force JMB Output 12 OFF**	1 = Force JMB Output 11 OFF**	1 = Force JMB Output 10 OFF**	1 = Force JMB Output 9 OFF**	
1 = Reset Counters	1 = Reset Stats	1 = Page Down	1 = Page Up	1 = Resume Control	(Not Used)	(Not Used)	(Not Used)	26
(Not Used)	(Not Used)	(Not Used)	(Not Used)	(Not Used)	(Not Used)	(Not Used)	(Not Used)	
(Not Used)	(Not Used)	(Not Used)	(Not Used)	(Not Used)	(Not Used)	(Not Used)	(Not Used)	27
(Not Used)	(Not Used)	(Not Used)	(Not Used)	(Not Used)	(Not Used)	(Not Used)	(Not Used)	

* Set these three bits to specify the type of block as follows: 001 = Results, 010 = Configuration, 100 = Template, 101 = Statistics, 111 = Programmable Results Block Write

** If both ON & OFF bits are set, the output is forced OFF.

*** Do not write to these bits.

**** The first configuration on the card is 0000. When writing configurations from card to Color CVIM, remember that a card is made up of 32K slots, and the address is the address number of the slot (a 64K card contains two slots; a 512K card contains 16 slots). A complete image takes 12 slots.

**** *Used for Results and Stats Blocks only. Refer to page 4–17 for additional information.

Example Program for Accessing/Setting Discrete Bit Data

The following ladder logic program provides examples of:

- Triggering an inspection from a PLC.
- Enabling/Disabling the user access to the setup mode using the lightpen.
- Checking for valid results.
- Reading and displaying pass/fail/warning tool results (Window 1).
  Controlling screen display from a PLC.

The program assumes that the Color CVIM module is located in rack 02 (processor address is 074 octal) and the PLC is in rack 00.

```
Program Listing Report              PLC–5/15    File CCVIM1             Rung 2:0
Rung 2:0
Specify the half of the toolset you want
|                                                                   Post first |
|                                                                   half of    |
|                                                                   the TS     |
|                                                                     O:022    |
+----------------------------------------------------------------------( )-----+
|                                                                        00    |

Rung 2:1
This rung acquires an image, the Color CVIM one shots the input
| Trigger                                                                      |
| TS from                                                           Trigger    |
| outside                                                           the Camera |
|     B3                                                              O:020    |
+----] [---------------------------------------------------------------( )-----+
|       3                                                                12    |

Rung 2:2
The next two rungs control the SETUP bit so that the lightpen can not
get into the Color CVIM.
| Lock the                                                                     |
| SETUP                                                             Disable    |
| bit                                                               Setup      |
|     B3                                                              O:020    |
+----] [---------------------------------------------------------------( )-----+
|       1                                                                10    |

Rung 2:3
| Unlock the                                                                   |
| SETUP                                                             Enable     |
| bit                                                               Setup      |
|     B3                                                              O:020    |
+----] [---------------------------------------------------------------( )-----+
|       2                                                                11    |
```

Example Program For Accessing/Setting Discrete Bit Data (cont'd)

```
Rung 2:4
When Data Valid bit is high, read the discrete results for window 1, then
light the correct status light.
| Data Valid                                                                   |
| for the                                    Window 1  |Window 1  |Pass        |
| TS                                         warning   |Fault     |light       |
|   I:020                                      I:021      I:021      O:000     |
+----] [------------------------------------+---]/[--------]/[--------( )----+-+
|      06                                   |     00         01         10   | |
|                                           |Window 1              Fail      | |
|                                           |Fault                 light     | |
|                                           |  I:021                 O:000   | |
|                                           +---] [-------------------( )----+ |
|                                           |     01                    11   | |
|                                           |Window 1  |Window 1  |Warning   | |
|                                           |warning   |Fault     |light     | |
|                                           |  I:021      I:021      O:000   | |
|                                           +---] [--------]/[--------( )----+ |
|                                                 00         01         12     |

Rung 2:5
Tells the Color CVIM to display All Tools Page
| Display                                                                      |
| All Tools                                                        All Tools   |
| Page                                                             Page        |
|                                                                   DISPLAY    |
|     B3                                                             O:021     |
+----] [----------------------------------------------------------+---(L)----+-+
|       4                                                         |     02   | |
|                                                                 |I/O Page  | |
|                                                                 |  O:021   | |
|                                                                 +---(U)----+ |
|                                                                 |     03   | |
|                                                                 |Results   | |
|                                                                 |Page      | |
|                                                                 |  O:021   | |
|                                                                 +---(U)----+ |
|                                                                       04     |

Rung 2:6
Tells the Color CVIM to display the results page.
| Display                                                          All Tools   |
| I/O Page                                                         Page        |
|                                                                   DISPLAY    |
|     B3                                                             O:021     |
+----] [----------------------------------------------------------+---(U)----+-+
|       5                                                         |     02   | |
|                                                                 |I/O Page  | |
|                                                                 |  O:021   | |
|                                                                 +---(L)----+ |
|                                                                 |     03   | |
|                                                                 |Results   | |
|                                                                 |Page      | |
|                                                                 |  O:021   | |
|                                                                 +---(U)----+ |
|                                                                       04     |
```

Example Program For Accessing/Setting Discrete Bit Data (Cont’d)

```
Rung 2:7
Tells the Color CVIM to display the results page.
| Display                                                                      |
| Results                                                          All Tools   |
| Page                                                             Page        |
|                                                                   DISPLAY    |
|     B3                                                             O:021     |
+----] [----------------------------------------------------------+---(U)----+-+
|       6                                                         |     02    | |
|                                                                 |I/O Page  | |
|                                                                 |  O:021   | |
|                                                                 +---(U)----+ |
|                                                                 |     03    | |
|                                                                 |Results   | |
|                                                                 |Page      | |
|                                                                 |  O:021   | |
|                                                                 +---(L)----+ |
|                                                                       04     |

Rung 2:8
One must give a lightpen request to change display modes.
|  Display                                                                     |
|  All Tools                                                        Lightpen   |
|  Page                                                             Request    |
|      B3                                                              O:020    |
+-+---] [----+---------------------------------------------------------( )-----+
| |      4   |                                                           14    |
| |Display   |                                                                 |
| |I/O Page  |                                                                 |
| |    B3    |                                                                 |
| +---] [----+                                                                 |
| |      5   |                                                                 |
| |Display   |                                                                 |
| |Results   |                                                                 |
| |Page      |                                                                 |
| |    B3    |                                                                 |
| +---] [----+                                                                 |
|        6                                                                     |

Rung 2:9
|                                                                              |
+--------------------------------[END OF FILE]---------------------------------+
|                                                                              |

NO MORE FILES
```

Accessing Results and Configuration Information

A host also has access to actual results block information such as measured lengths, number of black pixels, etc. Transfer of result and configuration data is accomplished using block transfers. There are two types of blocks that can be transferred:

- Results Blocks
- Configuration Blocks

Depending upon the source and destination of the data blocks, the following transfers can be made:

Reading Results (Color CVIM module to SYS Host)

Results Blocks. There are six inspection results blocks (refer to Appendix A). Five of these blocks have a preconfigured structure. You can configure the sixth block so that only the information you require is transferred.

Transferring Configurations (Color CVIM module to CFG HOST and/or CFG HOST to Color CVIM module)

Configuration Blocks. There are 213 configuration blocks which contain the Color CVIM module setup information, tool parameters, operating environment instructions, camera setups, I/O operation, and operating modes. Each block transfer is limited to 64 words maximum. You can request blocks one at a time or in groups. Refer to Appendix A (Overview) and D (Configuration Data) for a description of the configuration block data.

When transferring blocks of data with the CFG or SYS Hosts, note the following requirements:

You should assign a length of 0 to all block transfer commands. This allows the Color CVIM module to specify the length of the block in words.

All block transfers address the lowest Group and Module Locations (0). You must set the bits in output word 3 to designate function of Results, Configuration, or Template transfer.

The SYS or CFG Host must initiate all block transfers.

Transferring Results Blocks

Results blocks are transferred using block transfer reads. These blocks contain inspection result information such as: tool results, fault data, etc. Of the six results blocks, five are pre–configured and one block is user configurable (refer to next section). This means that you can program the contents of the block to contain only the specific data you require. Before transferring a results block you must inform the Color CVIM module of the Block Transfer Type and Block Number by setting discrete bit information using simple ladder programming (refer to Table 4.C):

Set bit 0 of output word 3 to indicate RESULTS block transfer.

Use bits 10, 11, and 12 of output word 3 to indicate which of the six blocks to read. Refer to Appendix A and D.

Remember that result block addresses for the Color CVIM start with Block 01.

ATTENTION: To ensure that your results data is current and valid, you should use programming logic which synchronizes the transfer of data when inspections occur.

Use the Data Valid bits of input word 0 (bit 6) to detect when new inspection results are available. These bits are described in Chapter 3 (Local I/O).

Note: Later in this chapter we provide an example PLC program for retrieving results data.

Configuring Results Block 6 and Statistics Block Formats

13.Both the programmable results block and statistic blocks are configured to contain user specified results. To configure the data in results block #6:

Specify the information you want returned by setting the appropriate bits in the 10 word “programmable results / statistics block” in the PLC as shown in Table C.5 (Page C–15). For example, reference window 1 line gage, window 2, window 3, etc.

If configuring a results block– Set bits 0, 1, and 2 of output word 3. This will set the Color CVIM module to receive the “program” for results block 6.

If configuring a statistics block– Set bits 0 and 2 of output word 3. This will set the Color CVIM module to receive the “program” for the statistics block.

For the Results Block 6 and Statistics Block transfer writes, set the toolset active bits as follows:
O:23/04 = 1
05 = 0
06 = 0
07 = 01.

Perform a Block Transfer Write to transfer the 10 word “program“ from the host to the Color CVIM module.

If reading results block– Read results block 6 and check word 1 for error bits and words 2 through 63 for valid data.

Note: Refer to Appendix C and verify that your results will not require more than 62 words, this will ensure that the results will fit in the allocated block. The results are returned in the ascending order of their appearance in the programmable block (reference windows before windows, window 1 before window 2, etc.) It is the responsibility of the programmer to track the order and location of the data.

ATTENTION: The format information for the programmable results block and statistics blocks are stored in Color CVIM module RAM. The data does not get saved into the EEPROM with other configuration information. This means that the data will be lost when the power is turned off.

To avoid losing data, your program should contain a test for the Power Up condition. Discovery of this condition should initiate an automatic download of the configuration for this block.

Converting Results Data

Some of the results data described in Appendix C is stored in a “16 point 16” format while other data is stored as a 32 bit integer. Refer to the following chart:

WINDOW	**FORMAT**
Luminance	**16.16***
Object Counting	**32 bit**
Pixel Counting	**32bit**
Color ID	**32 bit**
Color Match	**32 bit**
GAGE	**FORMAT**
Linear Measure	**16.16**
Object	**32 bit**
Pixels	**32 bit**
Edge	**32 bit**
Angular Measure	**16.16**
Light Probe	**16.16**
Reference Line	**16 bit**
Reference Window	**16 bit**
Reference Window Theta	**16.16**

*16.16 means that the first 16 bits indicate the integer and the second 16 bits the fraction (refer to Appendix A for more information). If you are transferring results data to a PLC, you may need to convert the “16 point 16” format to a PLC floating point number. You can convert results data using the following equation:

$$\text{PLC Floating Point Number} = \text{Integer} + \frac{\text{Fraction}}{65536.0}$$

The following example assumes that you are converting a “16 point 16” value of 2.75. The value 2.75 is stored as follows:

Bit#		15**	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	←PLC
Integer → N7:1 = 2	=	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	
Fraction → N7:2 = .75	=	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

** This bit is the sign bit in PLC integer files (1 = Negative 0 = Positive)

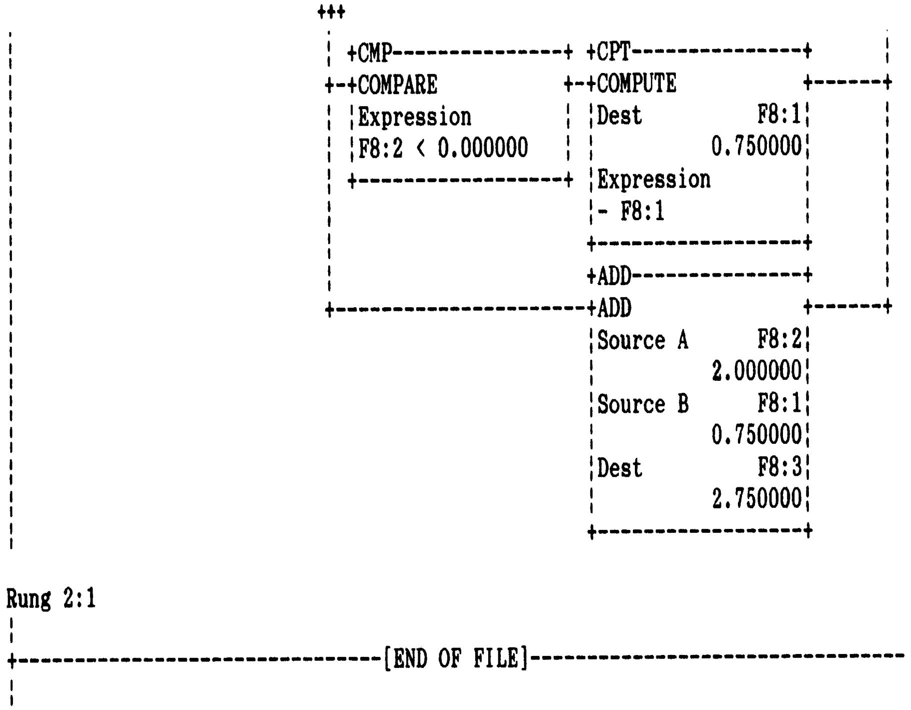
To help you, we have provided the following sample program. The program begins on the next page.

Converting Results Data (cont'd)

```
                                                                     Page 1
Ladder Listing               Processor Address: 000 octal          Rung 2:0
   Rung 2:0
   This program converts Color CVIM 16.16 to PLC Floating Point.
   The addresses N7:41 and N7:42 are used by gage 1 in results
   block 1.

¦  B3                                               +MOV---------------+
+--] [---------------------+----------------------+MOVE               +------+-
¦    1                     ¦                      ¦Source       N7:41¦       ¦
¦                          ¦                      ¦            -16384¦       ¦
¦                          ¦                      ¦Dest          F8:0¦       ¦
¦                          ¦                      ¦      49152.000000¦       ¦
¦                          ¦                      +------------------+       ¦
¦                          ¦ +CMP---------------+ +CPT---------------+       ¦
¦                          +-+COMPARE           +-+COMPUTE           +------+
¦                          ¦ ¦Expression        ¦ ¦Dest          F8:0¦       ¦
¦                          ¦ ¦F8:0 < 0.000000   ¦ ¦      49152.000000¦       ¦
¦                          ¦ +------------------+ ¦Expression        ¦       ¦
¦                          ¦                      ¦F8:0 +            ¦       ¦
¦                          ¦                      ¦65536.000000      ¦       ¦
¦                          ¦                      +------------------+       ¦
¦                          ¦                      +DIV--------------------+  ¦
¦                          +----------------------+DIVIDE                 +--+
¦                          ¦                      ¦Source A          F8:0¦  ¦
¦                          ¦                      ¦          49152.000000¦  ¦
¦                          ¦                      ¦Source B  65536.000000¦  ¦
¦                          ¦                      ¦                      ¦  ¦
¦                          ¦                      ¦Dest              F8:1¦  ¦
¦                          ¦                      ¦              0.750000¦  ¦
¦                          ¦                      +----------------------+  ¦
¦                          ¦                      +MOV---------------+      ¦
¦                          +----------------------+MOVE              +------+
¦                          ¦                      ¦Source       N7:40¦      ¦
¦                          ¦                      ¦                 2¦      ¦
¦                          ¦                      ¦Dest          F8:2¦      ¦
¦                          ¦                      ¦          2.000000¦      ¦
¦                          ¦                      +------------------+      ¦
                          +++
```

Converting Results Data (cont’d)



```
                              +++
|                              | +CMP---------------+ +CPT---------------+       |
|                              +-+COMPARE           +-+COMPUTE           +-------+
|                              | |Expression        | |Dest          F8:1|       |
|                              | |F8:2 < 0.000000   | |          0.750000|       |
|                              | +------------------+ |Expression        |       |
|                              |                      |- F8:1            |       |
|                              |                      +------------------+       |
|                              |                      +ADD---------------+       |
|                              +----------------------+ADD               +-------+
|                                                     |Source A      F8:2|
|                                                     |          2.000000|
|                                                     |Source B      F8:1|
|                                                     |          0.750000|
|                                                     |Dest          F8:3|
|                                                     |          2.750000|
|                                                     +------------------+

Rung 2:1
|
+--------------------------------[END OF FILE]---------------------------------
|
```

Transferring Configuration Blocks

You can transfer configuration block data between the Color CVIM module and CFG Host using block transfer reads and writes. These blocks contain the operating instructions for the Color CVIM module (refer to Appendix D). When transferring configuration blocks, note the following:

When the Color CVIM module is receiving configuration blocks from a CFG Host, the Color CVIM module will leave the active run mode, set the module busy bit, turn off local I/O, turn off the data valid bit, and ignore any input triggers (setup menu option is also disabled). After receiving one or more new configuration blocks (and the last block bit), the Color CVIM module will validate the entire configuration since many of the operating parameters are interrelated.

ATTENTION: You **must** monitor both the input and the output status bits.

If the Color CVIM module detects an invalid configuration, the new configuration will be ignored and the Color CVIM module will set the Configuration Fault bit and operate using the old configuration.

Transferring Configuration Blocks (cont'd)

You must use the discrete I/O bits in conjunction with block transfers to inform the Color CVIM module of the Block Transfer Type, Block Number and, Last Block by setting discrete bit information using simple ladder programming (refer to Table 4.C):

Set bit 1 of output word 3 to indicate a CONFIGURATION block transfer.

Use bits 10 through 17 of output word 3 to indicate which block to transfer. Refer to Appendix D for block numbers.

Set bit 3 in output word 3 to tell the Color CVIM to send the last block. This bit must be set one full scan before sending the last block. If you forget to set this bit, the Color CVIM module will wait for an indefinite period of time for more data.

Note: Later in this chapter we provide an example PLC program for accessing configuration data.

Example Program for Accessing Results Data

The following program illustrates one way to read Results Blocks from the Color CVIM. It does the following:

1. Initializes the bits needed to do the Block Transfer of Results Block 1 and posts the Data Value of the tool on the RIO channel.
2. Triggers the Color CVIM
3. Waits for Data Valid signal for the inspection (Note that the Trigger must go *low* before Data Valid can go *high*).
4. If the Block Transfer enables, resets the enable bit.
5. After the Block Transfer is complete, moves the number of missed triggers to N7:70.
6. Compares the new value of missed triggers with the old value, and turns off the output if they are equal.
7. Compares the number of missed triggers with the stored value. If they are not equal, turns on an output and transfers the output.

Note: We offer the following example only to show you how to use discrete bits via Remote I/O with a Color CVIM even though using the Data Valid signal via Remote I/O is not necessarily the preferred method. Due to real–world timing considerations, you will normally use a hard–wired Data Valid signal from the 2801–JMB board to your control device.

Note: Be sure to reset the counters on the Color CVIM and set N7:200 to zero.

The program begins on the next page.

Example Program for Accessing Results Data, Cont’d

```
Program Listing Report              PLC–5/15    File CCVIM2             Rung 2:0
Rung 2:0
Initialize the appropiate bits to read the results block and use Data Valid.
|                                                                  Specify     |
|                                                                  first half  |
|                                                                  of the      |
|                                                                  toolset     |
|                                                                    O:022     |
+-----------------------------------------------------------------+---(L)----+-+
|                                                                 |     00   | |
|                                                                 |Specify   | |
|                                                                 |results   | |
|                                                                 |block     | |
|                                                                 |  O:023   | |
|                                                                 +---(L)----+ |
|                                                                 |     00   | |
|                                                                 |Block 1 of| |
|                                                                 |6 possible| |
|                                                                 |  BLOCKS  | |
|                                                                 |  O:023   | |
|                                                                 +---(L)----+ |
|                                                                       10     |

Rung 2:1
This rung acquires an image, the Color CVIM one shots the input
| External                                                                     |
| trigger                                                           Trigger    |
| bit                                                               the Camera |
|     B3                                                              O:020    |
+----] [---------------------------------------------------------------( )-----+
|       1                                                               12    |

Rung 2:2
Read results block from the Color CVIM.
| Data Valid                                              Read from            |
| for the                                                 the Color            |
| TS                                                      CVIM                 |
|   I:020      B3                               +BTR--------------------+      |
+----] [-----[ONS]------------------------------+BLOCK TRNSFR READ      +-(EN)-+
|      06        2                              |Rack                 02|      |
|                                               |Group                 0+-(DN) |
|                                               |Module                0|      |
|                                               |Control Block    N7:100+-(ER) |
|                                               |Data file          N7:0|      |
|                                               |Length                0|      |
|                                               |Continuous            N|      |
|                                               +-----------------------+      |

Rung 2:3
Clear block transfer read error, if it occurs.
| BTR Error                                                         BTR Enable |
| bit                                                               bit        |
|   N7:100                                                            N7:100   |
+----] [---------------------------------------------------------------(U)-----+
|      12                                                               15    |
```

Example Program for Accessing Results Data, Cont'd

```
Rung 2:4
Set n7:70 to the number of triggers, before giving the system a new trigger.
|                                                              See if the      |
|                                                              number of       |
|                                                              missed          |
| BTR Done                                                     triggers        |
| bit                                                          increased       |
|   N7:100                                                +FLL---------------+ |
+----] [--------------------------------------------------+FILL FILE         +-+
|      13                                                 |Source       N7:59| |
|                                                         |Dest        #N7:70| |
|                                                         |Length           1| |
|                                                         +------------------+ |

Rung 2:5
If the number of missed triggers did not increase turn off the warning light.
|      See if the                                                              |
|      number of                                                               |
|      missed                                                                  |
|      triggers                                                     Pass       |
|      increased                                                    light      |
| +EQU---------------+                                                O:000    |
+-+EQUAL             +-------------------------------------------------(U)-----+
| |Source A     N7:70|                                                   10    |
| |                 0|                                                         |
| |Source B    N7:200|                                                         |
| |                 0|                                                         |
| +------------------+                                                         |

Rung 2:6
If the number of missed triggers are not equal, move the new larger value and
turn on the warning light.
|      See if the                                                              |
|      number of                                              Move the         |
|      missed                                                 new number       |
|      triggers                                               of missed        |
|      increased                                              triggers         |
| +NEQ---------------+                                   +FLL---------------+  |
+-+NOT EQUAL         +----------------------------------++FILL FILE         ++-+
| |Source A     N7:70|                                  ||Source      N7:70|| |
| |                 0|                                  ||Dest       #N7:200|| |
| |Source B    N7:200|                                  ||Length           1|| |
| |                 0|                                  |+------------------+| |
| +------------------+                                  |         Pass       | |
|                                                       |         light      | |
|                                                       |           O:000    | |
|                                                       +-------------(L)----+ |
|                                                                       10     |

Rung 2:7
|                                                                              |
+--------------------------------[END OF FILE]---------------------------------+
|                                                                              |

NO MORE FILES
```

Example Program For Accessing Configuration Data

The following program provides an example of using bi–directional block transfers to:

Transfer Color CVIM module configuration data to a PLC.

Modify the data. In this program we move the location of Window 1 up or down.

Transfer the reconfigured data back to the Color CVIM module from the PLC.

```
Rung 2:0
SET O:023/01 TO INDICATE A CONFIGURATION BLOCK.  SET APPROPRIATE BTX BITS TO
INDICATE BLOCK 110.  THIS COMBINATION OF BITS IDENTIFIES CONFIGURATION BLOCK
110 FOR THE BTR AND BTW USED IN THIS SAMPLE PROGRAM.

MAKE SURE THAT YOUR COLOR CVIM CONFIGURATION HOST IS REMOTE I/O.  THE COLOR
CVIM MUST HAVE REMOTE I/O ENABLED, RACK 2 SELECTED AND A BAUD RATE OF 57.6K.
|                                                                  BTX TYPE    |
|                                                                    O:023     |
+-----------------------------------------------------------------+---( )----+-+
|                                                                 |      01   | |
|                                                                 |BTX 64 BIT| |
|                                                                 |  O:023   | |
|                                                                 +---( )----+ |
|                                                                 |      16   | |
|                                                                 |BTX 32 BIT| |
|                                                                 |  O:023   | |
|                                                                 +---( )----+ |
|                                                                 |      15   | |
|                                                                 |BTX 8 BIT | |
|                                                                 |  O:023   | |
|                                                                 +---( )----+ |
|                                                                 |      13   | |
|                                                                 |BTX 4 BIT | |
|                                                                 |  O:023   | |
|                                                                 +---( )----+ |
|                                                                 |      12   | |
|                                                                 |BTX 2 BIT | |
|                                                                 |  O:023   | |
|                                                                 +---( )----+ |
|                                                                       11     |

Rung 2:1
| REQUEST TO                                                                   |
| MOVE WIN 1                                                        MOVE WIN 1 |
| UP                                                                UP         |
|   I:010      B3                                                       B3     |
+----] [-----[ONS]-----------------------------------------------------(L)-----+
|      01       11                                                        1    |

Rung 2:2
| REQUEST TO                                                                   |
| MOVE WIN 1                                                        MOVE WIN 1 |
| DOWN                                                              DOWN       |
|   I:010      B3                                                       B3     |
+----] [-----[ONS]-----------------------------------------------------(L)-----+
|      02       12                                                        2    |
Rung 2:3
THE DATA LOCATION THIS BTR WILL READ IS DEFINED ON RUNG 0.
|                                                         BTR WIN 1            |
|  MOVE WIN 1|                                            CONFIG               |
|  UP        |BTR ENABLE|BTW ENABLE                       DATA.                |
|      B3       N7:100     N7:110               +BTR--------------------+      |
+-+---] [----+---]/[--------]/[-----------------+BLOCK TRNSFR READ      +-(EN)-+
| |       1   |     15         15                |Rack                 02|      |
| |MOVE WIN 1|                                  |Group                 0+-(DN) |
| |DOWN      |                                  |Module                0|      |
| |    B3    |                                  |Control Block    N7:100+-(ER) |
| +---] [----+                                  |Data file          N7:0|      |
|        2                                      |Length                0|      |
|                                               |Continuous            N|      |
|                                               +-----------------------+      |

Rung 2:4
SUBTRACT 20 FROM THE Y LOCATION VALUE OF THE WINDOW TO MOVE THE WINDOW UP 20
PIXELS.
|           |MOVE WIN 1                                        WIN 1 Y         |
| BTR DONE  |UP                                                LOCATION        |
|   N7:100       B3                                       +SUB---------------+ |
+----] [--------] [---------------------------------------+SUBTRACT          +-+
|      13          1                                      |Source A      N7:4| |
|                                                         |               120| |
|                                                         |Source B        20| |
|                                                         |                  | |
|                                                         |Dest          N7:4| |
|                                                         |               120| |
|                                                         +------------------+ |

Rung 2:5
ADD 20 TO THE Y LOCATION VALUE OF THE WINDOW TO MOVE THE WINDOW DOWN 20 PIXELS.
|           |MOVE WIN 1                                        WIN 1 Y         |
| BTR DONE  |DOWN                                              LOCATION        |
|   N7:100       B3                                       +ADD---------------+ |
+----] [--------] [---------------------------------------+ADD               +-+
|      13          2                                      |Source A      N7:4| |
|                                                         |               120| |
|                                                         |Source B        20| |
|                                                         |                  | |
|                                                         |Dest          N7:4| |
|                                                         |               120| |
|                                                         +------------------+ |

Rung 2:6
| BTR DONE                                                          LAST BLOCK |
|   N7:100                                                            O:023    |
+----] [---------------------------------------------------------------(L)-----+
|      13                                                                03    |
Rung 2:7
AFTER READING AND MODIFYING THE CONFIGURATION DATA, SEND THE MODIFIED DATA TO
THE COLOR CVIM.  THE MODIFIED DATA WILL HAVE THE NEW WINDOW Y LOCATION.  ONCE
THE DATA IS SENT TO THE MODULE, THE NEW POSITION WILL BE DISPLAYED AFTER THE
NEXT TRIGGER.
|                                                        BTW WIN 1             |
|                      |MODULE    |                      CONFIG                |
| BTR DONE  |BTW ENABLE|BUSY      |LAST BLOCK            DATA                  |
|   N7:100     N7:110     I:020      O:023     +BTW--------------------+       |
+----] [--------]/[--------]/[--------] [-----++BLOCK TRNSFR WRITE     +-(EN)+-+
|      13         15         03         03    ||Rack                 02|     | |
|                                             ||Group                 0+-(DN)| |
|                                             ||Module                0|     | |
|                                             ||Control Block    N7:110+-(ER)| |
|                                             ||Data file          N7:0|     | |
|                                             ||Length                0|     | |
|                                             ||Continuous            N|     | |
|                                             |+-----------------------+     | |
|                                             |                    BTR DONE  | |
|                                             |                      N7:100  | |
|                                             +-----------------------(U)----+ |
|                                                                       13     |

Rung 2:8
| BTW DONE                                                         BTW DONE    |
|   N7:110                                                           N7:110    |
+----] [----------------------------------------------------------+---(U)----+-+
|      13                                                         |     13   | |
|                                                                 |MOVE WIN 1| |
|                                                                 |UP        | |
|                                                                 |    B3    | |
|                                                                 +---(U)----+ |
|                                                                 |      1   | |
|                                                                 |MOVE WIN 1| |
|                                                                 |DOWN      | |
|                                                                 |    B3    | |
|                                                                 +---(U)----+ |
|                                                                 |      2   | |
|                                                                 |LAST BLOCK| |
|                                                                 |  O:023   | |
|                                                                 +---(U)----+ |
|                                                                       03     |

Rung 2:9
THIS RUNG ALLOWS YOU TO RESET THE PROGRAM IF NECESSARY.
|                                                                  MOVE WIN 1  |
|                                                                  UP          |
|   B3                                                                 B3      |
+--] [------------------------------------------------------------+---(U)----+-+
|     3                                                           |      1   | |
|                                                                 |MOVE WIN 1| |
|                                                                 |DOWN      | |
|                                                                 |    B3    | |
|                                                                 +---(U)----+ |
|                                                                 |      2   | |
|                                                                 |LAST BLOCK| |
|                                                                 |  O:023   | |
|                                                                 +---(U)----+ |
|                                                                       03     |

Rung 2:10
|                                                                              |
+--------------------------------[END OF FILE]---------------------------------+
|                                                                              |
NO MORE FILES
```

Example 6008–SI Program

The following program was written using Microsoft® C Version 6.00 with an Allen–Bradley 6008–SI Series B card. The program will:

Identify the program and display preliminary information about it.

Prompt the user for the 6008–SI card address. This address is determined by the DIP switch settings on the card.

Prompt the user for the 6008–SI card baud rate multiplier. The multiplier is 1 for 57.6K Baud, and 2 for 115.2K Baud. (Note that the 6008–SI card does not support 230.4K Baud.)

Prompt the user for the 6008–SI card interrupt control line. This is determined by the jumper setting on the board itself.

Initialize the 6006SI card and prompts the user for the Color CVIM module rack address (0–7). The Color CVIM module address was configured on the Color CVIM module monitor using the light pen.

Display a five item menu which allows the user to perform the following functions:

1. Trigger. This initiates an inspection cycle.
2. Read Results. Reads the 128 discrete input bits of first part results.
3. Read Configuration. Uploads the entire Color CVIM module configuration.
4. Write Configuration. Downloads the entire Color CVIM module configuration.
5. Quit

Example 6008–SI Program (cont’d)

```
/* This program was compiled using Microsoft C Version 6.0 */
/* Color CVIM to 6008–SI sample communications program     */
/* Copyright Allen–Bradley      1–12–92         jrm, dms   */

#include <stdio.h>
#include <stdlib.h>

                         /* Include the 6008–SI definitions */
#include <h_6008si.h>

#define TRIGGER_1_BIT 0x0400

                         /* define storage for configuration data */
unsigned config[213][64], configlen[213];

void main()
    {

    QMR mr_pkt;
    unsigned segment;     /* segment of 6008–SI card */
    int baud, irq_l; /* 6008–SI card baud rate and interrupt line */
    unsigned status, err, CVIM_rack, block_num, numblocks, block1;
    unsigned block2, last_blk, x, t;
    int op_num, block_1;

    /* Display Program Execution and Description. */
    printf (”\nColor CVIM to 6008–SI sample communication program.\n”);
    printf (”Copyright Allen–Bradley  01–12–90 jrm, dms.\n”);
    printf (”This program was used and tested on a 33 MHz 386 Computer.\n”);
    printf (”The Color CVIM Communication Parameters for this program are:\n”);
    printf (”    CFG Host: Remote I/O\n”);
    printf (”    SYS Host: Remote I/O\n”);
    printf (”    Trigger Source = (Hosted)\n”);
    printf (”    Remote I/O Enabled\n”);
    printf (”    Remote I/O Rack Address 2\n”);
    printf (”    Remote I/O Data Rate 57.6\n”);
    printf (”    Color CVIM must be in RUNMODE\n”);

    /* Prompt – enter address, baud rate, and interrupt line of 6008 card */
    printf (”\nColor CVIM to 6008–SI communications sample program\n\n”);
    printf (”Enter hex RAM address for 6008–SI card (e.g. D000): ”);
    scanf (”%x”, &segment);
    printf (”Enter baud rate multiplier for 6008–SI card (e.g. 1=57.6 2=115.2): ”);
    scanf (”%d”, &baud);
    printf (”Enter interrupt control line for 6008–SI card (e.g. 3 5 10 or 12): ”);
    scanf (”%d”, &irq_l);

    /* initialize the 6008–SI */
    status = setup_6008(baud, 1, irq_l, segment, &mr_pkt);
    if (status != OK)
        {
        printf (”Setup failed: command=%s, status=%s\n”,
        xlat_cmd(status), xlat_conf(mr_pkt.qmr_stat));
    if (status != C_AUTOCONF && status != C_SETUP)
        printf (”Scanner fatal error %d\n”, fatal_6008());
    abort();
    }

/* Place scanner in RUN mode */
mr_pkt.qmr_data[0] = CM_RUN;
status = mr_wait (C_SETMODE, &mr_pkt);
if (status != OK)
    {
    printf (”Setup failed: command=%s, status=%s\n”,
    xlat_cmd(status), xlat_conf(mr_pkt.qmr_stat));
    if (status != C_AUTOCONF && status != C_SETUP)
        printf (”Scanner fatal error %d\n”, fatal_6008());
    abort();
    }

/* Disable host watchdog.  For sample program ONLY ––
   not recommended for any application programs. */
host_active(–1);

/* Get CVIM rack address from the user */
printf (”Enter CVIM remote–I/O rack number (0–7):  ”);
scanf (”%d”, &CVIM_rack);

g_oit[8*CVIM_rack + 2] |= 0x0001; /* post tool results */

/* Start of main loop */
do  {
    printf (”\n\nOperations: \n\n”);
    printf (”1. Trigger Tool Set\n”);
    printf (”2. Read Results, Part 1\n”);
    printf (”3. Read Configuration\n”);
    printf (”4. Write Configuration\n”);
    printf (”\nEnter operation number (1–4) or –1 to quit: ”);

    scanf(”%d”, &op_num );   /* Convert user string input to a number */
    err = 0;
    switch (op_num)
        {

        case 1:         /* trigger tool set by toggling trigger bit */
            {
            g_oit[8*CVIM_rack] |= TRIGGER_1_BIT;
            for (t=0; t<25000; t++);
            g_oit[8*CVIM_rack] &= ~TRIGGER_1_BIT;
            err = g_op_stat & SO_FAULT;
            } break;

        case 2:         /* read discrete results part 1 */
            {
            /* display all 8 input words in hex */
            for (x=0; x<8; x++)
                printf (”%04X ”, g_ipt[8*CVIM_rack + x]);
            printf (”\n”);
```

```
                err = g_op_stat & SO_FAULT;
                } break;

            case 3:             /* read configuration */
                {
                /* read all config. blocks */
                for (block_num = 0; (block_num < 213) && !err; block_num++)
                    err = get_CVIM_block (CVIM_rack, 2, block_num+1,
                              config[block_num], &configlen[block_num]);
                } break;

            case 4:             /* write configuration */
                {
                /* write all config. but last block (0–212) */
                for (block_num = 0; (block_num < 212) && !err; block_num++)
                    err = send_CVIM_block (CVIM_rack, 2, block_num+1,
                              config[block_num], &configlen[block_num]);

                /* write last config block (213) */
                err = send_CVIM_block (CVIM_rack, 10, 213,
                          config[212], &configlen[212]);

                /* wait until CVIM busy bit is low  */
                for (t=65535; t>0 && (g_ipt[8*CVIM_rack] & 8); t––)
                    for (x=1; x<100; x++);
                if (t==0)
                    {
                    printf (”Time–out error: CVIM busy\n”);
                    err = –1;
                    }
                if (g_ipt[8*CVIM_rack] & 2)
                    printf (”Configuration ERROR.\n”);
                else
                    printf (”Configuration validation OK.\n”);
                } break;
            }                  /* end switch (op_num) statement */

        if (err)
            printf (”Error code: %4x\n”,err);
        } while (op_num >= 0);

    stop_6008();          /* shut down 6008 before quitting */
    }

/* do a BTR (read) from the CVIM */
int get_CVIM_block (CVIM_rack, block_type, block_num, data, length)
unsigned CVIM_rack, block_type, block_num, *data, *length;

    {
    static QBT block_pkt;
    unsigned err, status,x;

    /* display msg for program monitoring */
    printf (”get_CVIM_block %d(%d)\n”,block_type, block_num);
    /* Tell CVIM block number and type */
    g_oit[8*CVIM_rack + 3] = block_type + block_num * 256;

    /* Initiate the block transfer read */
    block_pkt.qbt_len = 0;       /* request 0 words */
    status = bt_read(16*CVIM_rack,&block_pkt);
    err = (status != OK);
    if (!err)
        {
        /* wait for completion of BTR */
        while (!bt_done(&block_pkt));
        err = (block_pkt.qbt_stat != SC_OK);
        if (!err)
            {
            /* store the block data and length */
            *length = block_pkt.qbt_len;
            memcpy (data, block_pkt.qbt_data, *length * 2);
            }
        }
    return (err);
    }

/* performs a BTW (write) to the CVIM */
int send_CVIM_block (CVIM_rack, block_type, block_num, data, length)
unsigned CVIM_rack, block_type, block_num, *data, *length;

    {
    static QBT block_pkt;
    unsigned err, status,x;

    /* display msg for program monitoring */
    printf (”send_CVIM_block %d(%d)\n”,block_type, block_num);

    /* Tell CVIM block number and type */
    g_oit[8*CVIM_rack + 3] = block_type + block_num * 256;

    /* Initiate the block transfer write */
    block_pkt.qbt_len = *length;
    memcpy (block_pkt.qbt_data, data, *length * 2);
    status = bt_write(16*CVIM_rack,&block_pkt);
    err = (status != OK);

    if (!err)
        {
        /* wait for completion of BTW */
        while (!bt_done(&block_pkt));
        err = (block_pkt.qbt_stat != SC_OK);
        }
    return (err);
    }
```

Chapter 5

Using the RS–232 Ports

Chapter Objectives

In this chapter we describe how to:

- Connect RS–232 device(s) to the Color CVIM module.
- Obtain results data using ASCII or DF1 protocols.
- Upload and download configurations.

In addition, this chapter provides example programs.

RS–232 Communications

Using the RS–232 interface you can link a variety of devices to the Color CVIM module:

- Computers
- Operator Interfaces such as Allen–Bradley Industrial Computers and Terminals with serial ports.
- I/O modules such as the Flexible Interface Module (Catalog No. 2760–RB) or ASCII module (Catalog No. 1771–DA).

All commands are simple ASCII and/or Hexadecimal strings. Refer to Appendix E for an ASCII conversion chart. These commands can be generated using a variety of programming languages (C, Fortran, BASIC). This chapter provides a sample ASCII program (written in BASIC) and a sample DF1 program (written in C).

ASCII and DF1 Protocols

There are two protocol options when you select an RS–232 communications port (A or B):

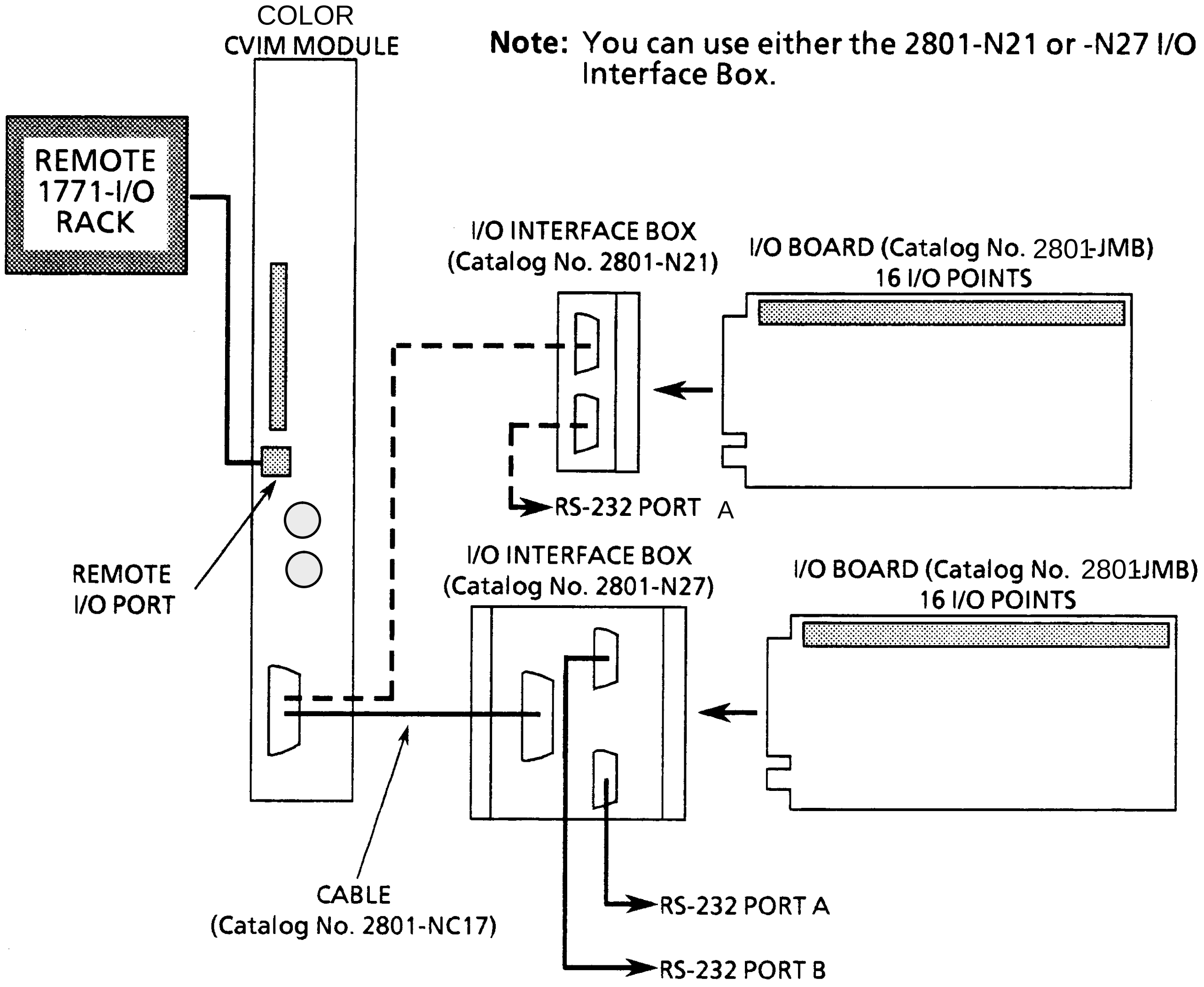
- ASCII
- DF1

This chapter describes both of these options. First we describe the ASCII protocol (page 5–6) and then the DF1 protocol (page 5–33).

Equipment Connections

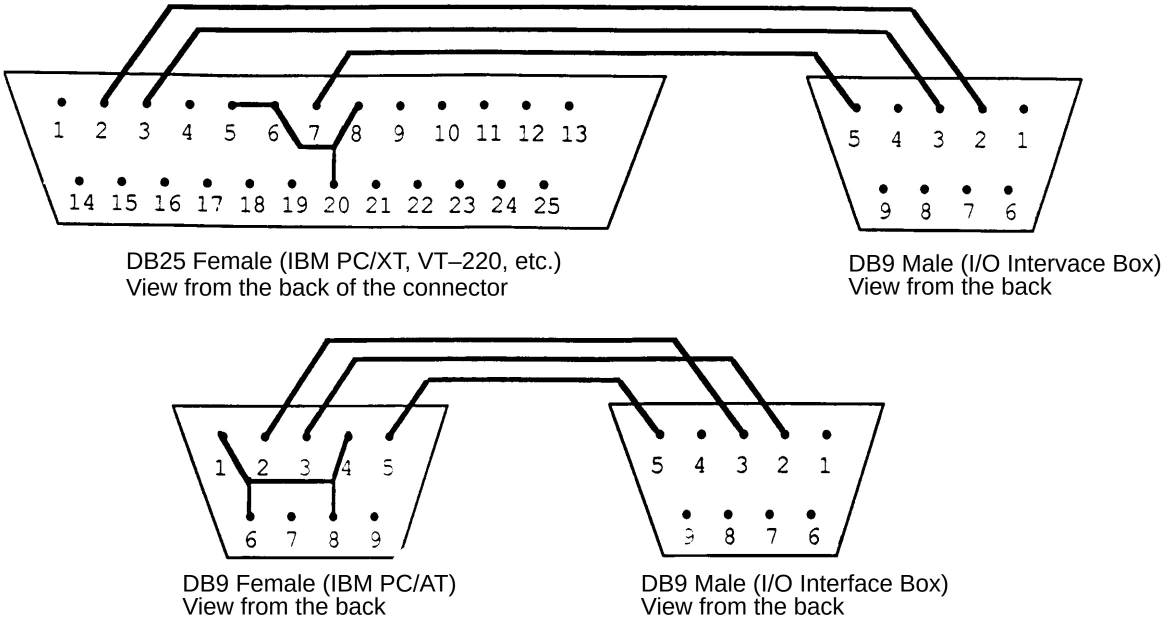
As shown in Figure 5.1, the RS–232 ports (A & B) are located on the I/O Interface Boxes (Catalog No. 2801–N21, –N27). The I/O Interface Box is connected to the MODULE I/O port on the front of the Color CVIM module. You will need a communications cable to link your host device to the Color CVIM module. Refer to Figure 5.2 for diagrams of host to I/O Interface Box cabling.

Figure 5.1
RS–232 Equipment Connections.



Equipment Connections (cont'd)

Figure 5.2
RS–232 Cabling.



Note: Connections for Catalog No. 2801–N27 I/O Interface Box RS–232 Port A with Color CVIM Series A Module is shown in this illustration. Refer to Chapter 3 (Tables 3.B, 3.C, 3.D, and 3.E) for other RS–232 Connections.

What Functions can be performed over the RS–232 Interfaces?

Any Color CVIM communications port can be used to read inspection result information. (Refer to Appendices B and C for more information.) In addition, any device selected as a system (SYS) host can:

- Change run–time display menus.
- Enable/Disable local I/O board.
- Force local I/O On or Off

A host device selected as a configuration host can also:

- Upload or download Color CVIM module configurations for inspections. Refer to Appendix D.
- Issue Configuration Read/Write commands between the following Color CVIM module memory locations:

  Color CVIM module Random Access Memory (RAM) and Color CVIM module Electrically Erasable Programmable Read Only Memory (EEPROM). RAM is volatile and EEPROM is non–volatile.

Color CVIM module RAM and RAM card. The RAM card slides into a slot on front of the Color CVIM module. Color CVIM module RAM and host memory.

Color CVIM Module Configuration Instructions

If you are using the RS–232 ports (A or B), you must configure the Color CVIM module as follows:

Set the Baud Rate(s)

1 Select the setup menu <Setup>.
2 Select the environment menu <Envir/Cam>.
3 Select the I/O menu <I/O>.
4 Select RS–232 communications <RS–232 A> or <RS–232 B).
5 Select protocol <DF1> or <ASCII>.
6 Select the Baud rate which matches your host device; from 300 to 19.2K Baud.

Note: When you select RS–232 communications, the data format is fixed as follows:

- 8 Data Bits
- 1 Stop Bit
- No Parity

Select the CFG and SYS Hosts

Note: The following steps are not necessary if you are just reading results data.

1 Select the setup menu <Setup>.
2 Select the environment menu <Envir/Cam>
3 Select the system menu <System>
4 Select a host menu <CFG Host> or <SYS Host>.
5 Select RS–232 port for host communications <RS–232A> or <RS–232B>.

Select the Protocol

6 Select the I/O menu <I/O>.
7 Select RS–232 communications <RS–232 A> or <RS–232 B>.
8 Select either <ASCII> or <DF1>.

9 Select Baud Rate <300>, <1200>, <2400>, <4800>, <9600>, or <19,200>.

Select the Color CVIM module Trigger Source

10 Select the trigger source menu <Trigger Source>.

11 Select either <I/O>, <Hosted>, or <Auto Trigger> trigger source. *Select hosted trigger if you are using the RS–232 trigger commands. Use I/O trigger if you are using the discrete I/O inputs as a trigger.*

Note: The next section of this chapter describes ASCII protocol followed by a description of DF1 protocol.

ASCII Protocol

In describing the ASCII Protocol we use the following conventions:

- Non–printable ASCII control characters are represented as follows:

  [CR] = Carriage Return
  [LF] = Line Feed
  ___ = Space

- ASCII commands are provided in large bold characters:

  >RR, RB,3 [CR]

  Unless _ is specified, there are no spaces between characters. Some commands have fields which can contain variable data such as number of times a command is repeated, block numbers, data, etc. These fields are shown using lowercase lettering:

  >W,CBn,d [CR]

  In this example, the letters **n** and **d** indicate data which is variable. The other characters indicate fixed data.

Overview

After you have made the equipment connections and configured the Color CVIM module for RS–232 communications, all ASCII strings generated by the host will be interpreted as commands. The Color CVIM module will then validate the command structure. If the command has an acceptable structure the Color CVIM module will reply: **[CR][LF].** Refer to Appendix E for an ASCII conversion chart. If the command has an incorrect structure the Color CVIM module will respond: ? **[CR] [LF].** The Color CVIM module will process all validated commands and discard any invalid commands. Data may or may not be returned with a command depending upon the type of command that was sent.

Note: A simple way to test the RS–232 links is to send the Color CVIM module a [CR]. If you have the port properly connected and the Color CVIM module configured for RS232, the Color CVIM module should send a ? [CR][LF] in response. If no response is provided, check your connections and Color CVIM module configuration.

Note: Some commands cause a continuous flow of returned data. To stop the flow of data you should send another command (valid or invalid). We recommend using a [CR] to stop the transmission of data.

ASCII Character Set

The Color CVIM module recognizes the following ASCII characters; all other characters are ignored.

- Upper and lowercase letters A through Z (case is insignificant).
- Symbols:

  > (greater than)
  * (star)
  , (comma)
  – (dash)
  (space) represented by __

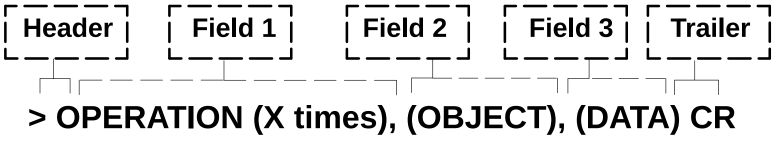
- Nonprintable control characters:

  CR (carriage return)
  LF (line feed)
  XON
  XOFF

- Numbers 0 through 9

Command Structure

Each command the host device sends to the Color CVIM module consists of an ASCII string of characters beginning with > and terminated with a **[CR].** Characters in between are separated into fields by commas. The following shows the structure of a typical command:



( ) Indicates Optional Information

Note: The x times modifier is only used with certain commands to indicate the number of times the command is to be performed. The range for this value is between 0 and 255. A value of 0 indicates infinity. If you do not specify a value, a default of 1 is provided.

Command Structure (cont'd)

There are three types of fields:

- Operation Field– This field contains commands directed to the Color CVIM module. There can only be one operation per command line. Some operations don't require any additional fields while others may require an object field, data field, or both. Note that some commands cannot be used while the Color CVIM module is in SETUP mode. If an operation cannot be performed because either the wrong host port has been selected or the Color CVIM module is in the SETUP mode, the Color CVIM module will respond to each command with **?[CR][LF]**.

- Object Field– Object fields specify data that configures the operation of the Color CVIM module.

  The object field contains alphanumeric characters which specify one or more objects. Individual objects are specified by name. Multiple objects (of the same type) are specified with an "*" for all objects of this type or by using a "–" to indicate a range of objects.

  In the description of each command we specify the objects that can be entered into a command.

- Data Field– Contains data.

XON/OFF Flow Control

XON/XOFF characters control the flow of data between the Color CVIM module and the host. The XON character is transmitted by the receiving device to indicate that data can be transmitted. The XOFF character is transmitted when the receiving device cannot accept any more data (data buffers are filled). When the receiving device can accept more data, it sends another XON character. The following characters are used: XON =^Q (CTRL Q) XOFF =^S (CTRL S).

Deactivate Forces

Use the deactivate force command to return outputs on the 2801–JMB local I/O board to the Color CVIM module assigned functions. The deactivate forces command is:

>DF [CR]

After executing the command, the Color CVIM module will return: **[CR][LF].** No data is returned If you do not have the proper command structure, the Color CVIM module will return: **?(CR][LF].**

Echoing Data

Use the echo command to check the communications link. This command will return the same same string of characters that are sent out with the command. This command has the following structure:

>Ex,d [CR]

Where **x** specifies the number of times the Color CVIM module will echo the data field back to the host device. If you fail to specify an **x** value, a default value of 1 is assumed. **d** is the data that is to be echoed. The command is valid at any time.

For example:

>E2,HELLO [CR]

This example will cause the Color CVIM module to return the string:

[CR] [LF]
HELLO [CR] [LF]
HELLO [CR] [LF]

If you do not have the proper command structure the Color CVIM module will return:

?[CR] [LF]

Enable/Disable Outputs

Use this command to enable or disable outputs on the Local I/O Board (Catalog No. 2801–JMB). Use the following commands:

> EO [CR] *This command enables the outputs.*

> DO [CR] *This command disables the outputs.*

After executing the command, the Color CVIM module will return: **[CR] [LF]**. No data is returned. If you do not have the proper command structure, the Color CVIM module will return: **?[CR][LF]**.

Forcing Local I/O

Use the force command to turn the local I/O outputs either on or off. This function can only be executed once per command. Use one of the following commands:

> F,On,1 [CR]	*Forces output(s) on.*
> F,On,0 [CR]	*Forces output(s) off.*

Where n is the output being forced on or off, outputs 1 through 14.

n = 1 to 14	*(individual outputs, can be nonconsecutive)*
X – Y	*(range of outputs X through Y)*
*	*(all of the outputs)*

For example:

> F,O*,1 [CR]	*This example will force all outputs on.*

Another example:

> F,O3–9,0 [CR]	*This example forces outputs 3 through 9 off.*

For example:

> F,O4–6,1 [CR]	*Forces outputs 4–6 on.*
> F,O8,1 [CR]	*Forces output 8 on.*
>F,O1–4,0 [CR]	*Forces ouputs 1–4 off.*

Notice that output #4 was forced on and then forced off. The force off takes precedence over the force on.

After executing a command, the Color CVIM module will return: **[CR][LF]**. If you do not have the proper command structure the Color CVIM module will return: **?[CR][LF]**. The outputs will remain in their forced states until a Deactivate Forces command is sent.

Loading Configurations

Use the load command to transfer configuration data to the Color CVIM module's RAM. Use one of the following commands:

> LO [CR] *Transfers configuration from the EEPROM to the Color CVIM module internal RAM.*

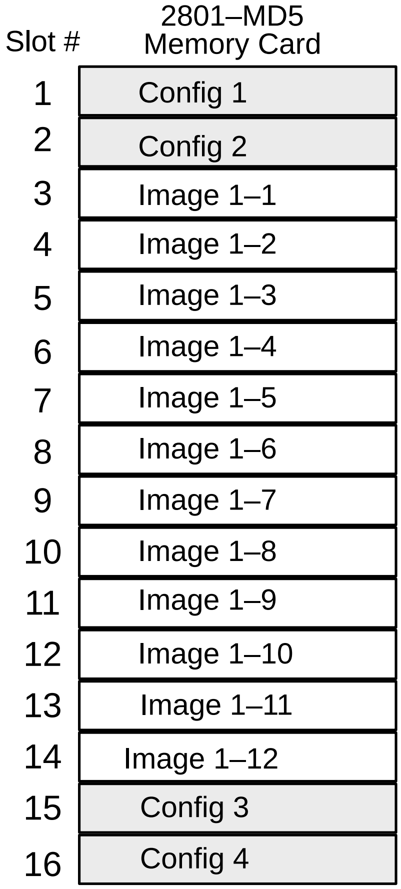
> LO,CC,1 [CR] *Transfers memory from the RAM Card area 1 memory to the Color CVIM module internal RAM.*

> LO,CC,2 [CR] *Transfers memory from the RAM Card area 2 memory to the Color CVIM module internal RAM.*

This function can only be executed once per command. You cannot use this command when the Color CVIM module is in the SETUP mode.

After executing a command, the Color CVIM module will return: **[CR][LF]**. No data is returned by the command. If you do not have the proper command structure or the Color CVIM module is in the SETUP mode, the Color CVIM module will return: **?[CR][LF]**.

Figure 5.3
Memory Requirements for Images and Configurations



Slot #	2801–MD5 Memory Card
1	Config 1
2	Config 2
3	Image 1–1
4	Image 1–2
5	Image 1–3
6	Image 1–4
7	Image 1–5
8	Image 1–6
9	Image 1–7
10	Image 1–8
11	Image 1–9
12	Image 1–10
13	Image 1–11
14	Image 1–12
15	Config 3
16	Config 4

Note: Memory cards are laid out in slots. A 64K card (2801–MD4) has room for two configurations. The 512K memory card (2801–MD5) has 16 slots, and can store up to 16 configurations. An image must be one of the first two items stored to the 512K card, and will occupy 12 contiguous free slots.

Lock Command

Use the lock command to disable the setup menu box so that the SETUP mode cannot be entered using the light pen. This function can only be executed once per command. There is no object associated with this command. The command has the following structure:

```
>L[CR]
```

After executing a command, the Color CVIM module will return: **[CR][LF].** No data is returned by the command. If you do not have the proper command structure the Color CVIM module will return: **?[CR][LF].** Use the unlock command to enable the setup menu box.

Unlock Command

Use the unlock command to enable the setup menu box so that a user can access the SETUP mode using the light pen. Use the following command:

>U[CR]

This function can only be executed once per command. There is no object associated with this command. After executing a command, the Color CVIM module will return: **[CR][LF]**. No data is returned. If you do not have the proper command structure, the Color CVIM module will return: **?[CR][LF]**.

Read Output Status

Use the read data command to read the status of the local I/O. This command has the following structure:

>Rx,On [CR]

Where n =	1 to 14	*(individual outputs)*
	X–Y	*(range of outputs X through Y)*
	*	*(all of the outputs)*

This function can be executed more than once per command by specifying an x times value.

For example:

> R,O14 [CR]	*This example reads the status of output #14 once.*

Another example:

> R0,O*[CR]	*This example continuously reads the status of all fourteen outputs.*

After executing a command, the Color CVIM module will return: **[CR][LF]** followed by the data. If you do not have the proper command structure, the Color CVIM module will return: **?[CR][LF]**. The format of the requested data is an ASCII representation of the output state (**1** = ON and **0** = OFF). Each character is followed by a space. The output conditions are transmitted in numerical order (output #1 then #2, etc.). The number of characters returned depends upon the number of outputs that are read. Since there are fourteen outputs, up to 28 data characters can be returned. After the data is sent, the Color CVIM module will terminate the data with: **[CR][LF]**. The following is an example of returned data from three outputs.

[CR][LF]1 0 0 [CR][LF]

Read Configuration Blocks

Use the read configuration command to read configuration data for the specified blocks (Upload Configurations). The command has the following structure:

>RC,CBn [CR]

Where n =	1 to 213	*(individual blocks)*
	X – Y	*(range of blocks X through Y)*
	*	*(all of the blocks)*

This function can only be executed once per command.

Refer to Appendix C for a description of the configuration blocks. You cannot use this command while the Color CVIM module is in the SETUP mode.

Examples:

>RC,CB135 [CR]	*Reads configuration block 135.*
>RC,CB99,CB7,CB1[CR]	*Reads configuration blocks 1, 7, then 99.*
>RC,CB1–213[CR]	*Reads all of the configuration blocks .*

After executing a command, the Color CVIM module will return: **[CR][LF]** followed by the data. If you do not have the proper command structure, the Color CVIM module will return: **?[CR][LF]**. The format of the requested data is in an ASCII representation of the specified block(s) in bytes. Each byte is represented by two hexadecimal characters (00 through FF) followed by a space. The first two words are the signature word indicating block type and number (refer to appendix D). Twenty bytes of data are transmitted in a line terminated with a **[CR][LF]**. The size of the configuration block(s) determines the number of lines that are returned.

Read Configuration Blocks (cont'd)

Refer to Appendix D for block descriptions and sizes. The following is an example of how the returned data appears for command **>RC,CB1–2 [CR]:**

Figure 5.4
Configuration Block Returned Data Format*

```
[CR] [LF]
48_01_02_00_00_00_00_00_44_65_66_61_75_6C_74_00_00_00_00_00_[CR][LF]
00_00_00_00_00_03_00_00_00_00_00_00_04_00_00_01_00_01_00_00_[CR][LF]
00_02_06_01_00_00_00_00_00_00_00_00_00_00_00_00_00_00_06_01_[CR][LF]
00_00_00_01_00_00_00_00_00_00_00_00_00_00_00_01_01_00_01_01–[CR][LF]
00_0D_FC_5F_9A_0A_00_19_FB_D1_[CR][LF]

48_02_FF_02_01_00_00_00_00_FA_00_10_00_01_01_00_00_00_00_00_[CR][LF]
00_00_00_3F_00_00_00_00_00_00_00_3F_00_00_00_00_00_00_00_64_[CR][LF]
00_32_01_2C_00_96_0A_62_00_01_00_00_00_01_00_00_00_01_00_00_[CR][LF]
00_01_00_00_00_01_00_00_00_01_00_00_00_64_00_32_01_2C_00_96_[CR][LF]
0A_62_00_01_00_00_00_01_00_00_00_00_80_00_00_00_00_00_00_00_[CR][LF]
00_00_00_00_00_00_00_01_00_00_00_00_00_00_00_00_00_00_00_01_[CR][LF]
00_00_00_00_00_00_49_CC_[CR][LF]
```

*The example above is valid as shown for DF1 Return Data mode. In ASCII mode there are no spaces.

Read Inspection Results

Use this command to read the *results of the last inspection*. Refer to Appendix B for a description of the results blocks. Use the following commands:

>RRx,o,d [CR]

Where:	*x*	=	*Number of times command is repeated.*
	o	=	*RL (specifies Reference Line)* *RW (specifies Reference Window)* *G (specifies Gage)* *W (specifies Window)* *LP (specifies Light Probe)*
	d	=	*Gage, Window, Reference Line, or Reference Window number.*

>RR [CR]	*Reads all discrete bit results.*
>RRx,P1 [CR]	*Reads discrete bit first part results.* *x = Number of times command is repeated.*
>RRx,P2 [CR]	*Reads discrete bit second part results.* *x = Number of times command is repeated.*
>RRx,RB,d [CR]	*Reads results block(s).* *x = Number of times command is repeated.* *d = Block number.*
>RRx, S [CR]	*Reads Color CVIM module status.*

Read Inspection Results (cont'd)

The read operation can be executed more than once per command by specifying an x times value. The data in the read results block commands indicate which results block (1, 2, 3, 4, 5, or 6) is being read (refer to Appendix C).

>RR0,P1 [CR]

This command continuously reads the first discrete bit results. (24 bytes returned)

>RR,RB, 3[CR]

This command reads results block 3. This operation is only performed once in this example. (128 bytes returned)

>RRx,RL,1[CR]

Reads the results of reference line #1. (4 bytes returned)

>RRx,RW,3[CR]

Reads the results of reference window #3. (28 bytes returned)

>RRx,G,21[CR]

Reads the results of gage #21. (4 bytes returned)

>RRx,W,11[CR]

Reads the results of window #11. (4 bytes returned)

>RRx,LP[CR]

Reads the results of light probe. (12 bytes returned)

>RRx,S[CR]

Reads the Color CVIM module status. (2 bytes returned)

Note: Refer to Appendix B, Table B.1, RS–232 word 0 for a definition of Color CVIM module status.

Read Inspection Results (cont'd)

After executing a command, the Color CVIM module will return: **[CR][LF]** followed by the data. If you do not have the proper command structure, the Color CVIM module will return: **?[CR][LF]**. After reading the results, the Color CVIM module will return the requested data. The format of the requested data is in an ASCII representation of the specified block(s) in bytes.

- If you requested results blocks, each byte is represented by two hexadecimal characters (00 through FF) followed by a space. Twenty bytes of data are transmitted in a line terminated with a **[CR][LF]**. Since the results blocks are 128 bytes in size, each block requires seven lines. Refer to Appendix C for block descriptions. The following is an example of the returned data format:

Figure 5.5
Numerical Results Block Returned Data Format

```
[CR] [LF]
61_01_04_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00~00_[CR][LF]
00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00~00_[CR][LF]
00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_01_00_00_[CR][LF]
00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_32_00_00_00_00_[CR][LF]
00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_[CR][LF]
00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_[CR][LF]
00_00_00_00_00_00_00_CC_[CR][LF]
```

- If you requested discrete bit information, the returned data will contain two counters and the discrete bit results. Each counter has 12 positions (10 characters, 2 spaces) reserved for a maximum value of 4,294,967,295.

Note: Counters are decimal values. All other fields are hexadecimal values.

The counter data is left justified and the remaining field is filled with spaces. The first counter contains the total number of triggers processed. The second counter contains the total number of faults. Both counters are expressed as decimal values. The results bit information (128 bits per comment, or 256 bits total), which follows the counters, is 16 bytes long. Each byte is represented by two hexadecimal characters (00 through FF) followed by a space. The following is an example of the returned data format:

Figure 5.3
Discrete Bit Results Returned Data Format

```
[CR][LF]
1234567890__1234567890__80_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_[CR][LF]
```

Refer to Appendix B for a description of the returned bytes.

Save Configuration

Use the Save command to transfer Color CVIM module configuration data to the local storage area (EEPROM) or the external RAM card (credit card memory).

Note: Depending upon the card size, up to 16 configurations can be saved to the RAM card (512K card).

Use one of the following commands:

>S[CR]	*Transfers configuration data from the Color CVIM module RAM to the EEPROM.*
>S,CC,X [CR]	*Transfers configuration data from the Color CVIM module RAM to the RAM card area X (01 –16).*

For example:

>S,CC,15 [CR]	Transfers configuration data from the Color CVIM module RAM to the RAM card area 15.

You cannot use this command when the Color CVIM module is in the SETUP mode.

After executing a command, the Color CVIM module will return: **[CR][LF]**. No data is returned. If you do not have the proper command structure, the Color CVIM module will return: **?[CR][LF]**.

Figure 5.6
Memory Requirements for Images and Configurations

Slot #	2801–MD5 Memory Card
1	Config 1
2	Config 2
3	Image 1–1
4	Image 1–2
5	Image 1–3
6	Image 1–4
7	Image 1–5
8	Image 1–6
9	Image 1–7
10	Image 1–8
11	Image 1–9
12	Image 1–10
13	Image 1–11
14	Image 1–12
15	Config 3
16	Config 4

Note: Memory cards are laid out in slots. A 64K card (2801–MD4) has room for two configurations. The 512K memory card (2801–MD5) has 16 slots, and can store up to 16 configurations. An image must be one of the first two items stored to the 512K card, and will occupy 12 contiguous free slots.

Select Image Displayed

Use the display object commands to select the information that is displayed on the monitor:

>W,D,d [CR]

Where **d** is the data that specifies the display to be viewed:

d = 1 *(Image only displayed)*
or 2 *(Failed tools displayed)*
or 3 *(All tools displayed)*
or 4 *(I/O page displayed)*
or 5 *(Results page displayed)*
or 6 *(Stats 1 page displayed)*
or 7 *(Stats 2 page displayed)*
or 8 *(Page up same display)*
or 9 *(Page down same display)*

>W,F,d [CR]

d = 1 *(Go on reject)*
or 2 *(Freeze on 1st reject)*
or 3 *(Freeze on all rejects)*
or 4 *(Freeze on next inspection)*
or 5 *(Halt on reject)*

>w,DC,d [CR]

d = 1 *(Resume)*
or 2 *(Reset stats)*
or 3 *(Reset counters)*
or 8 *(Page up)*
or 9 *(Page down)*

Example:

>W,D,2[CR] *This example will display failed tools.*

After executing the command, the Color CVIM module will return: **[CR][LF]**. No data is returned. If you do not have the proper command structure, the Color CVIM module will return: **?[CR][LF]**.

Set Configurable Results

Use this command to obtain a configurable results block. The results you want are specified by a list of tools and placed in results block #6. No data is returned until you use a read inspection results command for block #6. Use the following command:

```
>SR,d,d, etc. [CR]
```

Where d =	G1, G2, G3, G1–G3, etc.	(specifies Gages)
	W1, W2, W3, W2–5, etc.	(specifies Windows)
	RL1, RL2, etc.	(specifies Reference Lines)
	RLW, RW2, etc.	(specifies Reference Windows)
	LP	(specifies Light Probe)

The returned results block will be 128 bytes including the block signature (2 bytes) and trigger counter (last 4 bytes). Refer to page C–14, the ordering of the tools and data lengths are the same as the Remote I/O configurable results block.

Example:

```
>SR,G1,W2–5[CR]
```

This command places the results for gage 1 and Windows 2 through 5 in results block #6.

```
>RR,RB,6 [CR]
```

This command reads results block #6.

After executing the command, the Color CVIM module will return: **[CR] [LF]**. If you do not have the proper command structure, the Color CVIM module will return: **?[CR][LF]**. Refer to Read Inspection Results command for a description of the returned data format.

Set/Read Configurable Statistics

Use the read command to read statistical data for the light probe, reference windows, gages, and windows. Use the separate set command to set the number of samples and configure the statistics block. This command must be sent by the system host (SYS).

The set statistics command has the following structure:

>SSn,d,d,etc.[CR] *(Set command)*

Where n = Number of samples

Note: If n is 0, the Color CVIM module will continue to use the sample count configured during setup. Any other value will change the sample count (if this port is the SYS host).

Where d =	G1, G2, G3, G1–G3, etc.	(specifies Gages)
	WI, W2, W3, W2–5, etc.	(specifies Windows)
	RW1, RW2, RW3	(specifies Reference Windows)
	LP	(specifies Light Probe)

The read statistics command has the following structure:

> RSn [CR] *(Read Statistics Command)*

Where n = Number of times statistics block is read.

Statistics are accumulated until the number of samples is reached, at which point the statistics begin to reaccumulate. The statistics are accumulated based upon the number of triggers.

Examples of Set Statistics Command:

>SS50,LP,RW2[CR] *This example sets the number of samples to 50, configures the block to contain light probe and reference window #2 statistics.*

>SS100,G5,W12[CR] *This example sets the number of samples to 100, configures the block to contain gage #5 and window #12 statistics.*

Set/Read Configurable Statistics (cont'd)

Example of Read Statistics Command:

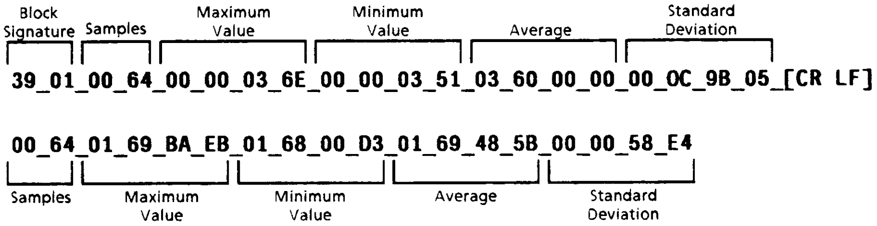
> RS5 [CR] *This example reads the statistics block five times.*

The data returned from the statistics block consists of:

- Block signature
- Number of samples, maximum, minimum, average, and standard deviation for each tool configured in the block.

The block signature is 2 bytes long. The number of samples is a 2 byte integer. The maximum and minimum values are each 4 bytes. The format of the data depends upon the operation (e.g. pixel count is an integer and linear gaging is a 16.16 fixed point value). Refer to page C–24 for data formats. Standard deviations are also 4 bytes each but are always 16.16 fixed point values. Averages are 24.8 fixed point values. Therefore, each tool statistic consists of 18 bytes with the exception of reference windows and the light probe, which contain 18 bytes for each feature or a total of 54 bytes. The statistics block is transmitted as two hexadecimal characters for each byte. The total number of bytes including the block signature should not exceed 128 bytes. The statistics block is read once for every number of specified samples. This means that if you read the statistics block five times with a sample number of 50, 250 triggers will have to be processed before the five reads are completed. The following shows the format of the returned data:

Figure 5.7
Statistics Block Returned Data Format



```
39_01_00_64_00_00_03_6E_00_00_03_51_03_60_00_00_00_OC_9B_05_[CR LF]

00_64_01_69_BA_EB_01_68_00_D3_01_69_48_5B_00_00_58_E4
```

Trigger Operation

Use the trigger operation command to initiate an inspection by the color CVIM module. Use the following commands:

>T, [CR] *Triggers an inspection.*

This function can only be executed once per command.

Note: When using this command you should make sure that the module is configured for a “hosted trigger source”.

After executing a command, the Color CVIM module will return: **[CR][LF]**. No data is returned. If you do not have the proper command structure, the Color CVIM module will return: **?[CR][LF]**.

Write Configuration (W)
Write Configuration (WC)

Use the write command to write data to configuration memory (download configuration). Use the following commands:

>W,CBn[CR] d

or

>WC,CBn[CR] d

Where n =	1 to 213	*(individual blocks)*
	X – Y	*(range of blocks X through Y)*
	1–213	*(all of the blocks)*

d = the data that is being written. The format of the data is in an ASCII representation of the specified block(s) in bytes. Each byte is represented by two hexadecimal characters (00 through FF) followed by a space.

Note: The WC write command functions like the W write command but allows listing of configuration blocks.

Write Configuration (W) Write Configuration (WC) (cont'd)

This function can only be executed once per command.

Refer to Appendix D for a description of the configuration blocks. You cannot use this command when the Color CVIM module is in the setup mode. When the Color CVIM module is receiving configuration blocks from a Host, the Color CVIM module will leave the active run mode and ignore any input triggers (setup menu option is also disabled). After receiving one or more new configuration blocks, the Color CVIM module will validate the entire configuration since many of the operating parameters are interrelated.

Example:

>W,CB1 [CR] 00__F1__etc

This example writes the data 00, F1, etc into configuration block #1. "_" = space character.

Example:

>WC,CB1,CB30–35,CB21[CR](data)*

This example writes the data into the specified blocks.

*All blocks and lines of data are separated by a **[CR],** and a response of **[CR][LF]** is sent for every command and data block.

After executing the command, the Color CVIM module will return: **[CR][LF]**. No data is returned. If you do not have the proper command structure, the Color CVIM module will return: **?[CR][LF]**.

Note: We recommend that you first read Module Busy low, then send the configuration. When you're finished downloading the configuration, check to make sure the "Configuration Fault" bit has not been set. Refer to Appendix B. You can check this bit by using the read inspection results command (>**RR,S [CR]**). No other ports should be active.

Command Summary

After you have become familiar with the ASCII commands, you can use the following command summary as a quick reference guide.

Table 5.A
ASCII Command Summary

Command	Command Structure	Field Descriptions
Deactivate Forces	>DF [CR]	
Disable Outputs	>DO [CR]	
Enable Outputs	> EO [CR]	
Echo Data	>E, data [CR]	Data = ASCII string

Force Outputs	>F, On, d [CR]	n = 1 to 14
		d = 0 or 1

Load Configuration From EEPROM to RAM	>LO [CR]	
Load Configuration From RAM Card to RAM	>LO, CC, d[CR]	d = 1 to 16*
Lock	>L [CR]	
Unlock	>U [CR]	
Read Output Condition	>R, On [CR]	n = 1 to 14
Read Configurable Statistics	>RSn [CR]	n = number of times read
Read Configuration	>RC, CBn [CR]	n = 1 to 213
Read All Discrete Bit Results	>RR [CR]	
Read Discrete Bit Results	>RR, Pn [CR]	n = 1 or 2
Read Results Block	>RR, RB, d [CR]	d = 1, 2, 3, 4, 5, or 6
	>RR,o,d[CR]	o = RL,RW,G,W,LP d = gage or window number
	>RR,S**	S = Status
Save to EEPROM from RAM	>S [CR]	
Save to RAM Card from RAM	>S, CC, d [CR]	d = 1 to 16*
Set Configurable Results	>SR,d,d,etc. [CR]	d = G1,G2,W1,W2, RW1, RL3, LP, etc.

* The number of configurations that can be stored on a RAM card depends upon the card size (512K card can hold 16 configurations).

** Refer to Appendix B, Table B.1, RS–232 word 0 for a definition of Color CVIM status.

Command Summary (cont'd)

Table 5.A
ASCII Command Summary (Cont'd)

Command	Command Structure	Field Descriptions

Set Configurable Statistics	>SSn,d,d,etc. [CR]	n = number of samples.
		d = G1, G2, W1, W2, RW1, LP, etc.

Trigger Inspection	>T,[CR]	
Write Display	>W, D, data [CR]	Data = 1 to 9 1 = Image only 2 = Failed Tools 3 = All Tools 4 = I/O Page 5 = Results Page 6 = Stats 1 Page 7 = Stats 2 Page 8 = Page Up 9 = Page Down
	>W, F, data [CR]	Data = 1 to 5 1 = Go On Reject 2 = Freeze On First Reject 3 = Freeze On All Rejects 4 = Freeze On Next Inspection 5 = Halt On Reject
	>W, DC, data (CRI	Data = 1 to 9 1 = Resume 2 = Reset Statistics 3 = Reset Counters 8 = Page Up 9 = Page Down

Write Configuration BIock(W)	>W, CBn [CR] data	n = 1 to 213
		Data= ASCII configuration data

Write Configuration Block(WC)	>WC,CBn,CBn,etc. [CR] data	n = 1 to 213
		Data = ASCII configuration data

Explanation of ASCII Programming Example

The following sample program (called CCVIMASC.BAS) was written on an Allen–Bradley 1784–T47 terminal using GW BASIC. This program obtains and processes discrete results from the Color CVIM module. The program will:

- Read the trigger count.
- Trigger an inspection.
- Detect when new data is available.
- Read discrete results part 1.
- Display discrete results part 1.
- Display a screen message if any of the first four windows fail.

A basic outline of the program is as follows:

Lines 10 to 99	Program identification and related information. Initialize program variables, configure the RS–232 port for 8 bit transmissions, select no parity, select 9600 Baud, and initialize the display monitor.
Lines 100 to 120	Prompt user to run program or exit.
Subroutine 2000	Reads results to find the current number of total triggers.
Subroutine 1000	Triggers the Color CVIM module inspection.
Line 200	Causes a continuous read of Color CVIM module block 1 results until new results are detected. New results are detected by an incrementing of the “total trigger” data.
Subroutine 2500	Converts the Color CVIM module results from hexadecimal to integer.
Lines 240 to 270	Analyze the discrete fail bits for windows 1 through 4 and display a message if a failure is detected.
Line 400	Sends the program to input line 100.

The program manipulates the returned data as follows:

Explanation of ASCII Programming Example (cont'd)

Assume the ASCII string from the Color CVIM module is:

```
CR LF 2114 __ __ __ __ __ __389 __ __ __ __ __ B0__80__
A2__ 00__00__(etc.)CR LF
```

Note: (__ = space, LF = Line Feed, CR = Carriage Return)

The 18 element hexadecimal array after the program receives the data:

R1(0) = 2114 = Decimal representation of total triggers processed.

R1(1) = 389 = Decimal representation of total master faults (failed inspections).

R1(2) = B0 = Hexadecimal representation of discrete input word 0 low byte.

R1(3) = 80 = Hexadecimal representation of discrete input word 0 high byte.

R1(4) = A2 = Hexadecimal representation of discrete input word 1 low byte (Window 1 Fault/Warning, Window 2 Fault/Warning, etc).

• • •

R1(17) = 00 = Hexadecimal representation of discrete input word 7 high byte (Gage 32 Fault/Warning, Gage 31 Fault/Warning, etc).

The decimal display on the monitor will appear as follows after the program manipulates the array:

```
2114      389      176      135      162

0         0        0        0        0
0         0        0        0        0
0         0        0
```

Analysis of R1(4) for window failure:

R1(4) = 162 (decimal). The binary representation is:

1 0 1 0 0 0 1 0

The three ones in this representation indicate fail discrete input conditions in windows 1, 3, and 4 (bits 1, 5, and 7 of word 1, see Table 4.A.).

To run the program, make sure the GWBASIC.EXE and program files are in the same directory. Then, at the DOS prompt, type GWBASIC b/ CCVIMASC <CR>.

ASCII Programming Example

The following is a sample ASCII program written in BASIC:

```
1 REM     RS–232 ASCII COLOR CVIM COMMUNICATIONS SAMPLE PROGRAM
2 REM    COPYRIGHT ALLEN–BRADLEY COMPANY, INC. 1992 JRM, DMS
3 :
4 :
10 OPEN”com1:9600,n,8,1,DS”AS#1: REM Open communications channel
20 DIM R1(17): REM Allocate storage for tool set results
30 HE$=”0123456789ABCDEF”: REM Used for hex to decimal conversion
50 CLS
60 PRINT ”RS–232 ASCII TO ALLEN–BRADLEY COLOR CVIM COMMUNICATIONS PROGRAM”
61 PRINT ”    COPYRIGHT ALLEN–BRADLEY COMPANY, INC. 1992 JRM, DMS”
62 PRINT
63 PRINT ”THIS PROGRAM WILL PERFORM THE FOLLOWING:”
64 PRINT ”    READ THE TRIGGER COUNT”
65 PRINT ”    TRIGGER AN INSPECTION”
66 PRINT ”    READ TRIGGER COUNT UNTIL IT IS INCREMENTED”
67 PRINT ”    READ DISCRETE RESULTS PART 1”
68 PRINT ”    DISPLAY THE DISCRETE RESULTS PART 1”
69 PRINT ”    DISPLAY THE FAILED WINDOWS 1–4”
70 PRINT
71 PRINT ”THIS PROGRAM WAS USED AND TESTED ON AN ALLEN–BRADLEY”
72 PRINT ”    T47 COMPUTER WITH GWBASIC.EXE FROM MS–DOS VER 4.01”
73 PRINT
74 PRINT ”THE COLOR CVIM COMMUNICATION PARAMETERS FOR THIS PROGRAM ARE:”
75 PRINT ”    CFG HOST: RS232 A”
76 PRINT ”    SYS HOST: RS232 A”
77 PRINT ”    TRIGGER SOURCE = (HOSTED)”
78 PRINT ”    RS232 A PROTOCOL = ASCII”
79 PRINT ”    RS232 A BAUD RATE = 9600”
80 PRINT ”    COLOR CVIM MUST BE IN RUNMODE”
90 PRINT
99 :
100 PRINT:INPUT ”ENTER 1 TO RUN PROGRAM AND 2 TO EXIT PROGRAM”;PROCESS
105 PRINT
110 IF PROCESS = 1 THEN GOTO 130
120 SYSTEM
130 GOSUB 2000: REM Read tool set results to get # of triggers processed
140 IF R1(0)<0 THEN 100 ELSE NT = R1(0)
150 GOSUB 1000: REM Trigger an inspection
200 GOSUB 2000: IF R1(0)=NT THEN 200: REM Read until the trigger is processed
210 GOSUB 2500: REM Convert hex result string RE$ to integers
220 IF R1(0) <0 THEN 100: REM Quit on input error
230 PRINT: FOR X=0 TO 17: PRINT R1(X),: NEXT: PRINT: REM Print results
240 IF R1(4) AND 2 THEN PRINT ”Window 1 FAIL”
250 IF R1(4) AND 8 THEN PRINT ”Window 2 FAIL”
260 IF R1(4) AND 32 THEN PRINT ”Window 3 FAIL”
270 IF R1(4) AND 128 THEN PRINT ”Window 4 FAIL”
400 GOTO 100
```

```
999 :
1000 REM Subroutine to trigger an inspection on tool set
1050 PRINT#1,”>t”; CHR$(13);: REM Send the command
1080 RETURN
1999 :
2000 REM  Subroutine to read discrete results from tool set
2040 IF LOC(1) THEN R$=INPUT$(LOC(1),#1): REM clear out any garbage characters
2050 PRINT#1,”>rr,p1”; CHR$(13);: REM Send the command
2060 CR$=INPUT$(2,#1): REM get CR/LF or ?/CR
2070 IF CR$=CHR$(13)+CHR$(10) THEN 2090
2080 PRINT CR[0]:PRINT”Input error”: R$=INPUT$(LOC(1),#1): R1(0)=–1: RETURN
2090 R$=INPUT$(1,#1): IF ASC(R$)<32 THEN 2090: REM ignore junk
2100 LINE INPUT#1,RE$: RE$=R$+RE$: REM get entire response
2120 R1(0) = VAL(MID$(RE$,1,9)): R1(1) = VAL(MID$(RE$,10,9))
2130 R$=INPUT$(LOC(1),#1): RETURN: REM Clear out any remaining characters
2499 :
2500 REM Subroutine to convert hex values in discrete result string RE$
2501 REM to integer values
2510 FOR RN=0 TO 15
2515 REM The following line converts each pair of hex digits to an integer
2520 D1=INSTR(HE$,MID$(RE$,25+RN*3,1))–1: D2=INSTR(HE$,MID$(RE$,26+RN*3,1))–1
2530 R1(RN+2) = 16*D1+D2: NEXT RN: RETURN
```

DF1 Protocol

The remainder of this chapter describes the Color CVIM implementation of the DF1 protocol. After you have made the equipment connections and configured the Color CVIM module for RS–232 communications, DF1 packets of data can be sent to the Color CVIM module.

What is DF1?

DF1 is an Allen–Bradley developed software convention used for RS–232 communications. DF1 provides some handshaking and data–packing formats which allow for fast communications with integrity of the data. The Color CVIM uses only the Data Link Layer of DF1.

This chapter describes a simple application level of DF1 for communications between a Color CVIM module and a computer host. This application level requires that all transmitted data be preceded by a header and terminated by a trailer and a Block Check Character (BCC). In addition, ACK / NAK characters and simple time out conventions are used to ensure the integrity of the data. This is called the Data Link Layer.

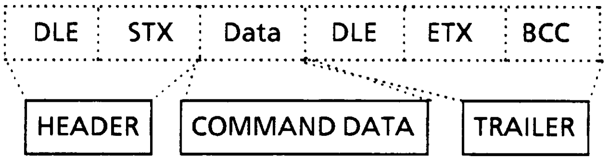
We do not provide a complete description of DF1 in this manual. We have only provided the information necessary to transmit data between a host computer and the Color CVIM module. If you want to learn more about DF1, we suggest reading Publication 1707–6.5.16.

DF1 Character Set

In the DF1 protocol mode, all data is transferred between the Color CVIM module and a host as bytes with a value between 00(hex) and FF(hex). Refer to Appendix E to convert control codes like ACK and NAK to/from hexadecimal values.

Command Structure

Each command the host device sends to the Color CVIM module is represented by a block of data beginning with DLE STX (Data Link Escape, Start of Transmission) and terminated with DLE ETX BCC (Data Link Escape, End Transmission, Block Check Character). The data between the header and trailer characters is the command data. The following shows the structure of a typical command:



Where: DLE = 10(hex)
STX = 02(hex)
ETX = 03(hex)

Note: To avoid any confusion between DLE (10 hex) and data equal to 10 (hex), a value of 10(hex) is transmitted as 10(hex) 10(hex). The DLE code is transmitted simply as 10 (hex). This is referred to as “DLE Stuffing”.

The following shows the typical structure of the command data.

OPERATION	n times (H)	n times (L)	Object	Flags	Data

There are up to five fields in a command:

- Operation Field — This field contains the command directed to the Color CVIM module. There can only be one operation per command line. Some commands don’t require any additional fields while others may require an object field,a data field, or both. Some commands cannot be used while the Color CVIM module is in the SETUP mode. If an operation cannot be performed because either the wrong host is selected or the Color CVIM module is in the SETUP mode, the Color CVIM module will not send a response.

- **n** times (H) and **n** times (L) — These two fields indicate the High and Low bytes of the n times modifier. The n times modifier is used with certain commands to indicate the number of times the command is to be performed. The range for this value is 0000 to 00FF (255). A value of 0000 indicates infinity. The default value for this field is 0001.

Note: The number of times fields are optional.

Command Structure (cont'd)

- Object Field — The Object field specifies data that configures the operation of the Color CVIM module.

  In the description of each command we specify the objects that can entered into a command.

- Flags — This optional field specifies outputs on the local I/O board or specific blocks of data.
- Data Field — Contains data.

ACK/NAK, BCC Characters

After receiving a DF1 data packet, the Color CVIM module validates the Block Check Character.

Note: The block check character is a technique used to check the integrity of of data packet. BCC are explained in the next section.

Depending upon whether or not the BCC is validated, the following will occur:

- If the BCC is not acceptable, the Color CVIM module will reply with a DLE NAK (Negative Acknowledgment) character and discard the data packet.
- If the command has an acceptable BCC the Color CVIM module will reply with a DLE ACK (Positive Acknowledgment) character and try to execute the command.

After receiving a data packet and validating the BCC, one of the following will occur.

- If data packet has a valid BCC but the Color CVIM module cannot execute the command the Color CVIM module will discard the data package. No message is returned. The host should be set to time out after waiting for a response.
- If the command can be executed, the Color CVIM module will respond with any returned data packets.

ACK/NAK, BCC Characters (cont’d)

After receiving the data, the host should respond with a DLE ACK to let the Color CVIM module know that the message was received properly. If the host returns a DLE NAK, the Color CVIM module will retransmit the data up to three times before discarding the data packet.

Note: Some commands request a continuous flow of data from the Color CVIM module. You can stop the flow of data by sending another command.

Note: A simple way to test the RS–232 links is to send the Color CVIM module a DLE ENQ (enquiry). If you have the port properly connected and the Color CVIM module is configured for RS–232, the Color CVIM module should send a DLE ACK or DLE NAK in response. If no response is provided, check your connections and Color CVIM module configuration.

Block Check Character

The block check character (BCC) is a means of checking the accuracy of each message packet transmission. It is the 2’s complement of the 8–bit sum (modulo–256 arithmetic sum) of all data bytes between the DLE STX and the DLE ETX BCC. It does not include any other message packet codes or response codes.

For example, if a message packet contained the data codes 8, 9, 6, 0, 2, 4, and 3, the message packet codes would be (in hex):

$\underbrace{10\quad 02\quad 08\quad 09\quad 06}$ 00 02 04 03 10 03 E0

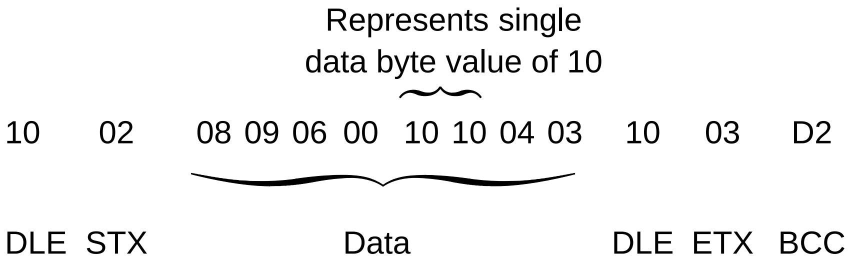
DLE STX Data DLE ETX BCC

The sum of the data bytes in this message packet is 20 hex. The BCC is the 2’s complement of this sum, or E0 hex. This is shown in the following binary calculation:

```
0010          000020 hex
1101          11111’s complement
              +1
__________
1110          00002’s complement (E0 hex)
```

Block Check Character (cont'd)

To transmit the data value 10 hex, you must use the data code DLE DLE. However, only one of these DLE data bytes is included in the BCC sum. For example, to transmit the values 8, 9, 6, 0, 10, 4, and 3 hex, you would use the following message codes:



In this case, the sum of the data bytes is 2E hex because only one DLE text code is included in the BCC. So the BCC is D2 hex.

The BCC algorithm provides a medium level of data security. It cannot detect transposition of bytes during transmission of a packet. It also cannot detect the insertion or deletion of data values of zero within a packet.

Deactivate Forces

Use the deactivate force command to return outputs on the 2801–JMB local I/O board to the Color CVIM module assigned functions. The deactivate forces command is:

0C

If the BCC is not valid, the Color CVIM module will respond with a DLE NAK and the command will not be executed.

If the BCC is valid, the Color CVIM module will respond with a DLE ACK. Then the Color CVIM module will validate the command structure. If the command is valid, the Color CVIM module will execute the command and return the data. If the command structure is invalid, the Color CVIM module will not execute the command or respond.

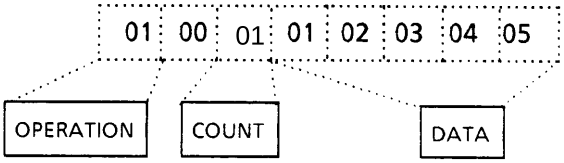
Echoing Data

Use the echo command to check the communications link. This command will return the same same string of characters that are sent out with the command. This command has the following structure:

01	00	n times	Data

Where n times specifies the number of times the Color CVIM module will echo the data field back to the host device. There is no object associated with this command. The command is valid at any time.

For example:



This example will cause the Color CVIM module to return the string:
DLE ACK DLE STX 0102030405 DLE ETX BCC

If the BCC is not valid, the Color CVIM module will respond with a DLE NAK and the command will not be executed.

If the BCC is valid, the Color CVIM module will respond with a DLE ACK. Then the Color CVIM module will validate the command structure. If the command is valid, the Color CVIM module will execute the command and echo the data. If the command structure is invalid, the Color CVIM module will not execute the command or respond.

Enable/Disable Outputs

Use this command to enable or disable discrete outputs or local *Outputs* I/O. Use the following commands:

15	*This command disables the outputs.*
14	*This command enables the outputs.*

If the BCC is not valid, the Color CVIM module will respond with a DLE NAK and the command will not be executed.

If the BCC is valid, the Color CVIM module will respond with a DLE ACK. Then the Color CVIM module will validate the command structure. If the command is valid, the Color CVIM module will execute the command. If the command structure is invalid, the Color CVIM module will not execute the command or respond.

Forcing Local I/O

Use the force command to turn the local I/O outputs either on or off. This function can only be executed once per command. The only valid objects are the fourteen discrete outputs. This command has the following structure:

02	06	Flags	Data

Where the flags specify outputs 1 through 14:

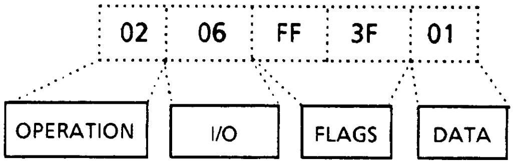
Byte 3 Bits 7 → 0								Byte 4 Bits 7 → 0							
8	7	6	5	4	3	2	1	X	X	14	13	12	11	10	9

Set individual Bits to select outputs 1 through 14

The data in this command indicates on or off (1 = ON and 0 = OFF).

Forcing Local I/O (cont'd)

For example:



This example will force all outputs on. FF sets all bits in byte 3 (outputs 1 though 8) and 3F sets bits 0 through 5 of byte 4 (outputs 9 through 14).

It is possible to have outputs forced on and off at the same time. A force off takes precedence over the force on. If multiple force commands are sent, the forced on or off outputs will be added to those already forced.

If the BCC is not valid, the Color CVIM module will respond with a DLE NAK and the command will not be executed.

If the BCC is valid, the Color CVIM module will respond with a DLE ACK. Then the Color CVIM module will validate the command structure. If the command is valid, the Color CVIM module will execute the command. If the command structure is invalid, the Color CVIM module will not execute the command or respond.

Note: The outputs will remain in their forced states until a deactivate forces command is sent.

Loading Configurations

Use the load command to transfer configuration data between the Color CVIM module local storage area (EEPROM) and the external memory card. The RAM card slides into the slot on the front of the Color CVIM module. Use one of the following commands:

03	00

Transfers configuration from the EEPROM to the Color CVIM module internal RAM.

03	01	01

Transfers memory from RAM Card area 1 to the Color CVIM module internal RAM.

03	01	02

Transfers memory from RAM Card area 2 to the Color CVIM module internal RAM.

Note: Memory cards are laid out in slots. A 64K card (2801–MD4) has room for two configurations. The 512K memory card (2801–MD5) has 16 slots, and can store up to 16 configurations. An image must be one of the first two items stored to the 512K card, and can only occupy 12 contiguous free slots.

You cannot use these commands when the Color CVIM module is in the SETUP mode.

If the BCC is not valid, the Color CVIM module will respond with a DLE NAK and the command will not be executed.

If the BCC is valid, the Color CVIM module will respond with a DLE ACK. Then the Color CVIM module will validate the command structure. If the command is valid, the Color CVIM module will execute the command. If the command structure is invalid, the Color CVIM module will not execute the command or respond.

Lock Command

Use the lock command to disable the setup menu box so that the SETUP mode cannot be entered. This function can only be executed once per command. There is no object associated with this command. The command has the following structure:

04

If the BCC is not valid, the Color CVIM module will respond with a DLE NAK and the command will not be executed.

If the BCC is valid, the Color CVIM module will respond with a DLE ACK. Then the Color CVIM module will validate the command structure. If the command is valid, the Color CVIM module will execute the command. If the command structure is invalid, the Color CVIM module will not execute the command or respond.

Read Output Status

Use the read command to read the status of the 14 discrete outputs on the local I/O board The command has the following structure:

05	00	n times	06	Flags

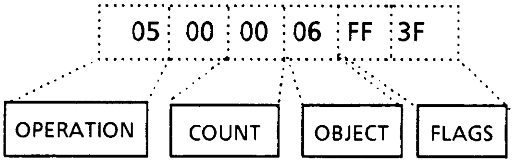
This function can be executed more than once per command by specifying an n times value.

The flags specify outputs 1 through 14:

Byte 3 Bits 7 → 0								Byte 4 Bits 7 → 0							
8	7	6	5	4	3	2	1	X	X	14	13	12	11	10	9

Set individual bits to select outputs 1 through 14

For Example:



Read Output Status (cont'd)

This example will read the status of all fourteen outputs. FF sets all bits in byte 3 (outputs 1 though 8) and 3F sets bits 0 through 5 of byte 4 (outputs 9 through 14).

One byte is returned to indicate the status of the output (1 = ON and 0 = OFF). The output bytes are transmitted in numerical order (output #1 then output #2, etc.). The amount of data returned depends upon the number of outputs being read.

If the BCC is not valid, the Color CVIM module will respond with a DLE NAK and the command will not be executed.

If the BCC is valid, the Color CVIM module will respond with a DLE ACK. Then the Color CVIM module will validate the command structure. If the command is valid, the Color CVIM module will execute the command and return the data. If the command structure is invalid, the Color CVIM module will not execute the command or respond.

Read Configuration Block Command

Use the read configuration command to read configuration data for the specified object. The command has the following structure:

06	07	Flags (27 Bytes)

This function can only be executed once per command. The only valid object for this command are the configuration blocks.

The flags indicate which of the 213 configuration blocks are going to be read. Set the bits in bytes 3 through 29 of the command to specify the block(s). Use the following chart to determine which bits to set:

Read Configuration Block Command (cont'd)

Use the read configuration command to read configuration data for the specified object. The command has the following structure:

Figure 5.8
Read Configuration Block Command Bytes 3–29

Byte 3
Bits 7 – 0

8	7	6	5	4	3	2	1

Byte 4
Bits 7 – 0

16	15	14	13	12	11	10	9

Byte 5
Bits 7 – 0

24	23	22	21	20	19	18	17

Byte 6
Bits 7 – 0

32	31	30	29	28	27	26	25

Byte 7
Bits 7 – 0

40	39	38	37	36	35	34	33

Byte 8
Bits 7 – 0

48	47	46	45	44	43	42	41

Byte 9
Bits 7 – 0

56	55	54	53	52	51	50	49

Byte 10
Bits 7 – 0

64	63	62	61	60	59	58	57

Byte 11
Bits 7 – 0

72	71	70	69	68	67	66	65

Byte 12
Bits 7 – 0

80	79	78	77	76	75	74	73

Byte 13
Bits 7 – 0

88	87	86	85	84	83	82	81

Byte 14
Bits 7 – 0

96	95	94	93	92	91	90	89

Byte 15
Bits 7 – 0

104	103	102	101	100	99	98	97

Byte 16
Bits 7 – 0

112	111	110	109	108	107	106	105

Byte 17
Bits 7 – 0

120	119	118	117	116	115	114	113

Byte 18
Bits 7 – 0

128	127	126	125	124	123	122	121

Byte 19
Bits 7 – 0

136	135	134	133	132	131	130	129

Byte 20
Bits 7 – 0

144	143	142	141	140	139	138	137

Byte 21
Bits 7 – 0

152	151	150	149	148	147	146	145

Byte 22
Bits 7 – 0

160	159	158	157	156	155	154	153

Byte 23
Bits 7 – 0

168	167	166	165	164	163	162	161

Byte 24
Bits 7 – 0

176	175	174	173	172	171	170	169

Byte 25
Bits 7 – 0

184	183	182	181	180	179	178	177

Byte 26
Bits 7 – 0

192	191	190	189	188	187	186	185

Byte 27
Bits 7 – 0

200	199	198	197	196	195	194	193

Byte 28
Bits 7 – 0

208	207	206	205	204	203	202	201

Byte 29
Bits 7 – 0

			213	212	211	210	209

Set individual bits to select blocks 1 through 213.

Read Configuration Block Command (cont'd)

For example: To read configuration blocks 49 and 50 you would send:

06(hex) for byte 1– Indicates a read command.

07(hex) for byte 2– Specifies the configuration blocks.

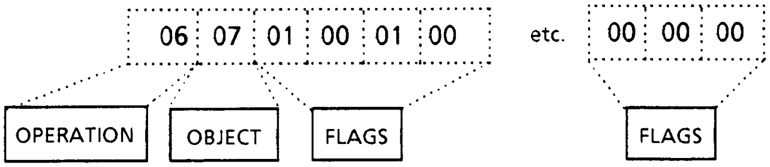
00(hex) for bytes 4 through 8.

03(hex) for byte 9– Sets the first two bits of byte 9 to indicate blocks 49 and 50.

00(hex) for bytes 10 through 29.

Refer to Appendix C for a description of the configuration blocks. You cannot use this command while the Color CVIM module is in the SETUP mode.

Example:



This example reads configuration blocks 1 and 17.

If the BCC is not valid, the Color CVIM module will respond with a DLE NAK and the command will not be executed.

If the BCC is valid, the Color CVIM module will respond with a DLE ACK. Then the Color CVIM module will validate the command structure. If the command is valid, the Color CVIM module will execute the command and return the data. If the command structure is invalid, the Color CVIM module will not execute the command or respond.

After reading the selected blocks, the Color CVIM module will return the requested data. Each word of a configuration block is sent as two bytes with the high byte transmitted first. A DLE (10 hex) is converted to DLE DLE (10 hex 10 hex). Refer to Appendix D for block descriptions and sizes. The following is an example of how the returned data appears (each pair of digits represents a single byte):

Read Configuration Block Command (cont'd)

Figure 5.9
Configuration Block Returned Data Format*

```
48 01 02 00 00 00 00 00 44 65 66 61 75 6C 74 00 00 00 00 00
00 00 00 00 00 03 00 00 00 00 00 00 04 00 00 01 00 01 00 00
00 02 06 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 06 01
00 00 00 01 00 00 00 00 00 00 00 00 00 00 00 01 01 00 01 01
00 0D FC 5F 9A 0A 00 19 FB D1

48 02 FF 02 01 00 00 00 00 FA 00 10 00 01 01 00 00 00 00 00
00 00 00 3F 00 00 00 00 00 00 00 3F 00 00 00 00 00 00 00 64
00 32 01 2C 00 96 0A 62 00 01 00 00 00 00 00 00 00 00 00 64
00 01 00 00 00 01 00 00 00 01 00 00 00 64 00 32 01 2C 00 96
0A 62 00 01 00 00 00 01 00 00 00 00 80 00 00 00 00 00 00 00
00 00 00 00 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00 01
00 00 00 00 00 00 49 CC
```

*The example above is valid as shown for DF1 Return Data mode.
In ASCII mode there are no spaces.

Read Results Command

Use this command to read the *results of the last inspection.* Refer to Appendix C for a description of the results blocks. Use one of the following commands:

07	00	n times	x	z

Where:

07 =	*Read Command*
00 =	*Always 00*
n times =	*Value for repeat read*
x =	*00 (Specifies All Discrete Bit Results)* *04 (Specifies Bit Besults First Part)* *05 (Specifies Bit Results Second Part)* *08 (Specifies Color CVIM module status)* *10 (Specifies Results Block)* *16 (Specifies Gage)* *17 (Specifies Window)* *18 (Specifies Reference Line)* *19 (Specifies Reference Window)* *1A (Specifies Light Probe)*
z =	*Number of window or gage being read or block result number.*

07	00	n times	04

Use this command to read the discrete bit results first part.

07	00	n times	05

Use this command to read the discrete bit results second part.

07	00	n times	10	Data

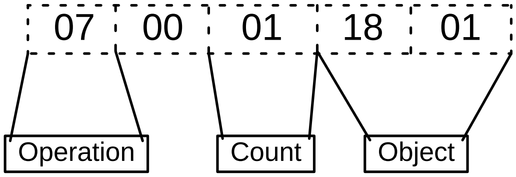
Use this command to read results blocks.

Where data results block number.

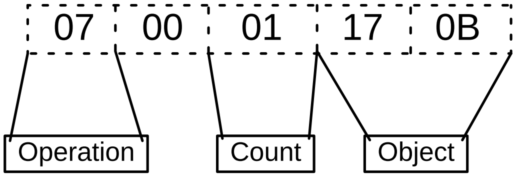
Read Results Command (cont’d)

The read results command can be executed more than once per command by specifying an n times value.

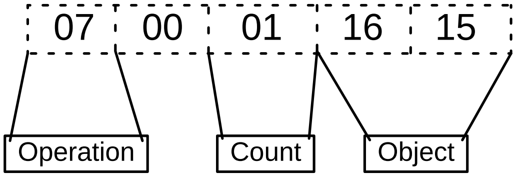
Examples:



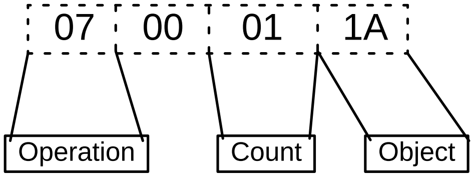
This command reads the results reference line #1 (4 data bytes returned)



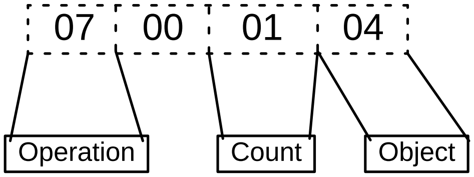
This command reads the results of window #11 (4 data bytes returned).



This command reads the results of gage #21 (4 data bytes returned).



This command reads the results of light probe (12 data bytes returned).



This command reads the first part of the discrete bit results once.

Read Results Command (cont'd)

This command reads the Color CVIM module status (2 data bytes returned).

Note: Refer to Appendix B, Table B.1, RS–232 word 0 for a definition of Color CVIM module status.

If the BCC is not valid, the Color CVIM module will respond with a DLE NAK and the command will not be executed.

If the BCC is valid, the Color CVIM module will respond with a DLE ACK. Then the Color CVIM module will validate the command structure. If the command is valid, the Color CVIM module will execute the command and return the data. If the command structure is invalid, the Color CVIM module will not execute the command or respond.

The format of the requested data is an ASCII representation of the specified block(s) in bytes.

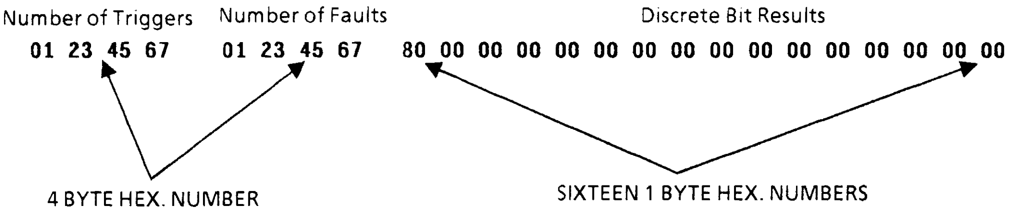
Figure 5.10
Results Block Returned Data Format*

```
61 01 04 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 32 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 CC
```

Refer to Appendix C for Block Descriptions

* The example above is valid as shown for DF1 Return Data mode. In ASCII mode there are no spaces.

Figure 5.11
Discrete Bit Results Returned Data Format*



Refer to Appendix B for a description of Returned Bytes

*We have added spaces for clarity; data is returned without spacing.

Save Command

Use the Save command to save Color CVIM module configuration data to the local storage area (EEPROM) or the external RAM card (credit card memory).

Note: Depending upon the card size, up to 16 configurations can be saved to the RAM card (512K card).

Use the following commands:

08	00

Saves configuration to EEPROM.

08	01	XX

Where XX = card storage location (01 to 16).

This function can only be executed once per command. You cannot use the commands when the Color CVIM module is in the SETUP mode.

If the BCC is not valid, the Color CVIM module will respond with a DLE NAK and the command will not be executed.

If the BCC is valid, the Color CVIM module will respond with a DLE ACK. Then the Color CVIM module will validate the command structure. If the command is valid, the Color CVIM module will execute the command. If the command structure is invalid, the Color CVIM module will not execute the command or respond.

Select Image Displayed

Use the display object command to control the image that is displayed on the monitor. Use the following commands:

0B	02	Data

The data in this command specifies the display to be viewed.

01 (hex) = Image only displayed
02 (hex) = Failed tools displayed
03 (hex) = All tools displayed
04 (hex) = I/O page displayed
05 (hex) = Results page displayed
06 (hex) = Stats 1 page displayed
07 (hex) = Stats 2 page displayed
08 (hex) = Page up same display
09 (hex) = Page down same display

0B	13	Data

The data in this command specifies the display to be viewed.

01 (hex) = Go on reject
02 (hex) = Freeze on first reject
03 (hex) = Freeze on all reject
04 (hex) = Freeze on next inspection
05 (hex) = Halt on reject

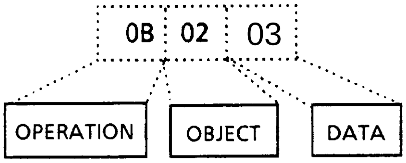
0B	14	Data

The data in this command specifies the display to be viewed.

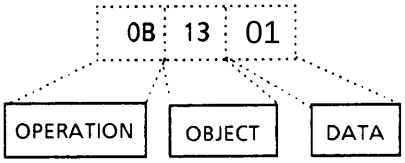
01 (hex) = Resume
02 (hex) = Reset Statistics
03 (hex) = Reset counters
04 (hex) = Page up
05 (hex) = Page down

Select Image Displayed (cont'd)

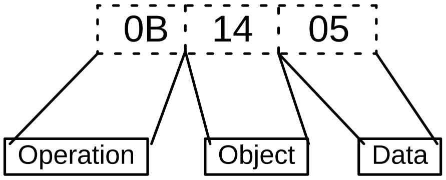
Examples:



This example displays all tools



This example selects go on reject



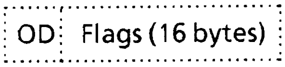
This example selects page down

If the BCC is not valid, the Color CVIM module will respond with a DLE NAK and the command will not be executed.

If the BCC is valid, the Color CVIM module will respond with a DLE ACK. Then the Color CVIM module will validate the command structure. If the command is valid, the Color CVIM module will execute the command and return the data. If the command structure is invalid, the Color CVIM module will not execute the command or respond.

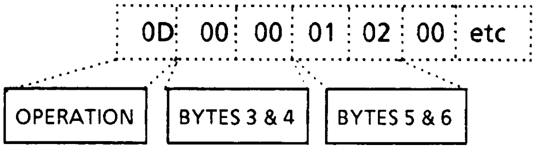
Set Configurable Results

Use this command to obtain a configurable results block. The results you want are specified by a list of tools and placed in results block #6. No data is returned until you use a read inspection results command for block #6. Use the following command:



Refer to Table C.7 in Appendix C. Set the bits to 1 for the tools you want.

Example:



This command configures results block #6 to contain Window #9 and Window #18 data.

If the BCC is not valid, the Color CVIM module will respond with a DLE NAK and the command will not be executed.

If the BCC is valid, the Color CVIM module will respond with a DLE ACK. Then the Color CVIM module will validate the command structure. If the command is valid, the Color CVIM module will execute the command and return the data. If the command structure is invalid, the Color CVIM module will not execute the command or respond.

Note: The set configurable results command only sets the contents of the configurable results block (block #6). You must use a read results command to obtain the data (07 00 nn 10 06, *where nn specify the number of times the command is performed).* Refer to Read Inspection Results command for the format of the returned data.

The returned results bloc will be 128 bytes including the block signature (2 bytes) and the total number of triggers (last 4 bytes). Refer to page C–16. The ordering of the tools and data lengths are the same as the Remote I/O configurable results block.

Set/Read Configurable Statistics

Use the read command to read statistical data for the light probe, reference windows, gages, and windows. Use the separate set command to set the number of samples and configure the statistics block. **Remember that only the system (SYS) host can change the sample numbers.**

The set configurable statistics command has the following structure:

16	00	nn	Flags (16 Bytes)

Where nn specifies the number of samples.

Note: If nn is 00, the Color CVIM module will continue to use the sample count configured during setup. Any other value will change the sample count.

The flags indicate which items are placed in the statistics block. Refer to Table C.5 in Appendix C. Set the bits to 1 for the statistics you need.

The read configurable statistics command has the following structure:

17	00	n times

Where nn specifies the number of times the statistics block is read.

Statistics are accumulated until the number of samples is reached, at which point the statistics begin to reaccumulate.

Example of Set Command:

16	00	32	Flags (16 Bytes)

This example sets the number of samples to 50.

Example of Read Command:

17	00	05

This example reads the statistics block five times.

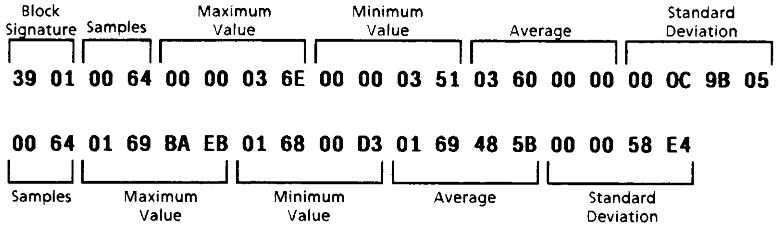
Set/Read Configurable Statistics (cont'd)

The data returned from the statistics block consists of:

- Block signature
- Number of samples, maximum, minimum, average, and standard deviation for each tool configured in the block.

The block signature is 2 bytes long. The number of samples is a 2 byte integer. The maximum and minimum values are each 4 bytes. The format of the data depends upon the operation (e.g. pixel count is an integer and linear gaging is a 16.16 fixed point value). Refer to page C–24 for data formats. The average and standard deviation are also 4 bytes each but are always 16.16 fixed point values. Therefore, each tool statistic consists of 18 bytes with the exception of reference windows and the light probe, which contain 18 bytes for each feature or a total of 54 bytes. The statistics block is transmitted as two hexadecimal characters for each byte. The total number of bytes including the block signature should not exceed 128 bytes. The statistics block is read once for every number of specified samples. This means that if you read the statistics block five times with a sample number of 50, 250 triggers will have to be processed before the five reads are completed. The following shows the format of the returned data:

Figure 5.12
Statistics Block Returned Data Format*



```
39 01 00 64 00 00 03 6E 00 00 03 51 03 60 00 00 00 0C 9B 05

00 64 01 69 BA EB 01 68 00 D3 01 69 48 5B 00 00 58 E4
```

*The example above is valid as shown for DF1 Return Data mode. In ASCII mode there are no spaces.

Trigger Operation Command

Use the trigger operation command to initiate an inspection. Use the following commands:

09

Triggers an inspection

This function can only be executed once per command.

If the BCC is not valid, the Color CVIM module will respond with a DLE NAK and the command will not be executed.

If the BCC is valid, the Color CVIM module will respond with a DLE ACK. Then the Color CVIM module will validate the command structure. If the command is valid, the Color CVIM module will execute the command and return the data. If the command structure is invalid, the Color CVIM module will not execute the command or respond.

Note: When using this command you should make sure that the Color CVIM module is configured for a “hosted trigger source”.

Unlock Command

Use the unlock command to enable the user interface (monitor and keyboard) so that a user can access the SETUP. This command has the following structure:

0A

If the BCC is not valid, the Color CVIM module will respond with a DLE NAK and the command will not be executed.

If the BCC is valid, the Color CVIM module will respond with a DLE ACK. Then the Color CVIM module will validate the command structure. If the command is valid, the Color CVIM module will execute the command. If the command structure is invalid, the Color CVIM module will not execute the command or respond.

Write Configuration Blocks

Use the write command to write data to configuration blocks.

Note: You can also write to the monitor display to select what is displayed. Refer to Select Image Displayed.

This command has the following structure:

0B	07	Flags

Where the flags specify the configuration blocks being written to. There are 213 configuration blocks. Set the bits in bytes 3 through 29 to indicate which blocks. Refer to the following diagram.

Write Configuration Blocks (cont’d)

Figure 5.13
Write Configuration Block Command Bytes 3–29

Byte 3
Bits 7 – 0

8	7	6	5	4	3	2	1

Byte 4
Bits 7 – 0

16	15	14	13	12	11	10	9

Byte 5
Bits 7 – 0

24	23	22	21	20	19	18	17

Byte 6
Bits 7 – 0

32	31	30	29	28	27	26	25

Byte 7
Bits 7 – 0

40	39	38	37	36	35	34	33

Byte 8
Bits 7 – 0

48	47	46	45	44	43	42	41

Byte 9
Bits 7 – 0

56	55	54	53	52	51	50	49

Byte 10
Bits 7 – 0

64	63	62	61	60	59	58	57

Byte 11
Bits 7 – 0

72	71	70	69	68	67	66	65

Byte 12
Bits 7 – 0

80	79	78	77	76	75	74	73

Byte 13
Bits 7 – 0

88	87	86	85	84	83	82	81

Byte 14
Bits 7 – 0

96	95	94	93	92	91	90	89

Byte 15
Bits 7 – 0

104	103	102	101	100	99	98	97

Byte 16
Bits 7 – 0

112	111	110	109	108	107	106	105

Byte 17
Bits 7 – 0

120	119	118	117	116	115	114	113

Byte 18
Bits 7 – 0

128	127	126	125	124	123	122	121

Byte 19
Bits 7 – 0

136	135	134	133	132	131	130	129

Byte 20
Bits 7 – 0

144	143	142	141	140	139	138	137

Byte 21
Bits 7 – 0

152	151	150	149	148	147	146	145

Byte 22
Bits 7 – 0

160	159	158	157	156	155	154	153

Byte 23
Bits 7 – 0

168	167	166	165	164	163	162	161

Byte 24
Bits 7 – 0

176	175	174	173	172	171	170	169

Byte 25
Bits 7 – 0

184	183	182	181	180	179	178	177

Byte 26
Bits 7 – 0

192	191	190	189	188	187	186	185

Byte 27
Bits 7 – 0

200	199	198	197	196	195	194	193

Byte 28
Bits 7 – 0

208	207	206	205	204	203	202	201

Byte 29
Bits 7 – 0

			213	212	211	210	209

Set individual bits to select blocks 1 through 213.

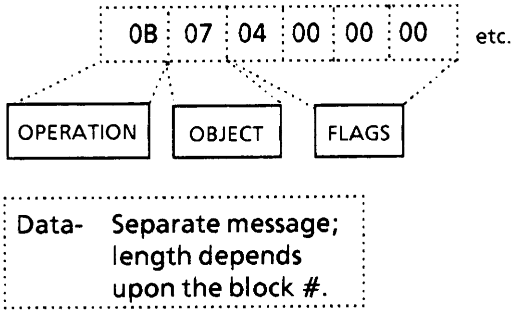
Write Configuration Blocks (cont'd)

This function can only be executed once per command.

Each configuration block is sent in a separate data packet (Header/Data/Trailer).

Refer to Appendix D for a description of the configuration blocks. You cannot use this command when the Color CVIM module is in the setup mode. After writing to the Color CVIM module, the Color CVIM module will validate all of the configuration blocks (refer to Chapter 4 for a description of memory validation).

Example:



This example writes the data into configuration block #3.

If the BCC is not valid, the Color CVIM module will respond with a DLE NAK and the command will not be executed.

If the BCC is valid, the Color CVIM module will respond with a DLE ACK. Then the Color CVIM module will validate the command structure. If the command is valid, the Color CVIM module will execute the command. If the command structure is invalid, the Color CVIM module will not execute the command or respond.

Note: We recommend that you first read the Module Status word to make sure the Module Busy bit is inactive (0). After writing the blocks, check the discrete bit "Configuration Fault" after loading a configuration. Refer to Appendix B.1 Since no data is returned with this command, the load operation may fail and you could detect this by checking the fault bit. No other ports should be active.

Command Summary

After you have become familiar with the DF1 commands, you can use the following command summary as a quick reference guide.

Table 5.B
DF1 Command Summary

Command	Command Structure	Field Descriptions
Deactivate Forces	0C	

Echo Data	01 00 nn Data	nn = times repeated
		Data = Hexadecimal string

Enable Outputs	14	
Disable Outputs	15	
Force Outputs	02 06 Flags Flags Data	Flags indicate outputs Data = 00 (on) or 01 (off)
Load Configuration From EEPROM to RAM	03 00	

Load Configuration From Credit Card to RAM*	03 01 Data	Data = 01 (area 1)
		02 (area 2)
		● ● ●
		15(area 15)
		16(area 16)

Lock	04	
Unlock	0A	
Read Output Condition	05 00 nn 06 Flags Flags	nn = times repeated Flags indicate outputs
Read Configuration	06 07 Flags	Flags indicate blocks
Read Discrete Bit Results	07 00 nn 04 07 00 nn 05	First discrete bit results Second discrete bit results nn = times repeated

* The number of configurations that can be stored on a RAM card depends upon the card size (512K card can hold 16 configurations).

Command Summary (cont‘d)

Table 5.B
DF1 Command Summary (Cont’d)

Command	Command Structure	Field Descriptions
Read Results Block	07 00 n y z	n = times repeated y = 16(Gages) 17 (Window) 18 (Reference Line) 19 (Reference Window 1A (Light Probe) z = Window, Gage, Reference Line, or Reference Window No.

	07 00 n 10 Data	Block Results
		n = times repeated
		Data = Block Number(s) 1, 2, 3, 4, 5 or 6

	07 00 n y	y = 00 (All discrete bit results) 04 (First part of dis–crete bit results) 05 (Second part of discrete bit results) n = times repeated
	07 00 n 08	Read Status
Save to EEPROM from RAM	08 00	

Save to RAM Card from RAM	08 01 Data	Data = 01 (area 1)
		02 (area 2)
		● ● ●
		15(area 15)
		16(area 16)

Set Configurable Results	OD Flags	Flags indicate specific tools
Set Configurable Statistics	16 00 n Flags	Flags indicate specific tools n = times repeated
Read Configurable Statistics	17 00 n	n = times repeated
Trigger Inspection	09	Trigger
Write Configuration Block	0B 07 Flags Data	Flags indicate blocks

Command Summary (cont‘d)

Table 5.B
DF1 Command Summary (Cont’d)

Command	Command Structure	Field Descriptions
Write Display	0B 02 Data	Data = 01–Image Only 02–Failed Tools 03–All Tools 04–I/O Page 05–Results Page 06–Stats 1 Page 07–Stats 2 Page 08–Page Up Same Display 09–Page Down Same Display
	OB 13 Data	Data = 01–Go On Reject 02–Freeze On First Reject 03–Freeze On All Reject 04–Freeze On Next Inspection 05–Halt On Reject
	OB 14 Data	Data = 01–Resume 02–Reset Statistics 03–Reset Counters 04–Page Up 05–Page Down

DF1 Programming Example

The following is a sample DF1 program written in Microsoft C, version 6.0. The program configures the host computer’s serial port for 9600 Baud communications. The program displays a description and related material, then asks you to select either COM 1 or COM 2. Next you’ll see a menu which prompts you to select one of the following operations:

1 Echo the word “HELLO” to test the communications port.
2 Trigger the Color CVIM module to perform an inspection.
3 Read the discrete bit results.
4 Read Results Block #1.
5 Read configuration blocks 1–213.
6 Write to configuration blocks 1–213.

Note: You must first read the configuration before you write it because it is stored in volatile memory (RAM).

The program demonstrates a simple implementation of the DF1 protocol. Your actual application program may require some enhancements such as increased error checking or time–out conventions.

The sample program begins on the next page.

DF1 Programming Example (cont’d)

```
/* This sample program was Compiled using Microsoft C Verision 6.0    */
/* Color CVIM RS–232 Comminication example program using DF1 protocol */
/* Copyright Allen–Bradley  09–10–92 jrm, aes, dms                    */

#include <stdio.h>
#include <stdlib.h>
#include <bios.h>

#define STX 0x02  /* Start of Text character */
#define ETX 0x03  /* End of Text character */
#define DLE 0x10  /* Data Link Escape (Control char) */
#define ACK 0x06  /* Positive acknowledgement */
#define NAK 0x15  /* Negative acknowledgement */

#define COMMFLAGS 0x8000  /* defines bits to check for comm error */

unsigned char config[213][128];  /* define storage for configuration data
*/
unsigned char configlen[213];  /* define storage for configuration length
*/

unsigned char last_response = 0;  /* define the last responce counter */

void main()
    {
    int x, op_num, portnum, err, replen, reslen, numblocks;
    unsigned char reply[200], results[128];

    /* Display Program Execution and Description. */
    printf (”\nColor CVIM RS–232 Communication example program using ”);
    printf (”DF1 protocol.\n”);
    printf (”Copyright Allen–Bradley  09–10–92 jrm, aes, dms.\n”);
    printf (”This program was used and tested on an Allen–Bradley ”);
    printf (”T47 computer.\n”);
    printf (”The Color CVIM communication Parameters for this program
are:\n”);
    printf (”    CFG Host: RS–232 A\n”);
    printf (”    SYS Host: RS232 A\n”);
    printf (”    Trigger Source = (Hosted)\n”);
    printf (”    RS232 A Protocol = DF1\n”);
    printf (”    RS232 A Baud Rate = 9600\n”);
    printf (”    Color CVIM must be in RUNMODE\n”);

    /* Get Serial port number from the user */
    printf (”\nEnter port number (1 for COM1 or 2 for COM2): ”);
    scanf (”%d”, &portnum);

    /* Open comm channel to Color CVIM at 9600 baud */
    _bios_serialcom (_COM_INIT, portnum–1, _COM_CHR8 | _COM_STOP1 |
    _COM_NOPARITY | _COM_9600);
    /* Print options menu on the screen */
    do  {
        printf (”\n\nOperations: \n\n”);
        printf (”0. Echo ’HELLO’\n”);
        printf (”1. Trigger Tool Set\n”);
        printf (”2. Read Discrete Results\n”);
        printf (”3. Read Results Block 1\n”);
        printf (”4. Read configuration\n”);
        printf (”5. Write configuration\n”);
        printf (”\nEnter operation number (0–5) or –1 to quit: ”);

        scanf(”%d”, &op_num );   /* Convert user string input to a number
*/
        replen = err = 0;    /* Initialize control variables */
        switch (op_num)      /* Determine what user selected */
            {

            case 0:      /* echo hello */
                {
                err = send_message (portnum,”\001\000\001HELLO”,8);
                if (!err)
                    err = get_message(portnum,reply,&replen);
                if (replen && !err)
                    {
                    printf (”Response: ”);
                    for (x=0; x<replen; x++)
                        printf (”%c ”,reply[x]);
                    printf (”\n”);
                    }
                }  break;

            case 1:      /* trigger tool set */
                {
                err = send_message (portnum,”\011”,1); /* no reply */
                }  break;

            case 2:      /* read discrete results */
                {
                err = send_message (portnum,”\007\000\001\000”,4);
                if (!err)
                    err = get_message(portnum,reply,&replen);
                if (replen && !err)
                    {
                    printf (”Response: ”);
                    for (x=0; x<replen; x++)
                        printf (”%02X ”,reply[x]);
                    printf (”\n”);
                    }
                }  break;
            case 3:      /* read results block 1 */
                {
                err = send_message (portnum, ”\007\000\001\020\001”,5);
                if (!err)
                    err = get_message(portnum, results, &reslen);
                /* Display the results block */
                if (reslen && !err)
                    {
                    printf (”Results block #1:\n”);
                    for (x=0; x<reslen; x+=2)
                        printf (”%04X ”,results[x]*256 + results[x+1]);
                    printf (”\n”);
                    }
                }  break;

            case 4:      /* read configuration */
                {
                /* send the read all config. blocks command*/
                err = send_message (portnum,”\6\7\377\377\377\377\377”
                          ”\377\377\377\377\377\377\377\377\377\377\377”

”\377\377\377\377\377\377\377\377\377\377\037”,29);
                /* read config blocks 1–213 */
                printf (”Reading Configuration Blocks 1–213. Please
Wait.\n”);
                for (x=0; (x<213) && !err; x++)
                    err = get_message(portnum,config[x],&configlen[x]);
                } break;

            case 5:      /* write configuration */
                {
                /* write entire config. */
                err = send_message (portnum,”\13\7\377\377\377\377\377”
                          ”\377\377\377\377\377\377\377\377\377\377\377”

”\377\377\377\377\377\377\377\377\377\377\037”,29);
                /* give Color CVIM time to prepare */
                for (x=0; x<25000; x++);
                if (!err)
                /* write config blocks 1–213 */
                printf (”Writing Configuration Blocks 1–213. Please
Wait.\n”);
                for (x=0; (x<213) && !err; x++)
                    err = send_message(portnum,config[x],configlen[x]);
                }  break;
            }               /* End switch (op_num) statement */

        if (err)
            printf (”Error code: %4x\n”,err);
        } while (op_num >= 0);      /* End do loop */
    }

int send_message (portnum, msg, len)
int portnum;          /* Communications port # */
unsigned char *msg;   /* Pointer to message data */
int len;              /* Length of message */

/*  Transmits the message pointed to by msg, consisting of len characters.
    If the message is not acknowledged, re–transmits it up to 2 more
times.
    Returns zero if successful or nonzero if an error occurs. */
    {

    int x=0, ch=0, chksum=0, err=0, retry=3;

    do  {

        /* if an incoming char is waiting, clear it out */
        while (_bios_serialcom(_COM_STATUS, portnum–1, 0) & 256)
            _bios_serialcom(_COM_RECEIVE, portnum–1, 0);

       /* send DLE STX to initiate message transfer */
       _bios_serialcom(_COM_SEND,portnum–1,DLE);
       _bios_serialcom(_COM_SEND,portnum–1,STX);

       /* Send all bytes of the selected command & compute checksum */
       for (x=chksum=0; x<len; x++)
           {
           _bios_serialcom(_COM_SEND,portnum–1,msg[x]);
           chksum += msg[x];

           /* If data within the message was 0x10 (a DLE), send another
              DLE so the receiver knows it is data, not a control
character.
              (But don't include the second DLE in the checksum) */

           if (msg[x] == DLE)
               _bios_serialcom(_COM_SEND,portnum–1,msg[x]);
           }

       /* send DLE ETX and CHKSUM */
       _bios_serialcom(_COM_SEND,portnum–1,DLE);
       _bios_serialcom(_COM_SEND,portnum–1,ETX);
       _bios_serialcom(_COM_SEND,portnum–1,(–chksum)&0xFF);

       /* check for DLE ACK */
       ch = _bios_serialcom(_COM_RECEIVE,portnum–1,0);
       err = ch & COMMFLAGS;
       if (err || ((ch & 0xFF) != DLE))   /* Mask character to 8 bits */
```

```
           err |= 0x01;            /* no DLE on send */
       if (!err)
           {
           ch = _bios_serialcom(_COM_RECEIVE,portnum-1,0);
           err = ch & COMMFLAGS;
           if (err || ((ch & 0xFF) != ACK))  /* Mask character to 8 bits
*/
                err |= 0x02;             /* no ACK on send */
           }
       } while (err && (--retry > 0)); /* if any error, retry */
    return (err);
    }

/*  Recieves a message into buffer pointed to by msg & places
    length in *len.  If the checksum is invalid or a timeout occurs,
    sends NAK and attempts to re-receive up to 2 more times.
    Returns zero on success or nonzero if an error occurs. */

int get_message (portnum, msg, len)
int portnum;         /* Serial port number */
unsigned char *msg;    /* Pointer to message buffer */
int     *len;         /* Pointer to return length of message */

    {
    int        good_string = 0;
    int        message_started = 0;
    int        ch=0, err=0, retry=4;
    int        length = 0;
    unsigned char    *msg_start_ptr, df1_bcc=0;

    msg_start_ptr = msg;

    while ( !good_string && retry )
        {
        ch = _bios_serialcom(_COM_RECEIVE,portnum-1,0);
        err |= ch & COMMFLAGS;
        if (( ch & 0xFF ) == DLE )
            {
            ch = _bios_serialcom(_COM_RECEIVE,portnum-1,0);
            err |= ch & COMMFLAGS;
            switch( ch & 0xFF )
                {
                case STX:
                    message_started = 1;
                    break;
                case ETX:
                    message_started = 0;
                    ch = _bios_serialcom(_COM_RECEIVE,portnum-1,0);
                    err |= ch & COMMFLAGS;
            df1_bcc = –df1_bcc;
            if (( (ch & 0xFF) == df1_bcc ) && !err )
                {
                _bios_serialcom(_COM_SEND,portnum–1,DLE);
                _bios_serialcom(_COM_SEND,portnum–1,ACK);
                good_string = 1;
                }
            else
                {
                _bios_serialcom(_COM_SEND,portnum–1,DLE);
                _bios_serialcom(_COM_SEND,portnum–1,NAK);
                if ( ––retry )
                    {
                    err = 0;
                    length = 0;
                    df1_bcc = 0;
                    msg = msg_start_ptr;
                    }
                }
            break;
        case DLE:
            if ( message_started )
                {
                if ( ++length > 128 )
                    {
                    _bios_serialcom(_COM_SEND,portnum–1,DLE);
                    _bios_serialcom(_COM_SEND,portnum–1,NAK);
                    message_started = 0;
                    err = 0;
                    length = 0;
                    df1_bcc = 0;
                    msg = msg_start_ptr;
                    }
                else
                    {
                    *msg++ = ( ch & 0xFF );
                    df1_bcc += ( ch & 0xFF );
                    }
                }
            break;
        default:
            break;
        }
    }
else if ( message_started )
    {
    if ( ++length > 128 )
```

```
            {
            _bios_serialcom(_COM_SEND,portnum–1,DLE);
            _bios_serialcom(_COM_SEND,portnum–1,NAK);
            message_started = 0;
            err = 0;
            length = 0;
            df1_bcc = 0;
            msg = msg_start_ptr;
            }
        else
            {
            *msg++ = ( ch & 0xFF );
            df1_bcc += ( ch & 0xFF );
            }
        }
    }
*len = length;
return (err);
}
```

Chapter 6

Using the Pyramid Integrator Backplane

Chapter Objectives

This chapter:

- Describes the Pyramid Integrator Backplane.
- Describes backplane communication techniques.
- Describes Color CVIM module setup requirements
- Contains a sample PLC–5/250 program.

Note: Refer to Publication No. 5000–2.3 (Allen–Bradley Pyramid Integrator Technical Overview) for a description of the basic hardware components and valid configurations. Use the related publications chart in Chapter 1 to reference other Pyramid Integrator manuals as required.

What Information can be Accessed?

Through the backplane, you can access an area of memory called Shared Memory. Shared memory consists of 1024 words (approximately half of which are presently used). Shared memory contains:

- Color CVIM module Discrete Bit Information (refer to Appendix B). These bits include pass/fail/warning data for inspections and command bits for Color CVIM module operation modes.

- Results Data (refer to Appendix C).

- In addition, you can access the Color CVIM module setup and configuration data through the backplane. Refer to Appendix D.

Host Designation

There are four communications ports which you can simultaneously use to access Color CVIM module data (Remote I/O, RS–232 (A&B), and Backplane). Only one of the communications ports can be designated as the *host* at any given time. Only the *host* can issue commands to control the operation of the Color CVIM module, trigger inspections, upload/download configurations, and change displays. You can read discrete bits and numerical results information through any of the three communications ports, even through non–host devices.

Note: See Chapter 2 for a description of multiple hosts.

What Functions can be Performed Over the Backplane?

A MicroVAX information processor, PLC–5/250, or other device in the Pyramid Integrator rack can request or manipulate the following data through the backplane.

- Obtain Color CVIM module inspection result information. Refer to Appendix B & C.
- Upload or download Color CVIM module configurations for inspections. Refer to Appendix D (CFG Host).
- Issue Read/Write commands between the following Color CVIM module memory locations (CFG Host):

  Color CVIM module Random Access Memory (RAM) and Electrically Erasable Programmable Read Only Memory (EEPROM).

  Color CVIM module RAM and RAM card. The RAM card slides into a slot on front of the Color CVIM module.

  Color CVIM module RAM and host memory.
- Change run–time display menus (SYS Host).
- Enable/Disable local I/O board (SYS Host).
- Force local I/O On or Off (SYS Host).

Note: When communicating with a device through the Pyramid Integrator backplane, Color CVIM module results are posted in shared memory immediately after processing. When communicating with a device through the other ports, results are only available at the end of the inspection cycle.

Color CVIM Module Configuration Instructions

If you are using the Pyramid Integrator backplane for communications, you must configure the Color CVIM module as follows:

Select the backplane for communications:

Note: This step is not required if you are only reading results.

1. Select the setup menu <Setup>.
2. Select the environment menu <Envir/Cam>.
3. Select the system menu <System>.
4. Select a Host menu <CFG Host:> or <SYS Host:>.
5. Choose Pyramid Integrator backplane option (twice) <Pyramid>.

Color CVIM Module Configuration Instructions (cont'd)

Select the Color CVIM module trigger source:

6. Select I/O menu <I/O>.
7. Select the trigger source menu <Trigger Source>.
8. Select either <I/O>, <Hosted> trigger sources or <Autotrigger>.

Note: When changing the host to/from the Pyramid Integrator, you must "pick" the selection twice and then reboot the Color CVIM module. **Make sure that you save your configuration first!**

Obtaining Inspection Result Information Using a PLC–5/250

If you are accessing results through a PLC–5/250, the 1024 words of shared memory are numbered 0 through 1023. The PLC–5/250 treats the shared memory area like an integer file. Table 6.A provides a summary of the data. Refer to Appendix A, B, or C for more detailed information.

Table 6.A
Shared Memory Overview

Block Name	Backplane Word #
Handshake	0
Reserved	1
Discrete I/O Outputs (Host Command Block)	2 to 9
Reserved	10 to 15
First Discrete I/O Inputs (Tool Inspection Results)	16 to 23
Second Discrete I/O Inputs (Tool Inspection Results)	24 to 31
Results Block 1	32 to 95
Results Block 2	96 to 159
Results Block 3	160 to 223
Results Block 4	224 to 287
Results Block 5	288 to 351
Results Block 6	352 to 415
Reserved	416 to 1022
Interrupt Control Word	1023
Configuration	Use Message Mail

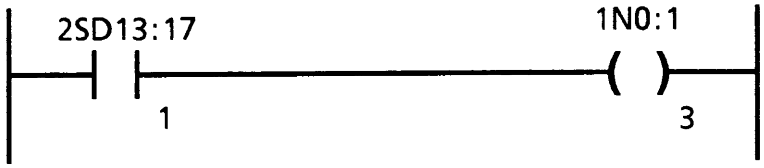
Obtaining Inspection Result Information Using a PLC–5/250 (cont'd)

The address of Color CVIM module shared memory is always SD13, with the addressing as follows:

Color CVIM Thumbwheel No.	SD13	Word/Bit No.

For example: Assume that the Color CVIM module has a thumbwheel setting of #2. The data that you want to read is window 1 fault flag. Refer to Appendix B, the bit you want to read is bit 01 of word 17. The PLC bit address would then be:

2SD13:17/01



To access numerical values, you can use any PLC instruction which manipulates file values. Values can be read individually or in groups. For example, you can use the compute (CPT) function to move a single word from the shared data (SD) file (resides in resource manager) to an integer file on the local logic processor. Other PLC file instructions such as copy (FAL) or compare (FSC) can also be used to manipulate shared data results.

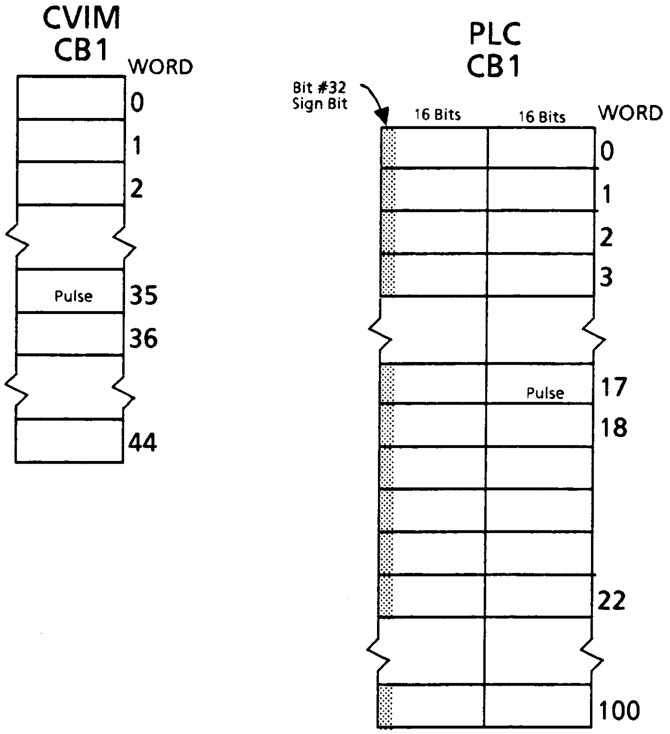
Note: ACVIM family processor in the same rack *must* have different thumbwheel settings.

Manipulating Configuration Data Using a PLC–5/250

When you transfer configuration data from the Color CVIM to a PLC–5/250 you use a message command and designate a long integer file as the destination (internal table address). Long integer files contain 32 bit elements. With this format, configuration block 1 would be arranged as shown in Figure 6.1.

Manipulating Configuration Data Using a PLC–5/250 (cont'd)

Figure 6.1
32 Bit Long Integer Files



Use the message instruction (MSG) to transfer configuration data between the PLC and the Color CVIM. The message instruction can transfer up to 10,000 elements of data/commands. When used for Pyramid Integrator backplane instructions, the message instruction commands the resource manager module to transfer data between two module addresses. Use the RS–232 ASCII command set in the external data table address field to perform the desired function. To read a configuration of a Color CVIM with a thumbwheel setting of #2 into the Resource Manager long integer file #9 starting at element 0, the message instruction parameters would be entered as follows:

F1	Requested size:	0
F2	Priority:	NORMAL
F3	Local/Remote:	LOCAL
F4	Local Link Type:	DH +
F5	REM Link Type	N/A
F6	Station ID:	Node# = 0
F7	Module ID:	Class CVIM TW# 1 Port # = 1
F8	COMM CMND	TYPED READ
F9	Internal Data Table Address	0L9:0
F10	External Data Table Address	> RC,CB 1–213"

Sample PLC–5/250 Program

The following program shows how to trigger an inspection,, upload an entire Color CVIM configuration for archiving and later downloading, read Gage 1, and control the display. All requests are done through the Request word 1N0:0.

Rung #0 Activates the 100 millisecond command timer and the command one shot.

Rung #1 Clears the request word if the timer is done timing the 100 milliseconds.

Rung #2 Triggers the Color CVIM to perform an inspection upon a false to true transition of 1N0:0/0.

Rung #3 Reads the integer value of gage 1 and places it in 1N0:2 upon a false to true transition of 1N0:0/1.

Rung #4 Sends one message (0MSG0:0) to read the configuration as described on page 6–13 upon a false to true transition of 1N0:0/2.

Rung #5 Sends one message (0MSG0:1) to write the configuration as described on page 6–13 upon a false to true transition of 1N0:0/3.

Rung #6 Locks Lightpen Requests upon a false to true transition of 1N0:0/4.

Rung #7 Unlocks Lightpen Request upon a false to true transition of 1N0:0/11.

Rung #8 Processes Lightpen Runtime Control Request upon a false to true transition of 1N0:0/5 with the 0N0:3/0–7 containing the runtime display control, 0N0:3/8–15 containing the freeze control, 0N0:4/3 the resume control, 0N0:4/4&5 the page control, 0N0:4/6 the reset statistics, and 0N0:4/7 the reset counters. See Appendix B for a description of the control bits.

Rung #9 Process force commands upon a false to true transition of 1N0:0/6, with 1N0:5 containing the ON mask and 1N0:6 containing the OFF mask.

Rung #10 Move configuration from EEPROM to RAM upon a false to true transition of 1N0:0/7.

Rung #11 Move configuration from RAM to EEPROM upon a false to true transition of 1N0:0/8.

Rung #12 Move configuration from Card Location 1 to RAM upon a false to true transition of 1N0:0/9.

Rung #13 Move configuration from RAM to card location 1 upon a false to true transition of 1N0:0/10.

Note: See Remote I/O (Chapter 3) for converting 16.16 data to a PLC floating point value. Also see Chapter 3 for additional sample programs, and Chapter 5 for RS–232 commands.

The program begins on the next page.

Sample PLC–5/250 Program (cont’d)

```
5370-CVIMC EXAMPLE PROGRAM
Program Listing Report                  PLC-5/250    File CVIMC
Rung 1TRN0:0
|                                                                              |
+-[AFI]------------------------------------------------------------------[EOT]-+
|                                                                              |

Rung 1TRN0:1
|                                                                              |
+--------------------------------[END OF FILE]---------------------------------+
|                                                                              |

Rung 1STEP0:0
|                                                          5370-CVIMC          |
|                                                          REQUEST             |
|      5370-CVIMC                                          100 MILLI-          |
|      REQUEST                                             SECOND              |
|      WORD                                                TIMER               |
| +CMP---------------+                              +TON---------------+       |
+-+COMPARE           +------------------------+-----+TIMER ON DELAY    +-(EN)+-+
| |Expression        |                        |     |Timer        1T0:0|     | |
| |1N0:0<>0          |                        |     |Time base     0.01+-(DN)| |
| +------------------+                        |     |Preset          10|     | |
|                                             |     |Accum            0|     | |
|                                             |     +------------------+     | |
|                                             |          5370-CIMVC          | |
|                                             |          ONE SHOT            | |
|                                             |          MESSGAGE            | |
|                                             |          ACTIVATION          | |
|                                             |          BIT                 | |
|                                             |+OSR--------------------+     | |
|                                             ++ONE SHOT RISING        +-(OB)+ |
|                                              |Output Bit      1N0:1/0|       |
|                                              |Storage Bit     1N0:1/1+-(SB)  |
|                                              +-----------------------+       |

Rung 1STEP0:1
| 5370-CVIMC|5370-CVIMC|5370-CVIMC                                             |
| 100 MILLI-|100 MILLI-|100 MILLI-                                             |
| SECOND    |SECOND    |SECOND                                 5370-CVIMC      |
| TIMER     |TIMER     |TIMER                                  REQUEST         |
| ENABLED   |TIMING    |DISABLED                               WORD            |
|   1T0:0      1T0:0      1T0:0                           +CLR---------------+ |
+----] [--------]/[--------] [----------------------------+CLEAR             +-+
|      EN         TT         DN                           |Dest         1N0:0| |
|                                                         |                 0| |
|                                                         +------------------+ |

Rung 1STEP0:2
| 5370-CVIMC                                                        5370-CVIMC |
| TRIGGER                                                           HOST       |
| INSPECTION                                                        TRIGGER    |
| REQUEST                                                           TOOLSET    |
| BIT                                                               BIT        |
|   1N0:0                                                           1SD13:1023 |
+----] [---------------------------------------------------------------( )-----+
|       0                                                                10    |
Rung 1STEP0:3
| GAGE 1                                                       SAVE THE        |
| RESULT                                                       5370-CVIMC      |
| DATA                                                         GAGE 1          |
| REQUEST                                                      RESULT          |
| BIT                                                          DATA            |
|   1N0:0                                                 +CPT---------------+ |
+----] [--------------------------------------------------+COMPUTE           +-+
|       1                                                 |Dest         1N0:2| |
|                                                         |                 0| |
|                                                         |Expression        | |
|                                                         |1SD13:73          | |
|                                                         +------------------+ |

Rung 1STEP0:4
| 5370-CVIMC|5370-CIMVC                                   5370-CVIMC           |
| READ ALL  |ONE SHOT                                     READ ALL             |
| CONFIG.   |MESSGAGE                                     CONFIG.              |
| REQUEST   |ACTIVATION                                   MESSAGE              |
| BIT       |BIT                                          COMMAND              |
|   1N0:0      1N0:1                            +MSG--------------------+      |
+----] [--------] [-----------------------------+SEND/RECEIVE MESSAGE   +-(EN)-+
|       2          0                            |Control Block   0MSG0:0+-(DN) |
|                                               |                       +-(ER) |
|                                               +-----------------------+      |

Rung 1STEP0:5
| 5370-CVIMC|5370-CIMVC                                   5370-CVIMC           |
| WRITE ALL |ONE SHOT                                     WRITE ALL            |
| CONFIG.   |MESSGAGE                                     CONFIG.              |
| REQUEST   |ACTIVATION                                   MESSAGE              |
| BIT       |BIT                                          COMMAND              |
|   1N0:0      1N0:1                            +MSG--------------------+      |
+----] [--------] [-----------------------------+SEND/RECEIVE MESSAGE   +-(EN)-+
|       3          0                            |Control Block   0MSG0:1+-(DN) |
|                                               |                       +-(ER) |
|                                               +-----------------------+      |

Rung 1STEP0:6
| REQUEST                                                          5370-CVIMC  |
| LOCK                                                             LOCK        |
| LIGHT PEN                                                        LIGHT PEN   |
| CONTROL                                                          REQUEST     |
| BIT                                                              BIT         |
|   1N0:0                                                          1SD13:1023  |
+----] [----------------------------------------------------------+---(L)----+-+
|       4                                                         |      8   | |
|                                                                 |5370-CVIMC| |
|                                                                 |UNLOCK    | |
|                                                                 |LIGHT PEN | |
|                                                                 |REQUEST   | |
|                                                                 |BIT       | |
|                                                                 |1SD13:1023| |
|                                                                 +---(U)----+ |
|                                                                        9     |
Rung 1STEP0:7
| REQUEST                                                          5370-CVIMC  |
| UNLOCK                                                           UNLOCK      |
| LIGHT PEN                                                        LIGHT PEN   |
| CONTROL                                                          REQUEST     |
| BIT                                                              BIT         |
|   1N0:0                                                          1SD13:1023  |
+----] [----------------------------------------------------------+---(L)----+-+
|      11                                                         |      9   | |
|                                                                 |5370-CVIMC| |
|                                                                 |LOCK      | |
|                                                                 |LIGHT PEN | |
|                                                                 |REQUEST   | |
|                                                                 |BIT       | |
|                                                                 |1SD13:1023| |
|                                                                 +---(U)----+ |
|                                                                        8     |

Rung 1STEP0:8
| REQUEST                                                     5370-CVIMC       |
| LIGHT PEN                                                   RUNTIME          |
| RUNTIME                                                     DISPLAY &        |
| CONTROL                                                     FREEZE           |
| BIT                                                         CONTROL          |
|   1N0:0                                               +CPT---------------+  |
+----] [------------------------------------------------++COMPUTE           ++-+
|       5                                               ||Dest       1SD13:3|| |
|                                                       ||                 0|| |
|                                                       ||Expression        || |
|                                                       ||1N0:3             || |
|                                                       |+------------------+| |
|                                                       |     5370-CVIMC     | |
|                                                       |     RESUME,        | |
|                                                       |     PAGE, &        | |
|                                                       |     RESET          | |
|                                                       |     CONTROL        | |
|                                                       |+CPT---------------+| |
|                                                       ++COMPUTE           ++ |
|                                                       ||Dest       1SD13:8|| |
|                                                       ||                16|| |
|                                                       ||Expression        || |
|                                                       ||1N0:4             || |
|                                                       |+------------------+| |
|                                                       |          5370-CVIMC| |
|                                                       |          LIGHT PEN | |
|                                                       |          REQUEST   | |
|                                                       |          BIT       | |
|                                                       |          1SD13:1023| |
|                                                       +-------------( )----+ |
|                                                                       12     |
Rung 1STEP0:9
| REQUEST                                                     5370-CVIMC       |
| LOCAL I/O                                                   LOCAL            |
| FORCE                                                       OUTPUTS          |
| ON OR OFF                                                   FORCE ON         |
| BIT                                                         MASK             |
|   1N0:0                                                +CPT---------------+  |
+----] [------------------------------------------------++COMPUTE           ++-+
|       6                                               ||Dest       1SD13:6|| |
|                                                       ||                 0|| |
|                                                       ||Expression        || |
|                                                       ||1N0:5             || |
|                                                       |+------------------+| |
|                                                       |     5370-CVIMC     | |
|                                                       |     LOCAL          | |
|                                                       |     OUTPUTS        | |
|                                                       |     FORCE OFF      | |
|                                                       |     MASK           | |
|                                                       |+CPT---------------+| |
|                                                       ++COMPUTE           ++ |
|                                                       ||Dest       1SD13:7|| |
|                                                       ||                 0|| |
|                                                       ||Expression        || |
|                                                       ||1N0:6             || |
|                                                       |+------------------+| |
|                                                       |          5370-CVIMC| |
|                                                       |          ENABLE    | |
|                                                       |          OUTPUTS   | |
|                                                       |          BIT       | |
|                                                       |           1SD13:4  | |
|                                                       +-------------( )----+ |
|                                                       |                5   | |
|                                                       |          5370-CVIMC| |
|                                                       |          ENABLE    | |
|                                                       |          FORCES    | |
|                                                       |          BIT       | |
|                                                       |           1SD13:4  | |
|                                                       +-------------( )----+ |
|                                                       |                7   | |
|                                                       |          5370-CVIMC| |
|                                                       |          I/O       | |
|                                                       |          REQUEST   | |
|                                                       |          BIT       | |
|                                                       |          1SD13:1023| |
|                                                       +-------------( )----+ |
|                                                                       13     |

Rung 1STEP0:10
| REQUEST                                                          5370-CVIMC  |
| MOVE                                                             MOVE        |
| CONFIG.                                                          CONFIG.     |
| EEPROM TO                                                        EEPROM TO   |
| RAM BIT                                                          RAM BIT     |
|   1N0:0                                                           1SD13:4    |
+----] [----------------------------------------------------------+---( )----+-+
|       7                                                         |      8   | |
|                                                                 |5370-CVIMC| |
|                                                                 |CONFIG.   | |
|                                                                 |MOVE      | |
|                                                                 |REQUEST   | |
|                                                                 |BIT       | |
|                                                                 |1SD13:1023| |
|                                                                 +---( )----+ |
|                                                                       14     |
Rung 1STEP0:11
| REQUEST                                                          5370-CVIMC  |
| MOVE                                                             MOVE        |
| CONFIG.                                                          CONFIG.     |
| RAM TO                                                           RAM TO      |
| EEPROM BIT                                                       EEPROM BIT  |
|   1N0:0                                                           1SD13:4    |
+----] [----------------------------------------------------------+---( )----+-+
|       8                                                         |      9   | |
|                                                                 |5370-CVIMC| |
|                                                                 |CONFIG.   | |
|                                                                 |MOVE      | |
|                                                                 |REQUEST   | |
|                                                                 |BIT       | |
|                                                                 |1SD13:1023| |
|                                                                 +---( )----+ |
|                                                                       14      |

Rung 1STEP0:12
| REQUEST                                                          5370-CVIMC  |
| MOVE                                                             MOVE        |
| CONFIG.                                                          CONFIG.     |
| CARD 1 TO                                                        CARD 1 TO   |
| RAM BIT                                                          RAM BIT     |
|   1N0:0                                                           1SD13:4    |
+----] [----------------------------------------------------------+---( )----+-+
|       9                                                         |     10   | |
|                                                                 |5370-CVIMC| |
|                                                                 |CONFIG.   | |
|                                                                 |MOVE      | |
|                                                                 |REQUEST   | |
|                                                                 |BIT       | |
|                                                                 |1SD13:1023| |
|                                                                 +---( )----+ |
|                                                                       14      |

Rung 1STEP0:13
| REQUEST                                                          5370-CVIMC  |
| MOVE                                                             MOVE        |
| CONFIG.                                                          CONFIG.     |
| RAM TO                                                           RAM TO      |
| CARD 1 BIT                                                       CARD 1 BIT  |
|   1N0:0                                                           1SD13:4    |
+----] [----------------------------------------------------------+---( )----+-+
|      10                                                         |     11   | |
|                                                                 |5370-CVIMC| |
|                                                                 |CONFIG.   | |
|                                                                 |MOVE      | |
|                                                                 |REQUEST   | |
|                                                                 |BIT       | |
|                                                                 |1SD13:1023| |
|                                                                 +---( )----+ |
|                                                                       14      |

Rung 1STEP0:14
|                                                                              |
+--------------------------------[END OF FILE]---------------------------------+
|                                                                              |
```

Sample PLC–5/250 Program (cont’d)

CONTROL BLOCK OMSG0:0

F1	Module ID:	Class = CVIM Pw# = 1 Port# = 1
F2	Priority:	HIGH
F3	Local/Remote:	LOCAL
F4	Local Link Type:	DH+
F5	Remote Link Type:	N/A
F6	Station ID:	Node# = 0
F7	Communication Command:	TYPED READ
F8	Internal Data Table Addr:	0L9:0
F9	External Data Table Addr:	”>rc,cb 1–213”
F10	Requested Size:	0
	Parameters:	N/A

CONTROL BLOCK OMSG0:1

F1	Module ID:	Class = CVIM Pw# = 1 Port# = 1
F2	Priority:	HIGH
F3	Local/Remote:	LOCAL
F4	Local Link Type:	DH+
F5	Remote Link Type:	N/A
F6	Station ID:	Node# = 0
F7	Communication Command:	TYPED WRITE
F8	Internal Data Table Addr:	0L9:0
F9	External Data Table Addr:	”>wc,cb1–213”
F10	Requested Size(element):	0
	Parameters:	N/A

Note: You must create a file using Memory Map to create the file 0L9 and enable words 0 through 9999 in that file.

Obtaining Inspection Result Information Using a MicroVAX Information Processor

If you are accessing results through a MicroVAX information processor, you should use the standard library functions to access the data. The MicroVAX library instructions are called up using a simplified “C” type language. For example:

DTL__READ__W(Address, &Result, &Error);

Refer to the MicroVAX manuals for the available library routines.

Appendix **A**

Results/Configuration Data Overview

Introduction

This appendix provides an overview of the word and bit addresses of data stored in memory if you are communicating with the Color CVIM through the REMOTE I/O, Backplane, or RS–232 port.

Also provided is an explanation of how the Color CVIM stores fractional data.

Overview

Table A.1 provides an overview of shared memory. Detailed descriptions of each word and bit functions can be found in the other appendices:

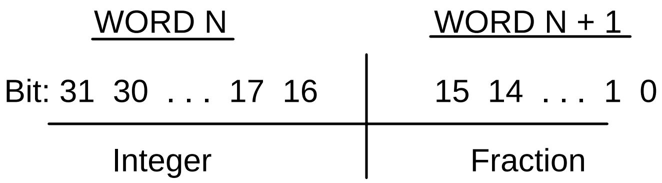
- Appendix B — Discrete Results Bits
- Appendix C — Numerical Results Data
- Appendix D — Configuration Data

Block Name	Backplane Word # XSD13i*	Backplane Word #
Handshake	0	N/A
Reserved	1	N/A
Discrete I/O Outputs (Host Command Block)	2 to 9	O:20 to O:27
Reserved	10 to 15	N/A
First Discrete I/O Inputs (Tool Inspection Results)	16 to 23	I:20 to I:27**
Second Discrete I/O Inputs (Tool Inspection Results)	24 to 31	I:20 to I:27**
Results Block 1	32 to 95	0 to 63***
Results Block 2	96 to 159	0 to 63***
Results Block 3	160 to 223	0 to 63***
Results Block 4	224 to 287	0 to 63***
Results Block 5	288 to 351	0 to 63***
Results Block 6	352 to 415	0 to 63***
Reserved	416 to 1022	N/A
Interrupt Control Word	1023	N/A
Configuration	Use Message Mail	0 to 63***

* X = Color CVIM thumbwheel number.
** The two parts of the toolset use the same addresses. The PLC must tell the Color CVIM which part to use by setting the appropriate output bit.
*** This is done by doing a Block Transfer Read of the Color CVIM. The PLC must set the appropriate output bits to get the correct block.

Fractional Notation

Inspections which produce results that are fractional are represented using two words (32 bits). The first 16 bits are the integer portion and the second 16 bits are the fractional portion:



The integer portion of the value is interpreted as a standard 16 bit signed integer where each bit is equal to:

Integer Bit Value $2^{(n)}$

Where n is the bit number.

Bits in the fractional portion of the value are interpreted as:

Fractional Bit Value $2^{(n-16)}$

Where n is the bit number.

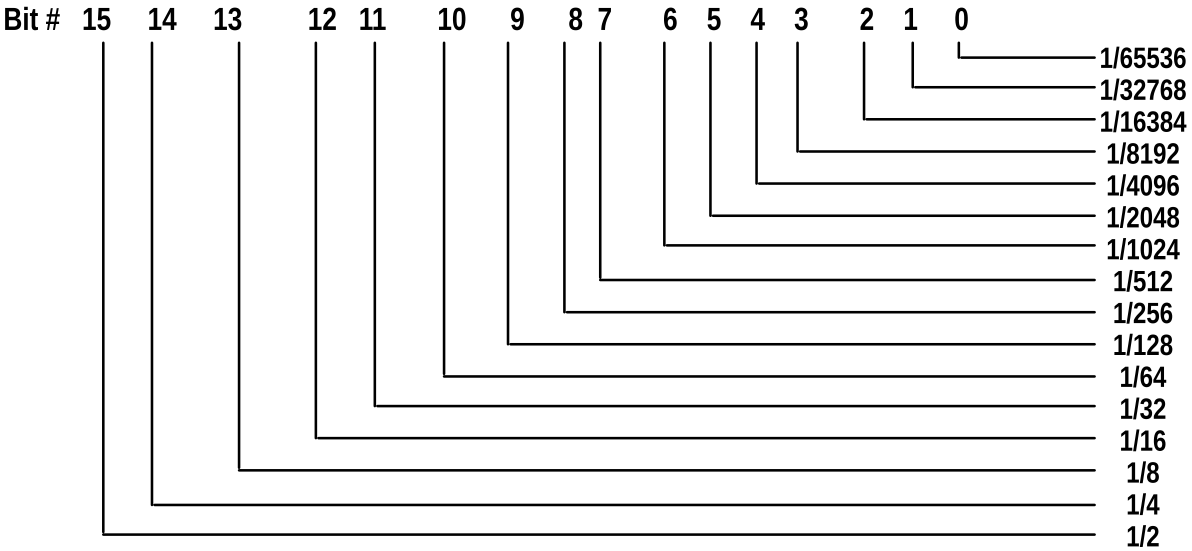
For example: bit 15 in the fractional portion of the value is equal to 1/2:

$$2^{(15-16)} = 2^{(-1)} = 1/2$$

Fractional Notation (cont'd)

We have provided the following chart to assist you:

Figure A.1
Fractional Notation

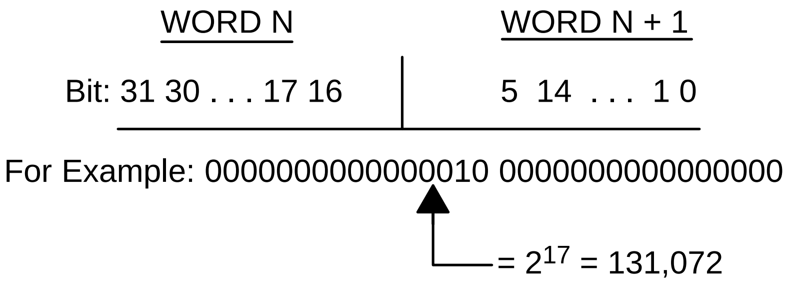


For example: 0000000000000011.0100000000000000 = 3.25

Note: For a sample PLC program which converts 16 point 16 values to floating point values, refer to Chapter 4.

32 Bit Integer Format

The Color CVIM stores some of the data as a 32 bit integer. The first byte contains bits 16 through 31 of the integer and the next byte contains bits 0 through 15 of the integer.



Appendix **B**

Discrete Bit Results Bits (Host Input & Output Bits)

Introduction

This appendix lists the function of both the discrete bit inputs and outputs. These bits can be accessed through the Remote I/O port and Pyramid Integrator backplane. You cannot manipulate these bits through the RS–232 ports (A or B) but you can perform many of their functions.

Discrete Bit Inputs

With each inspection that the Color CVIM module performs, individual bits are set. There are 256 bits that can be set as inputs to a host device. These bits (part of the inspection results) indicate:

- Configuration faults.
- Module Busy flag.
- Missed Trigger flags.
- Results Valid flags.
- Inspection Tool Pass/Fail/Warning flags

Note: If you are using the Remote I/O link, the bits for inspection tool pass/warning/fail apply to either t the first part or the second part. You select which part by setting bits 4 and 5 in word 3 of the discrete output bits.

Note: For your convenience, we have provided bit numbers for both a PLC (octal) and Pyramid Integrator (decimal).

Note to PLC–2 Users: When you use any PLC–2 family processor with the Color CVIM node adapter, you should understand the operation of the PLC Block Transfer Done bits for Read and Write instructions. PLC–2 family processors use the input image table for these bits all other PLCs can specify integer files for this function. This means that a PLC–2 user must use proper programming techniques to avoid confusion between the following bits:

- Color CVIM discrete I/O input word 0, bit 6 (data valid first part) and bit 7 (data valid second part).
- PLC–2 family input image table word 0, bit 6 (BTW done bit) and bit 7 (BTR done bit).

Table B.1 lists the discrete bit inputs.

Table B.1
Discrete Bit Inputs

Note: The Remote I/O Bit Number column (marked with an asterisk in the table below) are expressed in Octal. All other values are expressed in decimal.

Note: The following chart is only valid if you read the discrete bits for both parts of the toolset with the RS–232 command “>RRn”. The first two numbers are the number of triggers and number of failed inspections (in that order). The DF1 command “0700n00” uses hexidecimal long words. Word 1, bytes 0 through 3, are for the number of triggers. Word 2, bytes 4 through 7 is the number of failed inspections.

PI Backplane		Remote I/O		RS–232 ASCII		RS–232 DF1		Definition	Usage	Notes
Word #	Bit #	Word #	Bit #*	Hex #	Bit #	Byte #	Bit #			
16	0	0	0	2	0	9	0	Not Used		
16	1	0	1	2	1	9	1	Configuration Error	0 = No Error 1 = Error	Configuration error bit is set after any invalid configuration block write to the Color CVIM. This flag is also set after validation errors.
16	2	0	2	2	2	9	2	Mastership Flag	0 = Not Master 1 = Master	The device which reads this bit as 1 is the host. Not applicable for RS–232 communications.
16	3	0	3	2	3	9	3	Module Busy	0 = Not Busy 1 = Busy	Module Busy bit is set during the SETUP mode and while sending a configuration download to the Color CVIM.
16	4	0	4	2	4	9	4	Trigger NAK	0 = OK 1 = Trigger Missed	
16	5	0	5	2	5	9	5	Post Results	0 = Part 1 1 = Part 2	
16	6	0	6	2	6	9	6	Toolset Data Valid	0 = Not Valid 1 = Results Valid	Data Valid bit is reset when a user enters the SETUP mode and when pulsed output duration has expired. Refer to Chapter 3.
16	7	0	7	2	7	9	7	Not Used		

Table B.1
Discrete Bit Inputs (Cont'd)

PI Backplane		Remote I/O		RS–232 ASCII		RS–232 DF1		Definition	Usage	Notes
Word #	Bit #	Word #	Bit #*	Hex #	Bit #	Byte #	Bit #			
16	8	0	10	3	0	8	0	Reference Line 1 Flag	0 = Pass 1 = Fail	
16	9	0	11	3	1	8	1	Reference Line 2 Flag	0 = Pass 1 = Fail	
16	10	0	12	3	2	8	2	Reference Line 3 Flag	0 = Pass 1 = Fail	
16	11	0	13	3	3	8	3	Reference Window 1 Flag	0 = Pass 1 = Fail	
16	12	0	14	3	4	8	4	Reference Window 2 Flag	0 = Pass 1 = Fail	
16	13	0	15	3	5	8	5	Reference Window 3 Flag	0 = Pass 1 = Fail	
16	14	0	16	3	6	8	6	Light Probe Flag	0 = No Fault 1 = Fault	
16	15	0	17	3	7	8	7	Master Fault	0 = Pass 1 = Fail	Master Fault bit is set if any tool fails an inspection.
17	0	1	0	4	0	11	0	Window 1 Warning Flag	0 = Pass 1 = Fail	
17	1	1	1	4	1	11	1	Window 1 Fault Flag	0 = Pass 1 = Fail	
17	2	1	2	4	2	11	2	Window 2 Warning Flag	0 = Pass 1 = Fail	
17	3	1	3	4	3	11	3	Window 2 Fault Flag	0 = Pass 1 = Fail	
17	4	1	4	4	4	11	4	Window 3 Warning Flag	0 = Pass 1 = Fail	
17	5	1	5	4	5	11	5	Window 3 Fault Flag	0 = Pass 1 = Fail	
17	6	1	6	4	6	11	6	Window 4 Warning Flag	0 = Pass 1 = Fail	
17	7	1	7	4	7	11	7	Window 4 Fault Flag	0 = Pass 1 = Fail	

Table B.1
Discrete Bit Inputs (Cont'd)

PI Backplane		Remote I/O		RS–232 ASCII		RS–232 DF1		Definition	Usage	Notes
Word #	Bit #	Word #	Bit #*	Hex #	Bit #	Byte #	Bit #			
17	8	1	10	5	0	10	0	Window 5 Warning Flag	0 = Pass 1 = Fail	
17	9	1	11	5	1	10	1	Window 5 Fault Flag	0 = Pass 1 = Fail	
17	10	1	12	5	2	10	2	Window 6 Warning Flag	0 = Pass 1 = Fail	
17	11	1	13	5	3	10	3	Window 6 Fault Flag	0 = Pass 1 = Fail	
17	12	1	14	5	4	10	4	Window 7 Warning Flag	0 = Pass 1 = Fail	
17	13	1	15	5	5	10	5	Window 7 Fault Flag	0 = Pass 1 = Fail	
17	14	1	16	5	6	10	6	Window 8 Warning Flag	0 = Pass 1 = Fail	
17	15	1	17	5	7	10	7	Window 8 Fault Flag	0 = Pass 1 = Fail	
18	0	2	0	6	0	13	0	Window 9 Warning Flag	0 = Pass 1 = Fail	
18	1	2	1	6	1	13	1	Window 9 Fault Flag	0 = Pass 1 = Fail	
18	2	2	2	6	2	13	2	Window 10 Warning Flag	0 = Pass 1 = Fail	
18	3	2	3	6	3	13	3	Window 10 Fault Flag	0 = Pass 1 = Fail	
18	4	2	4	6	4	13	4	Window 11 Warning Flag	0 = Pass 1 = Fail	
18	5	2	5	6	5	13	5	Window 11 Fault Flag	0 = Pass 1 = Fail	
18	6	2	6	6	6	13	6	Window 12 Warning Flag	0 = Pass 1 = Fail	
18	7	2	7	6	7	13	7	Window 12 Fault Flag	0 = Pass 1 = Fail	

Table B.1
Discrete Bit Inputs (Cont'd)

PI Backplane		Remote I/O		RS–232 ASCII		RS–232 DF1		Definition	Usage	Notes
Word #	Bit #	Word #	Bit #*	Hex #	Bit #	Byte #	Bit #			
18	8	2	10	7	0	12	0	Window 13 Warning Flag	0 = Pass 1 = Fail	
18	9	2	11	7	1	12	1	Window 13 Fault Flag	0 = Pass 1 = Fail	
18	10	2	12	7	2	12	2	Window 14 Warning Flag	0 = Pass 1 = Fail	
18	11	2	13	7	3	12	3	Window 14 Fault Flag	0 = Pass 1 = Fail	
18	12	2	14	7	4	12	4	Window 15 Warning Flag	0 = Pass 1 = Fail	
18	13	2	15	7	5	12	5	Window 15 Fault Flag	0 = Pass 1 = Fail	
18	14	2	16	7	6	12	6	Window 16 Warning Flag	0 = Pass 1 = Fail	
18	15	2	17	7	7	12	7	Window 16 Fault Flag	0 = Pass 1 = Fail	
19	0	3	0	8	0	15	0	Window 17 Warning Flag	0 = Pass 1 = Fail	
19	1	3	1	8	1	15	1	Window 17 Fault Flag	0 = Pass 1 = Fail	
19	2	3	2	8	2	15	2	Window 18 Warning Flag	0 = Pass 1 = Fail	
19	3	3	3	8	3	15	3	Window 18 Fault Flag	0 = Pass 1 = Fail	
19	4	3	4	8	4	15	4	Window 19 Warning Flag	0 = Pass 1 = Fail	
19	5	3	5	8	5	15	5	Window 19 Fault Flag	0 = Pass 1 = Fail	
19	6	3	6	8	6	15	6	Window 20 Warning Flag	0 = Pass 1 = Fail	
19	7	3	7	8	7	15	7	Window 20 Fault Flag	0 = Pass 1 = Fail	

Table B.1
Discrete Bit Inputs (Cont'd)

PI Backplane		Remote I/O		RS–232 ASCII		RS–232 DF1		Definition	Usage	Notes
Word #	Bit #	Word #	Bit #*	Hex #	Bit #	Byte #	Bit #			
19	8	3	10	9	0	14	0	Window 21 Warning Flag	0 = Pass 1 = Fail	
19	9	3	11	9	1	14	1	Window 21 Fault Flag	0 = Pass 1 = Fail	
19	10	3	12	9	2	14	2	Window 22 Warning Flag	0 = Pass 1 = Fail	
19	11	3	13	9	3	14	3	Window 22 Fault Flag	0 = Pass 1 = Fail	
19	12	3	14	9	4	14	4	Window 23 Warning Flag	0 = Pass 1 = Fail	
19	13	3	15	9	5	14	5	Window 23 Fault Flag	0 = Pass 1 = Fail	
19	14	3	16	9	6	14	6	Window 24 Warning Flag	0 = Pass 1 = Fail	
19	15	3	17	9	7	14	7	Window 24 Fault Flag	0 = Pass 1 = Fail	
20	0	4	0	10	0	17	0	Gage 1 Warning Flag	0 = Pass 1 = Fail	
20	1	4	1	10	1	17	1	Gage 1 Fault Flag	0 = Pass 1 = Fail	
20	2	4	2	10	2	17	2	Gage 2 Warning Flag	0 = Pass 1 = Fail	
20	3	4	3	10	3	17	3	Gage 2 Fault Flag	0 = Pass 1 = Fail	
20	4	4	4	10	4	17	4	Gage 3 Warning Flag	0 = Pass 1 = Fail	
20	5	4	5	10	5	17	5	Gage 3 Fault Flag	0 = Pass 1 = Fail	
20	6	4	6	10	6	17	6	Gage 4 Warning Flag	0 = Pass 1 = Fail	
20	7	4	7	10	7	17	7	Gage 4 Fault Flag	0 = Pass 1 = Fail	

Table B.1
Discrete Bit Inputs (Cont’d)

PI Backplane		Remote I/O		RS–232 ASCII		RS–232 DF1		Definition	Usage	Notes
Word #	Bit #	Word #	Bit #*	Hex #	Bit #	Byte #	Bit #			
20	8	4	10	11	0	16	0	Gage 5 Warning Flag	0 = Pass 1 = Fail	
20	9	4	11	11	1	16	1	Gage 5 Fault Flag	0 = Pass 1 = Fail	
20	10	4	12	11	2	16	2	Gage 6 Warning Flag	0 = Pass 1 = Fail	
20	11	4	13	11	3	16	3	Gage 6 Fault Flag	0 = Pass 1 = Fail	
20	12	4	14	11	4	16	4	Gage 7 Warning Flag	0 = Pass 1 = Fail	
20	13	4	15	11	5	16	5	Gage 7 Fault Flag	0 = Pass 1 = Fail	
20	14	4	16	11	6	16	6	Gage 8 Warning Flag	0 = Pass 1 = Fail	
20	15	4	17	11	7	16	7	Gage 8 Fault Flag	0 = Pass 1 = Fail	
21	0	5	0	12	0	19	0	Gage 9 Warning Flag	0 = Pass 1 = Fail	
21	1	5	1	12	1	19	1	Gage 9 Fault Flag	0 = Pass 1 = Fail	
21	2	5	2	12	2	19	2	Gage 10 Warning Flag	0 = Pass 1 = Fail	
21	3	5	3	12	3	19	3	Gage 10 Fault Flag	0 = Pass 1 = Fail	
21	4	5	4	12	4	19	4	Gage 11 Warning Flag	0 = Pass 1 = Fail	
21	5	5	5	12	5	19	5	Gage 11 Fault Flag	0 = Pass 1 = Fail	
21	6	5	6	12	6	19	6	Gage 12 Warning Flag	0 = Pass 1 = Fail	
21	7	5	7	12	7	19	7	Gage 12 Fault Flag	0 = Pass 1 = Fail	

Table B.1
Discrete Bit Inputs (Cont'd)

PI Backplane		Remote I/O		RS–232 ASCII		RS–232 DF1		Definition	Usage	Notes
Word #	Bit #	Word #	Bit #*	Hex #	Bit #	Byte #	Bit #			
21	8	5	10	13	0	18	0	Gage 13 Warning Flag	0 = Pass 1 = Fail	
21	9	5	11	13	1	18	1	Gage 13 Fault Flag	0 = Pass 1 = Fail	
21	10	5	12	13	2	18	2	Gage 14 Warning Flag	0 = Pass 1 = Fail	
21	11	5	13	13	3	18	3	Gage 14 Fault Flag	0 = Pass 1 = Fail	
21	12	5	14	13	4	18	4	Gage 15 Warning Flag	0 = Pass 1 = Fail	
21	13	5	15	13	5	18	5	Gage 15 Fault Flag	0 = Pass 1 = Fail	
21	14	5	16	13	6	18	6	Gage 16 Warning Flag	0 = Pass 1 = Fail	
21	15	5	17	13	7	18	7	Gage 16 Fault Flag	0 = Pass 1 = Fail	
22	0	6	0	14	0	21	0	Gage 17 Warning Flag	0 = Pass 1 = Fail	
22	1	6	1	14	1	21	1	Gage 17 Fault Flag	0 = Pass 1 = Fail	
22	2	6	2	14	2	21	2	Gage 18 Warning Flag	0 = Pass 1 = Fail	
22	3	6	3	14	3	21	3	Gage 18 Fault Flag	0 = Pass 1 = Fail	
22	4	6	4	14	4	21	4	Gage 19 Warning Flag	0 = Pass 1 = Fail	
22	5	6	5	14	5	21	5	Gage 19 Fault Flag	0 = Pass 1 = Fail	
22	6	6	6	14	6	21	6	Gage 20 Warning Flag	0 = Pass 1 = Fail	
22	7	6	7	14	7	21	7	Gage 20 Fault Flag	0 = Pass 1 = Fail	

Table B.1
Discrete Bit Inputs (Cont'd)

PI Backplane		Remote I/O		RS–232 ASCII		RS–232 DF1		Definition	Usage	Notes
Word #	Bit #	Word #	Bit #*	Hex #	Bit #	Byte #	Bit #			
22	8	6	10	15	0	20	0	Gage 21 Warning Flag	0 = Pass 1 = Fail	
22	9	6	11	15	1	20	1	Gage 21 Fault Flag	0 = Pass 1 = Fail	
22	10	6	12	15	2	20	2	Gage 22 Warning Flag	0 = Pass 1 = Fail	
22	11	6	13	15	3	20	3	Gage 22 Fault Flag	0 = Pass 1 = Fail	
22	12	6	14	15	4	20	4	Gage 23 Warning Flag	0 = Pass 1 = Fail	
22	13	6	15	15	5	20	5	Gage 23 Fault Flag	0 = Pass 1 = Fail	
22	14	6	16	15	6	20	6	Gage 24 Warning Flag	0 = Pass 1 = Fail	
22	15	6	17	15	7	20	7	Gage 24 Fault Flag	0 = Pass 1 = Fail	
23	0	7	0	15	0	23	0	Gage 25 Warning Flag	0 = Pass 1 = Fail	
23	1	7	1	16	1	23	1	Gage 25 Fault Flag	0 = Pass 1 = Fail	
23	2	7	2	16	2	23	2	Gage 26 Warning Flag	0 = Pass 1 = Fail	
23	3	7	3	16	3	23	3	Gage 26 Fault Flag	0 = Pass 1 = Fail	
23	4	7	4	16	4	23	4	Gage 27 Warning Flag	0 = Pass 1 = Fail	
23	5	7	5	16	5	23	5	Gage 27 Fault Flag	0 = Pass 1 = Fail	
23	6	7	6	16	6	23	6	Gage 28 Warning Flag	0 = Pass 1 = Fail	
23	7	7	7	16	7	23	7	Gage 28 Fault Flag	0 = Pass 1 = Fail	

Table B.1
Discrete Bit Inputs (Cont'd)

PI Backplane		Remote I/O		RS–232 ASCII		RS–232 DF1		Definition	Usage	Notes
Word #	Bit #	Word #	Bit #*	Hex #	Bit #	Byte #	Bit #			
23	8	7	10	17	0	22	0	Gage 29 Warning Flag	0 = Pass 1 = Fail	
23	9	7	11	17	1	22	1	Gage 29 Fault Flag	0 = Pass 1 = Fail	
23	10	7	12	17	2	22	2	Gage 30 Warning Flag	0 = Pass 1 = Fail	
23	11	7	13	17	3	22	3	Gage 30 Fault Flag	0 = Pass 1 = Fail	
23	12	7	14	17	4	22	4	Gage 31 Warning Flag	0 = Pass 1 = Fail	
23	13	7	15	17	5	22	5	Gage 31 Fault Flag	0 = Pass 1 = Fail	
23	14	7	16	17	6	22	6	Gage 32 Warning Flag	0 = Pass 1 = Fail	
23	15	7	17	17	7	22	7	Gage 32 Fault Flag	0 = Pass 1 = Fail	
24	0	0	0	18	0	25	0	Not Used		
24	1	0	1	18	1	25	1	Configuration Error	0 = No Error 1 = Error	Configuration error bit is set after any invalid configuration block write to the Color CVIM. This flag is set after validation errors.
24	2	0	2	18	2	25	2	Mastership Flag	0 = Not Master 1 = Master	The device which reads this bit as 1 is the host. Not applicable for RS–232 communications.
24	3	0	3	18	3	25	3	Module Busy	0 = Not Busy 1 = Busy	Module Busy bit is set during the SETUP mode and while sending a configuration download to the Color CVIM.
24	4	0	4	18	4	25	4	Trigger NAK	0 = OK 1 = Trigger 1 Missed	
24	5	0	5	18	5	25	5	Post Results	0 = Part 1 1 = Part 2	
24	6	0	6	18	6	25	6	Toolset Data Valid	0 = Not Valid 1 = Results Valid	Data Valid bit is reset when a user enters the SETUP mode and when pulsed output duration has expired Refer to Chapter 3.
24	7	0	7	18	7	25	7	Not Used		

Table B.1
Discrete Bit Inputs (Cont'd)

PI Backplane		Remote I/O		RS–232 ASCII		RS–232 DF1		Definition	Usage	Notes
Word #	Bit #	Word #	Bit #*	Hex #	Bit #	Byte #	Bit #			
24	8	0	10	19	0	24	0	Reference Line 1 Flag	0 = Pass 1 = Fail	
24	9	0	11	19	1	24	1	Reference Line 2 Flag	0 = Pass 1 = Fail	
24	10	0	12	19	2	24	2	Reference Line 3 Flag	0 = Pass 1 = Fail	
24	11	0	13	19	3	24	3	Reference Window 1 Flag	0 = Pass 1 = Fail	
24	12	0	14	19	4	24	4	Reference Window 2 Flag	0 = Pass 1 = Fail	
24	13	0	15	19	5	24	5	Reference Window 3 Flag	0 = Pass 1 = Fail	
24	14	0	16	19	6	24	6	Light Probe Flag	0 = No Fault 1 = Fault	
24	15	0	17	19	7	24	7	Master Fault	0 = Pass 1 = Fail	Master Fault bit is set if any tool fails an inspection.
25	0	1	0	20	0	27	0	Window 25 Warning Flag	0 = Pass 1 = Fail	
25	1	1	1	20	1	27	1	Window 25 Fault Flag	0 = Pass 1 = Fail	
25	2	1	2	20	2	27	2	Window 26 Warning Flag	0 = Pass 1 = Fail	
25	3	1	3	20	3	27	3	Window 26 Fault Flag	0 = Pass 1 = Fail	
25	4	1	4	20	4	27	4	Window 27 Warning Flag	0 = Pass 1 = Fail	
25	5	1	5	20	5	27	5	Window 27 Fault Flag	0 = Pass 1 = Fail	
25	6	1	6	20	6	27	6	Window 28 Warning Flag	0 = Pass 1 = Fail	
25	7	1	7	20	7	27	7	Window 28 Fault Flag	0 = Pass 1 = Fail	

Table B.1
Discrete Bit Inputs (Cont'd)

PI Backplane		Remote I/O		RS–232 ASCII		RS–232 DF1		Definition	Usage	Notes
Word #	Bit #	Word #	Bit #*	Hex #	Bit #	Byte #	Bit #			
25	8	1	10	21	0	26	0	Window 29 Warning Flag	0 = Pass 1 = Fail	
25	9	1	11	21	1	26	1	Window 29 Fault Flag	0 = Pass 1 = Fail	
25	10	1	12	21	2	26	2	Window 30 Warning Flag	0 = Pass 1 = Fail	
25	11	1	13	21	3	26	3	Window 30 Fault Flag	0 = Pass 1 = Fail	
25	12	1	14	21	4	26	4	Window 31 Warning Flag	0 = Pass 1 = Fail	
25	13	1	15	21	5	26	5	Window 31 Fault Flag	0 = Pass 1 = Fail	
25	14	1	16	21	6	26	6	Window 32 Warning Flag	0 = Pass 1 = Fail	
25	15	1	17	21	7	26	7	Window 32 Fault Flag	0 = Pass 1 = Fail	
26	0	2	0	22	0	29	0	Window 33 Warning Flag	0 = Pass 1 = Fail	
26	1	2	1	22	1	29	1	Window 33 Fault Flag	0 = Pass 1 = Fail	
26	2	2	2	22	2	29	2	Window 34 Warning Flag	0 = Pass 1 = Fail	
26	3	2	3	22	3	29	3	Window 34 Fault Flag	0 = Pass 1 = Fail	
26	4	2	4	22	4	29	4	Window 35 Warning Flag	0 = Pass 1 = Fail	
26	5	2	5	22	5	29	5	Window 35 Fault Flag	0 = Pass 1 = Fail	
26	6	2	6	22	6	29	6	Window 36 Warning Flag	0 = Pass 1 = Fail	
26	7	2	7	22	7	29	7	Window 36 Fault Flag	0 = Pass 1 = Fail	

Table B.1
Discrete Bit Inputs (Cont'd)

PI Backplane		Remote I/O		RS–232 ASCII		RS–232 DF1		Definition	Usage	Notes
Word #	Bit #	Word #	Bit #*	Hex #	Bit #	Byte #	Bit #			
26	8	2	10	23	0	28	0	Window 37 Warning Flag	0 = Pass 1 = Fail	
26	9	2	11	23	1	28	1	Window 37 Fault Flag	0 = Pass 1 = Fail	
26	10	2	12	23	2	28	2	Window 38 Warning Flag	0 = Pass 1 = Fail	
26	11	2	13	23	3	28	3	Window 38 Fault Flag	0 = Pass 1 = Fail	
26	12	2	14	23	4	28	4	Window 39 Warning Flag	0 = Pass 1 = Fail	
26	13	2	15	23	5	28	5	Window 39 Fault Flag	0 = Pass 1 = Fail	
26	14	2	16	23	6	28	6	Window 40 Warning Flag	0 = Pass 1 = Fail	
26	15	2	17	23	7	28	7	Window 40 Fault Flag	0 = Pass 1 = Fail	
27	0	3	0	24	0	31	0	Window 41 Warning Flag	0 = Pass 1 = Fail	
27	1	3	1	24	1	31	1	Window 41 Fault Flag	0 = Pass 1 = Fail	
27	2	3	2	24	2	31	2	Window 42 Warning Flag	0 = Pass 1 = Fail	
27	3	3	3	24	3	31	3	Window 42 Fault Flag	0 = Pass 1 = Fail	
27	4	3	4	24	4	31	4	Window 43 Warning Flag	0 = Pass 1 = Fail	
27	5	3	5	24	5	31	5	Window 43 Fault Flag	0 = Pass 1 = Fail	
27	6	3	6	24	6	31	6	Window 44 Warning Flag	0 = Pass 1 = Fail	
27	7	3	7	24	7	31	7	Window 44 Fault Flag	0 = Pass 1 = Fail	

**Table B.1
Discrete Bit Inputs (Cont'd)**

PI Backplane		Remote I/O		RS–232 ASCII		RS–232 DF1		Definition	Usage	Notes
Word #	Bit #	Word #	Bit #*	Hex #	Bit #	Byte #	Bit #			
27	8	3	10	25	0	30	0	Window 45 Warning Flag	0 = Pass 1 = Fail	
27	9	3	11	25	1	30	1	Window 45 Fault Flag	0 = Pass 1 = Fail	
27	10	3	12	25	2	30	2	Window 46 Warning Flag	0 = Pass 1 = Fail	
27	11	3	13	25	3	30	3	Window 46 Fault Flag	0 = Pass 1 = Fail	
27	12	3	14	25	4	30	4	Window 47 Warning Flag	0 = Pass 1 = Fail	
27	13	3	15	25	5	30	5	Window 47 Fault Flag	0 = Pass 1 = Fail	
27	14	3	16	25	6	30	6	Window 48 Warning Flag	0 = Pass 1 = Fail	
27	15	3	17	25	7	30	7	Window 48 Fault Flag	0 = Pass 1 = Fail	
28	0	4	0	26	0	33	0	Gage 33 Warning Flag	0 = Pass 1 = Fail	
28	1	4	1	26	1	33	1	Gage 33 Fault Flag	0 = Pass 1 = Fail	
28	2	4	2	26	2	33	2	Gage 34 Warning Flag	0 = Pass 1 = Fail	
28	3	4	3	26	3	33	3	Gage 34 Fault Flag	0 = Pass 1 = Fail	
28	4	4	4	26	4	33	4	Gage 35 Warning Flag	0 = Pass 1 = Fail	
28	5	4	5	26	5	33	5	Gage 35 Fault Flag	0 = Pass 1 = Fail	
28	6	4	6	26	6	33	6	Gage 36 Warning Flag	0 = Pass 1 = Fail	
28	7	4	7	26	7	33	7	Gage 36 Fault Flag	0 = Pass 1 = Fail	

Table B.1
Discrete Bit Inputs (Cont’d)

PI Backplane		Remote I/O		RS–232 ASCII		RS–232 DF1		Definition	Usage	Notes
Word #	Bit #	Word #	Bit #*	Hex #	Bit #	Byte #	Bit #			
28	8	4	10	27	0	32	0	Gage 37 Warning Flag	0 = Pass 1 = Fail	
28	9	4	11	27	1	32	1	Gage 37 Fault Flag	0 = Pass 1 = Fail	
28	10	4	12	27	2	32	2	Gage 38 Warning Flag	0 = Pass 1 = Fail	
28	11	4	13	27	3	32	3	Gage 38 Fault Flag	0 = Pass 1 = Fail	
28	12	4	14	27	4	32	4	Gage 39 Warning Flag	0 = Pass 1 = Fail	
28	13	4	15	27	5	32	5	Gage 39 Fault Flag	0 = Pass 1 = Fail	
28	14	4	16	27	6	32	6	Gage 40 Warning Flag	0 = Pass 1 = Fail	
28	15	4	17	27	7	32	7	Gage 40 Fault Flag	0 = Pass 1 = Fail	
29	0	5	0	28	0	35	0	Gage 41 Warning Flag	0 = Pass 1 = Fail	
29	1	5	1	28	1	35	1	Gage 41 Fault Flag	0 = Pass 1 = Fail	
29	2	5	2	28	2	35	2	Gage 42 Warning Flag	0 = Pass 1 = Fail	
29	3	5	3	28	3	35	3	Gage 42 Fault Flag	0 = Pass 1 = Fail	
29	4	5	4	28	4	35	4	Gage 43 Warning Flag	0 = Pass 1 = Fail	
29	5	5	5	28	5	35	5	Gage 43 Fault Flag	0 = Pass 1 = Fail	
29	6	5	6	28	6	35	6	Gage 44 Warning Flag	0 = Pass 1 = Fail	
29	7	5	7	28	7	35	7	Gage 44 Fault Flag	0 = Pass 1 = Fail	

Table B.1
Discrete Bit Inputs (Cont’d)

PI Backplane		Remote I/O		RS–232 ASCII		RS–232 DF1		Definition	Usage	Notes
Word #	Bit #	Word #	Bit #*	Hex #	Bit #	Byte #	Bit #			
29	8	5	10	29	0	34	0	Gage 45 Warning Flag	0 = Pass 1 = Fail	
29	9	5	11	29	1	34	1	Gage 45 Fault Flag	0 = Pass 1 = Fail	
29	10	5	12	29	2	34	2	Gage 46 Warning Flag	0 = Pass 1 = Fail	
29	11	5	13	29	3	34	3	Gage 46 Fault Flag	0 = Pass 1 = Fail	
29	12	5	14	29	4	34	4	Gage 47 Warning Flag	0 = Pass 1 = Fail	
29	13	5	15	29	5	34	5	Gage 47 Fault Flag	0 = Pass 1 = Fail	
29	14	5	16	29	6	34	6	Gage 48 Warning Flag	0 = Pass 1 = Fail	
29	15	5	17	29	7	34	7	Gage 48 Fault Flag	0 = Pass 1 = Fail	
30	0	6	0	30	0	37	0	Gage 49 Warning Flag	0 = Pass 1 = Fail	
30	1	6	1	30	1	37	1	Gage 49 Fault Flag	0 = Pass 1 = Fail	
30	2	6	2	30	2	37	2	Gage 50 Warning Flag	0 = Pass 1 = Fail	
30	3	6	3	30	3	37	3	Gage 50 Fault Flag	0 = Pass 1 = Fail	
30	4	6	4	30	4	37	4	Gage 51 Warning Flag	0 = Pass 1 = Fail	
30	5	6	5	30	5	37	5	Gage 51 Fault Flag	0 = Pass 1 = Fail	
30	6	6	6	30	6	37	6	Gage 52 Warning Flag	0 = Pass 1 = Fail	
30	7	6	7	30	7	37	7	Gage 52 Fault Flag	0 = Pass 1 = Fail	

Table B.1
Discrete Bit Inputs (Cont'd)

PI Backplane		Remote I/O		RS–232 ASCII		RS–232 DF1		Definition	Usage	Notes
Word #	Bit #	Word #	Bit #*	Hex #	Bit #	Byte #	Bit #			
30	8	6	10	31	0	36	0	Gage 53 Warning Flag	0 = Pass 1 = Fail	
30	9	6	11	31	1	36	1	Gage 53 Fault Flag	0 = Pass 1 = Fail	
30	10	6	12	31	2	36	2	Gage 54 Warning Flag	0 = Pass 1 = Fail	
30	11	6	13	31	3	36	3	Gage 54 Fault Flag	0 = Pass 1 = Fail	
30	12	6	14	31	4	36	4	Gage 55 Warning Flag	0 = Pass 1 = Fail	
30	13	6	15	31	5	36	5	Gage 55 Fault Flag	0 = Pass 1 = Fail	
30	14	6	16	31	6	36	6	Gage 56 Warning Flag	0 = Pass 1 = Fail	
30	15	6	17	31	7	36	7	Gage 56 Fault Flag	0 = Pass 1 = Fail	
31	0	7	0	32	0	39	0	Gage 57 Warning Flag	0 = Pass 1 = Fail	
31	1	7	1	32	1	39	1	Gage 57 Fault Flag	0 = Pass 1 = Fail	
31	2	7	2	32	2	39	2	Gage 58 Warning Flag	0 = Pass 1 = Fail	
31	3	7	3	32	3	39	3	Gage 58 Fault Flag	0 = Pass 1 = Fail	
31	4	7	4	32	4	39	4	Gage 59 Warning Flag	0 = Pass 1 = Fail	
31	5	7	5	32	5	39	5	Gage 59 Fault Flag	0 = Pass 1 = Fail	
31	6	7	6	32	6	39	6	Gage 60 Warning Flag	0 = Pass 1 = Fail	
31	7	7	7	32	7	39	7	Gage 60 Fault Flag	0 = Pass 1 = Fail	

Table B.1
Discrete Bit Inputs (Cont'd)

PI Backplane		Remote I/O		RS–232 ASCII		RS–232 DF1		Definition	Usage	Notes
Word #	Bit #	Word #	Bit #*	Hex #	Bit #	Byte #	Bit #			
31	8	7	10	33	0	38	0	Gage 61 Warning Flag	0 = Pass 1 = Fail	
31	9	7	11	33	1	38	1	Gage 61 Fault Flag	0 = Pass 1 = Fail	
31	10	7	12	33	2	38	2	Gage 62 Warning Flag	0 = Pass 1 = Fail	
31	11	7	13	33	3	38	3	Gage 62 Fault Flag	0 = Pass 1 = Fail	
31	12	7	14	33	4	38	4	Gage 63 Warning Flag	0 = Pass 1 = Fail	
31	13	7	15	33	5	38	5	Gage 63 Fault Flag	0 = Pass 1 = Fail	
31	14	7	16	33	6	38	6	Gage 64 Warning Flag	0 = Pass 1 = Fail	
31	15	7	17	33	7	38	7	Gage 64 Fault Flag	0 = Pass 1 = Fail	

Discrete Bit Outputs

There are 128 bits that can be set as outputs from a host device to control the operation of the Color CVIM. These bits control:

- Monitor display.
- Camera trigger.
- Toolset selection.
- I/O forcing.
- Selection of operation after reject.
- Memory storage location. RAM, EEPROM, RAM Card, or external host memory.

Refer to Tables B.2 and B.3. Table B.2 only applies to Backplane communications. Table B.3 applies to both Backplane and Remote I/O communications.

Table B.2
Backplane Handshake Bits

Word #		Bit # PI	Definition	Usage	Notes
PI Backplane	RS–232 and Remote I/O				
0	N/A	0	Host Data Lock	0 = Color CVIM May Write 1 = Color CVIM Write Inhibited	The backplane host should write to bit 0 to prevent the Color CVIM from modifying results data.
0	N/A	1	Reserved		
0	N/A	2	Color CVIM Data Lock	0 = Color CVIM Not Writing 1 = Color CVIM Writing	The Color CVIM sets bit 2 while writing. The bit will be set back to 0 after writing.
0	N/A	3	Reserved		
0	N/A	4–15	Reserved		

Discrete Bit Outputs (cont'd)

Note: When using the backplane, don't write directly to word 2. Write to word 1023. The Color CVIM will copy the data from word 1023 to word 2.

Note: When communicating with a device through the Pyramid Integrator backplane, Color CVIM module results are posted in shared memory immediately after processing. When communicating with a device through the other ports, results are only available at the end of the inspection program.

Note: The Remote I/O Bit Number column (marked with an asterisk in the table below) are expressed in Octal. All other values are expressed in decimal.

Note: With RS–232, use commands based on the protocol you are using (DF1 or ASCII).

Table B.3
Discrete Bit Outputs

PI Backplane		Remote I/O		Definition	Usage	Notes
Word #	Bit #	Word #	Bit #			
2	0–7	0	0–7	Not Used		
2	8	0	10	Lock Request	0 = No Change 1 = Lock	Do not set lock and unlock at the same time. If you set both bits, the Color CVIM will be unlocked. You must reset this bit to 0, then back to 1 to repeat a lock request.
2	9	0	11	Unlock Request	0 = No Change 1 = Unlock	Do not set lock and unlock at the same time. If you set both bits, the Color CVIM will be unlocked. You must reset this bit to 0, then back to 1 to repeat an unlock request.
2	10	0	12	Host Trigger	0 = No Trigger 1 = Trigger	You must reset this bit to 0, then back to 1 to repeat a trigger request.
2	11	0	13	Not Used		
2	12	0	14	Light Pen Request	0 = No Request 1 = Request	Lightpen request commands are specified in: Output words 1 and 6 (Remote I/O) Output words 3 and 8 (Backplane)You must reset bit to 0, then back to 1 to repeat a light pen request.
2	13	0	15	I/O Request	0 = No Request 1 = Request	I/O Request commands are specified in: Output words 2, 4, and 5 (Remote I/O) Output words 4, 6 and 7 (Backplane)You must reset bit to 0, then back to 1 to repeat an I/O request.

Table B.3
Discrete Bit Outputs (Cont'd)

PI Backplane		Remote I/O		Definition	Usage	Notes
Word #	Bit #	Word #	Bit #			
2	14	0	16	Configuration Move	0 = No Request 1 = Request	Configuration Move commands are specified in: Output word 2 (Remote I/O) Output word 4 (Backplane)You must reset bit to 0, then back to 1 to repeat a configuration move request.
2	15	0	17	Not Used		
3	0–7	1	0–7	Runtime Display Control	00000000 = No Change 00000001 = Display Image Only 00000010 = Display Failed Tools 00000100 = Display All Tools 000010000 = Display I/O Page 00010000 = Display Results Page 00100000 = Display Statistics 1 Page 01000000 = Display Statistics 2 Page	
3	8–15	1	10–17	Freeze On Reject Control	00000000 = No Change 00000001 = Go On Reject 00000010 = Freeze First Reject 00000100 = Freeze All Rejects 00001000 = Freeze Next Image 00010000 = Halt On Reject	
4 Not Applicable to Backplane	N/A	2	0–3	Post Results to Remote I/O Flag	0000 = Not Valid 0001 = Part 1 0010 = Part 2	Post Results to Remote I/O Flag controls which part of the results are assigned to discrete input words 1 through 7. Remote I/O only.
4	4–5	2	4–5	Discrete I/O Control	00 = No Change 01 = Disable Outputs 10 = Enable Outputs	Discrete and Forced I/O Control refer to the local I/O module (Catalog No. 2801–JMB).
4	6–7	2	6–7	Forced I/O Control	00 = No Change 01 = Disable Forces 10 = Enable Forces	Forced I/O Control acts upon the bitmap you setup in words: 4 and 5 (Remote I/O) 6 and 7 (Backspace)

Table B.3
Discrete Bit Outputs (Cont'd)

PI Backplane		Remote I/O		Definition	Usage	Notes
Word #	Bit #	Word #	Bit #			
4	8–11	2	10–13	Configuration Move Control	0000 = No 0001 = EEPROM to RAM 0010 = RAM to Request EEPROM 0100 = RAM Card to RAM 1000 = RAM to RAM Card	
4	12–15	2	14–17	RAM Card Index	0000 = RAM Card 11111 = RAM Card Configuration Configuration 16	Numeric fields which specify RAM card locations 1 through 16.
5 Reference Only. Not used for Backplane.	N/A	3	0–2	Block Transfer Type	000 = Invalid Request 001 = Results Block 010 = Configuration Block 101 = Statistics Block – See Notes 111 = Programmable Block Transfer – See Notes.	This word only applies to Remote I/O interface. To send the programmable block transfer to the Color CVIM, set bits 0, 1, and 2 to equal 111 and then send a block transfer write. No other bits (last block, toolset, or block number) need to be sent. After sending a configuration block or template block to the Color CVIM, you should check the condition of the configuration fault bit (Word 0, Discrete Bit inputs). To send programmable statistics, set bits 0, 1, and 2 to equal 101 and then send a block transfer write to the Color CVIM. No other bits (last block, toolset, or block number) need to be sent.
5 Reference Only. Not used for Backplane.	N/A	3	3	Last Block Flag	0 = Not Last Block 1 = Last Block	Applies only to block transfer writes.
5 Reference Only. Not used for Backplane.	N/A	3	4–7	Toolset Request	0001 = Toolset Active	Applies only to results blocks. Must be 0001 for all Block Transfers.
5 Reference Only. Not used for Backplane.	N/A	3	10–17	Block Transfer Block Number	0 = Invalid Number See Notes.	Results block numbers may be from 1 to 4. Configuration block numbers may be from 1 to 136. Template block numbers may be from 1 to 255 (variable).

Table B.3
Discrete Bit Outputs (Cont'd)

PI Backplane		Remote I/O		Definition	Usage	Notes
Word #	Bit #	Word #	Bit #			
6	0	4	0	Local I/O Output 1 Force On	0 = No Force 1 = Force On	Refers to Catalog No. 2801–JMB Local I/O Board.
6	1	4	1	Local I/O Output 2 Force On	0 = No Force 1 = Force On	Refers to Catalog No. 2801–JMB Local I/O Board.
6	2	4	2	Local I/O Output 3 Force On	0 = No Force 1 = Force On	Refers to Catalog No. 2801–JMB Local I/O Board.
6	3	4	3	Local I/O Output 4 Force On	0 = No Force 1 = Force On	Refers to Catalog No. 2801–JMB Local I/O Board.
6	4	4	4	Local I/O Output 5 Force On	0 = No Force 1 = Force On	Refers to Catalog No. 2801–JMB Local I/O Board.
6	5	4	5	Local I/O Output 6 Force On	0 = No Force 1 = Force On	Refers to Catalog No. 2801–JMB Local I/O Board.
6	6	4	6	Local I/O Output 7 Force On	0 = No Force 1 = Force On	Refers to Catalog No. 2801–JMB Local I/O Board.
6	7	4	7	Local I/O Output 8 Force On	0 = No Force 1 = Force On	Refers to Catalog No. 2801–JMB Local I/O Board.
6	8	4	10	Local I/O Output 9 Force On	0 = No Force 1 = Force On	Refers to Catalog No. 2801–JMB Local I/O Board.
6	9	4	11	Local I/O Output 10 Force On	0 = No Force 1 = Force On	Refers to Catalog No. 2801–JMB Local I/O Board.
6	10	4	12	Local I/O Output 11 Force On	0 = No Force 1 = Force On	Refers to Catalog No. 2801–JMB Local I/O Board.
6	11	4	13	Local I/O Output 12 Force On	0 = No Force 1 = Force On	Refers to Catalog No. 2801–JMB Local I/O Board.
6	12	4	14	Local I/O Output 13 Force On	0 = No Force 1 = Force On	Refers to Catalog No. 2801–JMB Local I/O Board.
6	13	4	15	Local I/O Output 14 Force On	0 = No Force 1 = Force On	Refers to Catalog No. 2801–JMB Local I/O Board.
6	14–15	4	16–17	Not Used		

Table B.3
Discrete Bit Outputs (Cont'd)

PI Backplane		Remote I/O		Definition	Usage	Notes
Word #	**Bit #**	**Word #**	**Bit #**			
7	0	5	0	Local I/O Output 1 Force OFF	0 = No Force 1 = Force Off	Refers to Catalog No. 2801–JMB Local I/O Board.
7	1	5	1	Local I/O Output 2 Force OFF	0 = No Force 1 = Force Off	Refers to Catalog No. 2801–JMB Local I/O Board.
7	2	5	2	Local I/O Output 3 Force OFF	0 = No Force 1 = Force Off	Refers to Catalog No. 2801–JMB Local I/O Board.
7	3	5	3	Local I/O Output 4 Force OFF	0 = No Force 1 = Force Off	Refers to Catalog No. 2801–JMB Local I/O Board.
7	4	5	4	Local I/O Output 5 Force OFF	0 = No Force 1 = Force Off	Refers to Catalog No. 2801–JMB Local I/O Board.
7	5	5	5	Local I/O Output 6 Force OFF	0 = No Force 1 = Force Off	Refers to Catalog No. 2801–JMB Local I/O Board.
7	6	5	6	Local I/O Output 7 Force OFF	0 = No Force 1 = Force Off	Refers to Catalog No. 2801–JMB Local I/O Board.
7	7	5	7	Local I/O Output 8 Force OFF	0 = No Force 1 = Force Off	Refers to Catalog No. 2801–JMB Local I/O Board.
7	8	5	10	Local I/O Output 9 Force OFF	0 = No Force 1 = Force Off	Refers to Catalog No. 2801–JMB Local I/O Board.
7	9	5	11	Local I/O Output 10 Force OFF	0 = No Force 1 = Force Off	Refers to Catalog No. 2801–JMB Local I/O Board.
7	10	5	12	Local I/O Output 11 Force OFF	0 = No Force 1 = Force Off	Refers to Catalog No. 2801–JMB Local I/O Board.
7	11	5	13	Local I/O Output 12 Force OFF	0 = No Force 1 = Force Off	Refers to Catalog No. 2801–JMB Local I/O Board.
7	12	5	14	Local I/O Output 13 Force OFF	0 = No Force 1 = Force Off	Refers to Catalog No. 2801–JMB Local I/O Board.
7	13	5	15	Local I/O Output 14 Force OFF	0 = No Force 1 = Force Off	Refers to Catalog No. 2801–JMB Local I/O Board.
7	14–15	5	16–17	Not Used		

Table B.3
Discrete Bit Outputs (Cont'd)

PI Backplane		Remote I/O		Definition	Usage	Notes
Word #	Bit #	Word #	Bit #			
8	0–1	6	0–1	Not Used		
8	2	6	2	Not Used		
8	3	6	3	Resume Control	0 = No Change 1 = Resume	Use in conjunction with light pen request: Word 0 bit 14 (Remote I/O) Word 2 bit 12 (Backplane)
8	4–5	6	4–5	Page Control	00 = No Change 01 = Page Up 10 = Page Down	Use in conjunction with light pen request: Word 0 bit 14 (Remote I/O) Word 2 bit 12 (Backplane)
8	6	6	6	Reset Statistics	0 = No Change 1 = Reset	Use in conjunction with light pen request: Word 0 bit 14 (Remote I/O) Word 2 bit 12 (Backplane)
8	7	6	7	Reset Counters	0 = No Change 1 = Reset	Use in conjunction with light pen request: Word 0 bit 14 (Remote I/O) Word 2 bit 12 (Backplane)
8	8–15	6	10–17	Not Used		

Appendix C

Numerical Results Data

Results Block Overview

There are 6 results blocks for inspection. The following is an overview of the blocks.

Block Number 1 Contains:

- Block transfer signature
- Discrete output copy
- X/Y reference lines #1, #2, and #3
- Feature finder #1
- Windows 1–8
- Gages 1–8
- Total tumber of triggers, missed triggers, and master faults

Block #2 Contains:

- Block transfer signature
- Windows 9–16
- Gages 9–22
- Windows 17–24
- Total number of triggers

Block #3 Contains:

- Block transfer signature
- Gages 23–32
- Feature finders #2 and #3
- Brightness probe
- Total number of triggers, missed triggers, and master faults

Block #4 Contains:

- Block transfer signature
- Windows 25–36
- Gages 33–44
- Windows 37–42
- Total number of triggers

Results Block Overview (cont'd)

Block #5 Contains:

- Block transfer signature
- Gages 45–56
- Windows 43–48
- Gages 57–64
- Total number of triggers, missed triggers, and master faults

Block #6 Contains:

- Block transfer signature
- Status word
- User programmable block transfer data
- Total number of triggers

Block Transfer Signature

The block transfer signature is for user information only. The Color CVIM places the signature in each block sent to the PLC for identification and does not care if the PLC changes the signature prior to sending a block back to the Color CVIM.Bits 0–7 designate the block number:

00000000 = Not Valid
00000001 = Block #1
00000010 = Block #2
. . .
11111111 = Block #255

Bits 8–10 designate the toolset:

000 = Not Valid
001 = Toolset
010 = Not used
011 to 111 = Reserved

Bits 11 and 12 designate the block type:

00 = Results
01 = Configuration
10 = Templates
11 = Statistics

Bits 13–15 specify the module thumbwheel number:

000 to 111

Results Block #1

Table C.1 shows the function of each word in results block #1.

Table C.1
Numerical Results Data–Results Block 1

Word #		Bit #	Definition	Usage	Notes
PI Backplane	RS–232 and Remote I/O				
32	0	0–15	Block Transfer Signature		
33	1	0–15	Reserved		
34–35	2–3	0–15	Reserved		
36	4	0–15	Reference Line #1 X Position		From upper left corner. 16 bit integer.
37	5	0–15	Reference Line #1 Y Position		From upper left corner of display. 16 bit integer.
38	6	0–15	Reference Line #2 X Position		From upper left corner of display. 16 bit integer.
39	7	0–15	Reference Line #2 Y Position		From upper left corner of display. 16 bit integer.
40	8	0–15	Reference Line #3 X Position		From upper left corner of display. 16 bit integer.
41	9	0–15	Reference Line #3 Y Position		From upper left corner of display. 16 bit integer.
42	10	0–15	Reference Window #1 X1 Position		Feature #1. From upper left corner. 16 bit integer.
43	11	0–15	Reference Window #1 Y1 Position		Feature #1. From upper left corner. 16 bit integer.
44	12	0–15	Reference Window #1 X2 Position		Feature #2. From upper left corner. 16 bit integer.
45	13	0–15	Reference Window #1 Y2 Position		Feature #2. From upper left corner. 16 bit integer.
46	14	0–15	Reference Window #1 X3 Position		Feature #3. From upper left corner of display. 16 bit integer.
47	15	0–15	Reference Window #1 Y3 Position		Feature #3. From upper left corner of display. 16 bit integer.
48	16	0–15	Reference Window #1 X–Center		Centroid of enabled feature. 16 bit integer.

Results Block #1 (cont'd)

Table C.1
Numerical Results Data–Results Block 1 (cont'd)

Word #		Bit #	Definition	Usage	Notes
PI Backplane	RS–232 and Remote I/O				
49	17	0–15	Reference Window #1 Y–Center		Centroid of enabled feature. 16 bit integer.
50	18	0–15	Reference Window #1 Theta Integer		16.16 value. Refer to Appendix A. Only if 2 or 3 Features are enabled.
51	19	0–15	Reference Window #1 Theta Fraction		Only if 2 or 3 Features are enabled.
52	20	0	Reference Window #1 X1/Y1 Pass/Fail Bit	0 = Pass 1 = Fail	Feature 1
52	20	1	Reference Window #1 X2/Y2 Pass/Fail Bit	0 = Pass 1 = Fail	Feature 2
52	20	2	Reference Window #1 X3/Y3 Pass/Fail Bit	0 = Pass 1 = Fail	Feature 3
52	20	3	Combination Pass/Fail Bit	0 = Pass 1 = Fail	0 if all enabled features pass.
52	20	4–15	Not Used		
53	21	0–15	Score Reference Window #1 Feature #1		
54	22	0–15	Score Reference Window #1 Feature #2		
55	23	0–15	Score Reference Window #1 Feature #3		
56–57	24–25	0–15	Window #1 Value		Luminance–16.16 Object–32 bit integer Pixel–32 bit integer Color Match–32 bit integer Color ID–32 bit integer Gradient–32 bit integer
58–59	26–27	0–15	Window #2 Value		Same as window #1
60–61	28–29	0–15	Window #3 Value		Same as window #1
62–63	30–31	0–15	Window #4 Value		Same as window #1
64–65	32–33	0–15	Window #5 Value		Same as window #1
66–67	34–35	0–15	Window #6 Value		Same as window #1
68–69	36–37	0–15	Window #7 Value		Same as window #1
70–71	38–39	0–15	Window #8 Value		Same as window #1

Results Block #1 (cont'd)

Table C.1
Numerical Results Data–Results Block 1 (cont'd)

Word #		Bit #	Definition	Usage	Notes
PI Backplane	RS–232 and Remote I/O				
72–73	40–41	0–15	Gage #1 Value		Angular & Linear Measure–16.16 Edge–32 bit integer Object–32 bit integer Pixel–32 bit integer X Position & Y Position–16.16
74–75	42–43	0–15	Gage #2 Value		Same as gage #1
76–77	44–45	0–15	Gage #3 Value		Same as gage #1
78–79	46–47	0–15	Gage #4 Value		Same as gage #1
80–81	48–49	0–15	Gage #5 Value		Same as gage #1
82–83	50–51	0–15	Gage #6 Value		Same as gage #1
84–85	52–53	0–15	Gage #7 Value		Same as gage #1
86–87	54–55	0–15	Gage #8 Value		Same as gage #1
88	56	0–15	Reserved		
89	57	0–15	Reserved		
90	58	0–15	Reserved		
91	59	0–15	Missed Triggers		
92	60	0–15	Task Master Faultsr — Most Significant Word		32 bit integer. Value is incremented if at least one tool fails.
93	61	0–15	Task Master Faultsr — Least Significant Word		
94	62	0–15	Task Triggersr — Most Significant Word		32 bit integer.
95	63	0–15	Task Triggersr — Least Significant Word		

Results Block #2

Table C.2 shows the function of each word in results block #2.

Table C.2
Numerical Results Data–Results Block 2

Word #		Bit #	Definition	Usage	Notes
PI Backplane	RS–232 and Remote I/O				
96	0	0–15	Block Transfer Signature		
97	1	0–15	Reserved		
98–99	2–3	0–15	Window #9 Value		Luminance–16.16 Object–32 bit integer Pixel–32 bit integer Color ID–32 bit integer Color Match–32 bit integer Gradient–32 bit integer
100–101	4–5	0–15	Window #10 Value		Same as window #9
102–103	6–7	0–15	Window #11 Value		Same as window #9
104–105	8–9	0–15	Window #12 Value		Same as window #9
106–107	10–11	0–15	Window #13 Value		Same as window #9
108–109	12–13	0–15	Window #14 Value		Same as window #9
110–111	14–15	0–15	Window #15 Value		Same as window #9
112–113	16–17	0–15	Window #16 Value		Same as window #9
114–115	18–19	0–15	Gage #9 Value		Angular & Linear Measure 16.16 Edge–32 bit integer Object–32 bit integer Pixel–32 bit integer X Position & Y Position–16.16
116–117	20–21	0–15	Gage #10 Value		Same as gage #9
118–119	22–23	0–15	Gage #11 Value		Same as gage #9

Results Block #2 (cont'd)

Table C.2
Numerical Results Data–Results Block 2 (cont'd)

Word #		Bit #	Definition	Usage	Notes
PI Backplane	RS–232 and Remote I/O				
120–121	24–25	0–15	Gage #12 Value		Same as gage #9
122–123	26–27	0–15	Gage #13 Value		Same as gage #9
124–125	28–29	0–15	Gage #14 Value		Same as gage #9
126–127	30–31	0–15	Gage #15 Value		Same as gage #9
128–129	32–33	0–15	Gage #16 Value		Same as gage #9
130–131	34–35	0–15	Window #17 Value		Luminance–16.16 Object–32 bit integer Pixel–32 bit integer Color Match–32 bit integer Color ID–32 bit integer Gradient–32 bit integer
132–133	36–37	0–15	Window #18 Value		Same as window #17
134–135	38–39	0–15	Window #19 Value		Same as window #17
136–137	40–41	0–15	Window #20 Value		Same as window #17
138–139	42–43	0–15	Window #21 Value		Same as window #17
140–141	44–45	0–15	Window #22 Value		Same as window #17
142–143	46–47	0–15	Window #23 Value		Same as window #17
144–145	48–49	0–15	Window #24 Value		Same as window #17
146–147	50–51	0–15	Gage #17 Value		Angular & Linear Measure 16.16 Edge–32 bit integer Object–32 bit integer Pixel–32 bit integer Color Match–32 bit integer Color ID–32 bit integer X Position & Y Position–16.16
148–149	52–53	0–15	Gage #18 Value		Same as gage #17
150–151	54–55	0–15	Gage #19 Value		Same as gage #17
152–153	56–57	0–15	Gage #20 Value		Same as gage #17
154–155	58–59	0–15	Gage #21 Value		Same as gage #17
156–157	60–61	0–15	Gage #22 Value		Same as gage #17
158	62	0–15	Task Trigger — Most Significant Word		Same as gage #17
159	63	0–15	Task Trigger — Least Significant Word		32 Bit Integer

Results Block #3

Table C.3 shows the function of each word in results block #3.

Table C.3
Numerical Results Data– Results Block 3

Word #		Bit #	Definition	Usage	Notes
PI Backplane	RS–232 and Remote I/O				
160	0	0–15	Block Transfer Signature		
161	1	0–15	Reserved		
162–163	2–3	0–15	Gage #23 Value		Angular & Linear Measure 16.16 Edge–32 bit integer Object–32 bit integer Pixel–32 bit integer Color Match–32 bit integer Color ID–32 bit integer X Position & Y Position–16.16
164–165	4–5	0–15	Gage #24 Value		Same as gage #23
166–167	6–7	0–15	Gage #25 Value		Same as gage #23
168–169	8–9	0–15	Gage #26 Value		Same as gage #23
170–171	10–11	0–15	Gage #27 Value		Same as gage #23
172–173	12–13	0–15	Gage #28 Value		Same as gage #23
174–175	14–15	0–15	Gage #29 Value		Same as gage #23
176–177	16–17	0–15	Gage #30 Value		Same as gage #23
178–179	18–19	0–15	Gage #31 Value		Same as gage #23
180–181	20–21	0–15	Gage #32 Value		Same as gage #23
182	22	0–15	Reference Window #2 X1 Position		Feature #1 16 Bit Integer
183	23	0–15	Reference Window #2 Y1 Position		Feature #1 16 Bit Integer
184	24	0–15	Reference Window #2 X2 Position		Feature #2 16 Bit Integer
185	25	0–15	Reference Window #2 Y2 Position		Feature #2 16 Bit Integer
186	26	0–15	Reference Window #2 X3 Position		Feature #3 16 Bit Integer
187	27	0–15	Reference Window #2 Y3 Position		Feature #3 16 Bit Integer

Results Block #3 (cont'd)

Table C.3
Numerical Results Data–Results Block 2 (cont'd)

Word #		Bit #	Definition	Usage	Notes
PI Backplane	RS–232 and Remote I/O				
188	28	0–15	Reference Window #2 X–Center		Centroid of enabled feature. 16 Bit Integer.
189	29	0–15	Reference Window #2 Y–Center		Centroid of enabled feature. 16 Bit Integer.
190	30	0–15	Reference Window #2 Theta (Integer)		16.16 value. Refer to Appendix A. Only if 2 or 3 Features are enabled.
191	31	0–15	Reference Window #2 (Fraction)		Only if 2 or 3 features are enabled.
192	32	0	Reference Window #2 X1/Y1 Pass/Fail Bit	0 = Pass 1 = Fail	Feature #1
192	32	1	Reference Window #2 X2/Y2 Pass/Fail Bit	0 = Pass 1 = Fail	Feature #2
192	32	2	Reference Window #2 X3/Y3 Pass/Fail Bit	0 = Pass 1 = Fail	Feature #3
192	32	3	Combination Pass/Fail Bit	0 = Pass 1 = Fail	0 if all enabled features pass.
192	32	4–15	Not Used		
193	33	0–15	Score Reference Window #2 Feature #1		
194	34	0–15	Score Reference Window #2 Feature #2		
195	35	0–15	Score Reference Window #2 Feature #3		
196	36	0–15	Reference Window #3 X1 Position		Feature #1 16 Bit integer
197	37	0–15	Reference Window #3 Y1 Position		Feature #1 16 Bit integer
198	38	0–15	Reference Window #3 X2 Position		Feature #2 16 Bit integer
199	39	0–15	Reference Window #3 Y2 Position		Feature #2 16 Bit integer
200	40	0–15	Reference Window #3 X3 Position		Feature #3 16 Bit integer
201	41	0–15	Reference Window #3 Y3 Position		Feature #3 16 Bit integer
202	42	0–15	Reference Window #3 X–Center		Centroid of enabled feature. 16 Bit integer
203	43	0–15	Reference Window #3 Y–Center		Centroid of enabled feature. 16 Bit integer

Results Block #3 (cont'd)

Table C.3
Numerical Results Data–Results Block 2 (cont'd)

Word #		Bit #	Definition	Usage	Notes
PI Backplane	RS–232 and Remote I/O				
204	44	0–15	Reference Window #3 Theta (Integer)		16.16 value. Refer to Appendix A. Only if 2 or 3 Features are enabled.
205	45	0–15	Reference Window #3 (Fraction)		Only if 2 or 3 Features are enabled.
206	46	0	Reference Window X1/Y1 Pass/Fail Bit	0 = Pass 1 = Fail	Feature #1 16 Bit integer
206	46	1	Reference Window X2/Y2 Pass/Fail Bit	0 = Pass 1 = Fail	Feature #2 16 Bit integer
206	46	2	Reference Window X3/Y3 Pass/Fail Bit	0 = Pass 1 = Fail	Feature #3 16 Bit integer
206	46	3	Combination Pass/Fail Bit	0 = Pass 1 = Fail	0 if all enabled tools pass.
206	46	4–15	Not Used		
207	47	0–15	Score Reference Window #3 Feature #1		
208	48	0–15	Score Reference Window #3 Feature #2		
209	49	0–15	Score Reference Window #3 Feature #3		
210	50	0–15	Brightness Probe Red Integer		16.16 value
211	51	0–15	Brightness Probe Red Fraction		16.16 value
212	52	0–15	Brightness Probe Green Integer		16.16 value
213	53	0–15	Brightness Probe Green Fraction		16.16 value
214	54	0–15	Brightness Probe Blue Integer		16.16 value
215	55	0–15	Brightness Probe Blue Fraction		16.16 value
216–217	56–57	0–15	Reserved		
218	58	0–15	Reserved		
219	59	0–15	Task Missed Triggers		16 Bit integer
220	60	0–15	Task Master Faultsr — Most Significant Word		32 Bit Integer. Increments if at least 1 tool fails.
221	61	0–15	Task Master Faultsr — Least Significant Word		
222	62	0–15	Task Triggersr — Most Significant Word		32 Bit Integer.
223	63	0–15	Task Triggersr — Least Significant Word		

Results Block #4

Table C.4 shows the runction of each word in results block #4.

Table C.4
Numerical Results Data–Results Block 4

Word #		Bit #	Definition	Usage	Notes
PI Backplane	RS–232 and Remote I/O				
224	0	0–15	Block Transfer Signature		
225	1	0–15	Reserved		
226–227	2–3	0–15	Window #25 Value		Luminance–16.16 Object–32 bit integer Pixel–32 bit integer Color Match–32 bit integer Color ID–32 bit integer Gradient–32 bit integer
228–229	4–5	0–15	Window #26 Value		Same as window #25
230–231	6–7	0–15	Window #27 Value		Same as window #25
232–233	8–9	0–15	Window #28 Value		Same as window #25
234–235	10–11	0–15	Window #29 Value		Same as window #25
236–237	12–13	0–15	Window #30 Value		Same as window #25
238–239	14–15	0–15	Window #31 Value		Same as window #25
240–241	16–17	0–15	Window #32 Value		Same as window #25
242–243	18–19	0–15	Window #33 Value		Same as window #25
244–245	20–21	0–15	Window #34 Value		Same as window #25
246–247	22–23	0–15	Window #35 Value		Same as window #25
248–249	24–25	0–15	Window #36 Value		Same as window #25
250–251	26–27	0–15	Gage #33 Value		Angular & Linear Measure 16.16 Edge–32 bit integer Object–32 bit integer Pixel–32 bit integer X Position & Y Position–16.16
252–253	28–29	0–15	Gage #34 Value		Same as gage #33
254–255	30–31	0–15	Gage #35 Value		Same as gage #33
256–257	32–33	0–15	Gage #36 Value		Same as gage #33
258–259	34–35	0–15	Gage #37 Value		Same as gage #33
260–261	36–37	0–15	Gage #38 Value		Same as gage #33
262–263	38–39	0–15	Gage #39 Value		Same as gage #33
264–265	40–41	0–15	Gage #40 Value		Same as gage #33

Results Block #4 (cont'd)

Table C.4
Numerical Results Data–Results Block 4 (cont'd)

Word #		Bit #	Definition	Usage	Notes
PI Backplane	RS–232 and Remote I/O				
266–267	42–43	0–15	Gage #41 Value		Same as gage #33
268–269	44–45	0–15	Gage #42 Value		Same as gage #33
270–271	46–47	0–15	Gage #43 Value		Same as gage #33
272–273	48–49	0–15	Gage #44 Value		Same as gage #33
274–275	50–51	0–15	Window #37 Value		Luminance–16.16 Object–32 bit integer Pixel–32 bit integer Color Match–32 bit integer Color ID–32 bit integer Gradient–32 bit integer
276–277	52–53	0–15	Window #38 Value		Same as window #37
278–279	54–55	0–15	Window #39 Value		Same as window #37
280–281	56–57	0–15	Window #40 Value		Same as window #37
282–283	58–59	0–15	Window #41 Value		Same as window #37
284–285	60–61	0–15	Window #42 Value		Same as window #37
286	62	0–15	Task Trigger — Most Significant Word		32 Bit Integer.
287	63	0–15	Task Trigger — Least Significant Word		

Results Block #5

Table C.5 shows the function of each word in results block #5.

Table C.5
Numerical Results Data–Results Block 5

Word #		Bit #	Definition	Usage	Notes
PI Backplane	RS–232 and Remote I/O				
288	0	0–15	Block Transfer Signature		
289	1	0–15	Reserved		
290–291	2–3	0–15	Gage #45 Value		Angular & Linear Measure 16.16 Edge–32 bit integer Object–32 bit integer Pixel–32 bit integer X Position & Y Position–16.16
292–293	4–5	0–15	Gage #46 Value		Same as gage #45
294–295	6–7	0–15	Gage #47 Value		Same as gage #45
296–297	8–9	0–15	Gage #48 Value		Same as gage #45
298–299	10–11	0–15	Gage #49 Value		Same as gage #45
300–301	12–13	0–15	Gage #50 Value		Same as gage #45
302–303	14–15	0–15	Gage #51 Value		Same as gage #45
304–305	16–17	0–15	Gage #52 Value		Same as gage #45
306–307	18–19	0–15	Gage #53 Value		Same as gage #45
308–309	20–21	0–15	Gage #54 Value		Same as gage #45
310–311	22–23	0–15	Gage #55 Value		Same as gage #45
312–313	24–25	0–15	Gage #56 Value		Same as gage #45
314–315	26–27	0–15	Window #43 Value		Luminance 16.16 Object–32 bit integer Pixel–32 bit integer Color Match–32 bit integer Color ID–32 bit integer Gradient–32 bit integer
316–317	28–29	0–15	Window #44 Value		Same as window #43
318–319	30–31	0–15	Window #45 Value		Same as window #43
320–321	32–33	0–15	Window #46 Value		Same as window #43
322–323	34–35	0–15	Window #47 Value		Same as window #43
324–325	36–37	0–15	Window #48 Value		Same as window #43

Results Block #5 (cont'd)

Table C.5
Numerical Results Data–Results Block 5 (cont'd)

Word #		Bit #	Definition	Usage	Notes
PI Backplane	RS–232 and Remote I/O				
326–327	38–39	0–15	Gage #57 Value		Angular & Linear Measure 16.16 Edge–32 bit integer Object–32 bit integer Pixel–32 bit integer X Position & Y Position–16.16
328–329	40–41	0–15	Gage #58 Value		Same as gage #57
330–331	42–43	0–15	Gage #59 Value		Same as gage #57
332–333	44–45	0–15	Gage #60 Value		Same as gage #57
334–335	46–47	0–15	Gage #61 Value		Same as gage #57
336–337	48–49	0–15	Gage #62 Value		Same as gage #57
338–339	50–51	0–15	Gage #63 Value		Same as gage #57
340–341	52–53	0–15	Gage #64 Value		Same as gage #57
342–343	54–58	0–15	Reserved		
347	59	0–15	Task Missed Triggers		16 Bit integer
348	60	0–15	Task Master Faults — Most Significant Word		32 Bit Integer. Increments if at least 1 tool fails.
349	61	0–15	Task Master Faultss — Least Significant Word		
350	62	0–15	Task Triggerss — Most Significant Word		32 Bit Integer.
351	63	0–15	Task Triggerss — Least Significant Word		

Results Block #6

The sixth results block can be configured via either the RS–232 or Remote I/O. There are up to three configurations for this block. There is a unique configuration for each RS–232 port. The Remote I/O channel configures the block used for Remote I/O and backplane. See Table C.6.

Table C.6
Results Block #6

Word #		Bit #	Function	Notes
PI Backplane	RS–232 and Remote I/O			
352	0	0–15	Block Transfer Signature	
353	1	10–15	Status	The contents of this word will be 0 if all of the requested data fits in the 64 word results block. Otherwise, the contents will be non–0.
354–413	2–61	0–15	User Defined Inspection Results Data	
414–415	62–63	0–15	Task Triggers (32 Bits)	

Each type of result requires a specific number of words. Use the following as a guideline when setting up a programmable block transfer:

	NUMBER OF WORDS	FORMAT
Module Status	1 Word	
X/Y Reference Line	2 Words	16 Bit Integer
Window	2 Words	
	(Object & Pixel)	32 Bit Integer
	(Luminance)	16 Point 16
	(Color Match)	32 Bit Integer
	(Color ID)	32 Bit Integer
	(Gradient)	32 Bit Integer
Brightness Probe	6 Words	16 Point 16
Reference Window	14 Words	16 Bit Integer and 16 Point 16
Gage	2 Words	
	(Linear Measure)	16 Point 16
	(Edge, Pixel, Object)	32 Bit Integer
	X, Y Position	16 Point 16

Results Block #6 (cont'd)

If you request more results than will fit into 62 words, the Color CVIM will truncate the data and set an error bit in word 1.

Note: Words 2 through 9 select tools in Remote I/O only (1771 node).

Note: The first byte in Results RS–232 DF1 is the command of 0D (hex) preceded by the tool selection. The first three bytes in Statistics RS–232 DF1 are the command of 1600n (hex) where n is the statistic sample rate preceded by the tool selection. Word 0, block signature and word 1 are not used.

Use Table C.7 to set the contents of the programmable results block and statistics block.

Table C.7
Programmable Results/Statistics Block Configuration

1771 Node Adapter Results or Statistics		Function	RS–232 DF1 Results or Statistics	
Word #	Bit #		Byte #	Bit #
2	0	Module Status (NA for statistics)	0	0
2	1	Light Probe	0	1
2	2	Reference Line 1 (NA for statistics)	0	2
2	3	Reference Line 2 (NA for statistics)	0	3
2	4	Reference Line 3 (NA for statistics)	0	4
2	5	Reference Window 1	0	5
2	6	Reference Window 2	0	6
2	7	Reference Window 3	0	7
2	8	Window 1	1	0
2	9	Window 2	1	1
2	10	Window 3	1	2
2	11	Window 4	1	3
2	12	Window 5	1	4
2	13	Window 6	1	5
2	14	Window 7	1	6
2	15	Window 8	1	7
3	0	Window 9	2	0
3	1	Window 10	2	1
3	2	Window 11	2	2
3	3	Window 12	2	3
3	4	Window 13	2	4
3	5	Window 14	2	5
3	6	Window 15	2	6
3	7	Window 16	2	7

Results Block #6 (cont'd)

Table C.7
Programmable Results/Statistics Block Configuration (cont'd)

1771 Node Adapter Results or Statistics		Function	RS–232 DF1 Results or Statistics	
Word #	**Bit #**		**Byte #**	**Bit #**
3	8	Window 17	3	0
3	9	Window 18	3	1
3	10	Window 19	3	2
3	11	Window 20	3	3
3	12	Window 21	3	4
3	13	Window 22	3	5
3	14	Window 23	3	6
3	15	Window 24	3	7
4	0	Window 25	4	0
4	1	Window 26	4	1
4	2	Window 27	4	2
4	3	Window 28	4	3
4	4	Window 29	4	4
4	5	Window 30	4	5
4	6	Window 31	4	6
4	7	Window 32	4	7
4	8	Window 33	5	0
4	9	Window 34	5	1
4	10	Window **35**	5	2
4	11	Window 36	5	3
4	12	Window 37	5	4
4	13	Window 38	5	5
4	14	Window 39	5	6
4	15	Window 40	5	7
5	0	Window 41	6	0
5	1	Window 42	6	1
5	2	Window 43	6	2
5	3	Window 44	6	3
5	4	Window 45	6	4
5	5	Window 46	6	5
5	6	Window 47	6	6
5	7	Window 48	6	7
5	8	Gage 1	7	0

Results Block #6 (cont'd)

Table C.7
Programmable Results/Statistics Block Configuration (cont'd)

1771 Node Adapter Results or Statistics		Function	RS–232 Results or Statistics	
Word #	**Bit #**		**Byte #**	**Bit #**
5	9	Gage 2	7	1
5	10	Gage 3	7	2
5	11	Gage 4	7	3
5	12	Gage 5	7	4
5	13	Gage 6	7	5
5	14	Gage 7	7	6
5	15	Gage 8	7	7
6	0	Gage 9	8	0
6	1	Gage 10	8	1
6	2	Gage 11	8	2
6	3	Gage 12	8	3
6	4	Gage 13	8	4
6	5	Gage 14	8	5
6	6	Gage 15	8	6
6	7	Gage 16	8	7
6	8	Gage 17	9	0
6	9	Gage 18	9	1
6	10	Gage 19	9	2
6	11	Gage 20	9	3
6	12	Gage 21	9	4
6	13	Gage 22	9	5
6	14	Gage 23	9	6
6	15	Gage 24	9	7
7	0	Gage 25	10	0
7	1	Gage 26	10	1
7	2	Gage 27	10	2
7	3	Gage 28	10	3
7	4	Gage 29	10	4
7	5	Gage 30	10	5
7	6	Gage 31	10	6
7	7	Gage 32	10	7
7	8	Gage 33	11	0
7	9	Gage 34	11	1
7	10	Gage 35	11	2
7	11	Gage 36	11	3
7	12	Gage 37	11	4

Word #	Bit #	Function	Byte #	Bit #
7	13	Gage 38	11	5
7	14	Gage 39	11	6
7	15	Gage 40	11	7
8	0	Gage 41	12	0
8	1	Gage 42	12	1
8	2	Gage 43	12	2
8	3	Gage 44	12	3
8	4	Gage 45	12	4
8	5	Gage 46	12	5
8	6	Gage 47	12	6
8	7	Gage 48	12	7
8	8	Gage 49	13	0
8	9	Gage 50	13	1
8	10	Gage 51	13	2
8	11	Gage 52	13	3
8	12	Gage 53	13	4
8	13	Gage 54	13	5
8	14	Gage 55	13	6
8	15	Gage 56	13	7
9	0	Gage 57	14	0
9	1	Gage 58	14	1
9	2	Gage 59	14	2
9	3	Gage 60	14	3
9	4	Gage 61	14	4
9	5	Gage 62	14	5
9	6	Gage 63	14	6
9	7	Gage 64	14	7
9	8	Not Used	15	0
9	9	Not Used	15	1
9	10	Not Used	15	2
9	11	Not Used	15	3
9	12	Not Used	15	4
9	13	Not Used	15	5
9	14	Not Used	15	6
9	15	Not Used	15	7

Statistics Block

The statistics block can only be formatted through the Remote I/O port, but can be read through either the BAckplane or Remote I/O port. The statistics block has user defined contents, and may contain up to 64 words of data. See Table C.8.

Table C.8
Statistics Block

Word#	Bit#	Function	Notes
0	0–15	Block Transfer Signature	
1–63	0–15	User Defined Statistics Data	

Each type of statistic requires a specific number of words. Use the following as a guideline when setting up a programmable block transfer:

Block Signature	= 1 word
Number of Samples	= 1 word (2 Byte Integer)
Minimum and Maximum Values	= 4 bytes
Window (Object & Pixel)	32 Bit Integer
Window(Luminance)	16 Point 16
Window (Color Match)	32 Bit Integer
Window (Color ID)	32 Bit Integer
Window (Gradient)	32 Bit Integer
Brightness Probe	16 Point 16
Reference Window	16 Bit Integer & 16 Point 16
Gage (Linear Measure)	16 Point 16
Gage (Edge, Pixel, Object)	32 Bit Integer
Average	= 4 Bytes
(16 Point 16 for fixed point values)	
(24 Point 8 for integers)	
Standard Deviation	= 4 bytes (16 Point 16)

Each tool statistic consists of 18 bytes with the exception of reference windows and brightness probe which contain 18 bytes for each feature or a total of 54 bytes. The statistics block is transmitted as two hexadecimal characters for each byte. The total number of bytes including the block signature should not exceed 128 bytes.

If you request more results than will fit into 63 words, the Color CVIM will truncate the data and set an error bit in word 1.

Appendix D

Configuration Data

Configuration Block Overview

There are 213 configuration blocks. The following is an overview of the blocks.

Block Number 1	System Environment (42 words).
Block Number 2	Camera Definition Part 1 (50 words).
Block Number 3	Camera Definition Part 2 (36 words).
Block Numbers 4 through 35	Colors Definition 1 through 32 (14 words).
Block Number 36	Gray Scale Definition (8 words).
Block Numbers 37 through 39	Reference Lines 1 through 3 Part 1 (30 words).
Block Numbers 40, 42, and 44	Reference windows 1 through 3 Part 1 (9 words).
Block Numbers 41, 43, and 45	Reference windows 1 through 3 Part 2 (59 words).
Block Numbers 46 to 109	Gages 1 through 64 (28 words).
Block Numbers 110, 112 & 204	Windows 1 through 48 Part 1 (24 words).
Block Numbers 111, 113 & 205	Windows 1 through 48 Part 2 (56 words).
Block Numbers 206 through 213	Polygon Blocks 1 through 8 (35 words).

Note: When reading the configuration blocks, the PLC should set the block length to 0. This will allow the Color CVIM module to set the block size based upon the block number received. The Color CVIM module will then only send the amount of data required for each block type. This helps reduce the overall transfer time when writing. If the PLC sets the block length to 0 (64 words) when writing to a Color CVIM module, the Color CVIM module will ignore excess data at the end of each block. Alternatively, the PLC may set the exact length to reduce the transfer time.

Configuration Block #1

Table D.1 shows the function of each word in the system environment configuration block.

Table D.1
Configuration Block #1– System Environment

Remote I/O & RS–232 Word #*	Bit #	Definition	Usage	Notes
0	0–15	Block Transfer Signature		
1–3	0–15	Reserved		
4	0–7	Configuration ID (Char. 2)		
4	8–15	Configuration ID (Char.1)		
5	0–7	Configuration ID (Char. 4)		
5	8–15	Configuration ID (Char. 3)		
6	0–7	Configuration ID (Char. 6)		
6	8–15	Configuration ID (Char. 5)		
7	0–7	Configuration ID (Char. 8)		
7	8–15	Configuration ID (Char. 7)		
8	0–7	Configuration ID (Char. 10)		
8	8–15	Configuration ID (Char. 9)		
9	0–7	Configuration ID (Char. 12)		
9	8–15	Configuration ID (Char.11)		
10	0–7	Configuration ID (Char. 14)		
10	8–15	Configuration ID (Char.13)		
11	0–7	Configuration ID (Char. 16)		
11	8–15	Configuration ID (Char.15)		
12	0–15	Reserved		
13	0	Reserved		
13	1	Remote I/O Enable	0 = Disabled 1 = Enabled	
13	2 & 4	Protocol–Port A	0 = ASCII 1 = DF1 2 = Not Specified 3 = Not Specified	Port A, see bits 5 & 6 for port B.
13	3	Host Standby Mode	0 = Standby Enabled 1 = Standby Disabled	

* Refer to Chapter 6 for Pyramid Integrator long word descriptions.

Configuration Block #1 (cont'd)

Table D.1
Configuration Block #1– System Environment (cont'd)

Remote I/O & RS–232 Word #*	Bit #	Definition	Usage	Notes
13	5&6	Protocol–Port B	0 = ASCII 1 = DF1 2 = Not Specified 3 = Not Specified	Port B, see bits 2 & 4 for port A.
13	7–15	Reserved		
14–15	0–15	Reserved		
16	0–7	Display Status	0 = Disabled 1 = Enabled	All tools appear simultaneously while setting a given tool.
16	8–15	Tool Display	0 = No Tools 1 = Reference Lines 2 = Reference Windows 3 = Gages 4 = Windows 5 = Formulae 6 = All Tools	
17	0–7	Reserved		
17	8	Learn Mode	0 = Disabled 1 = Enabled	
17	9	Outputs Enable	0 = Disabled 1 = Enabled	
17	10	Freeze Enable	0 = Disabled 1 = Enabled	
17	11	Halt Enable	0 = Disabled 1 = Enabled	
17	12–15	Reserved		
18–31	0–15	Reserved		
32	0–15	Pulse Width		In milliseconds
33–35	0–15	Reserved		
36	0–15	Number of Samples for Statistics		
37–41	0–15	Reserved		

* Refer to Chapter 6 for Pyramid Integrator long word descriptions.

Configuration Blocks 2 & 3

Tables D.2 and D.3 show the function of each word in the camera definition configuration blocks.

Table D.2
Configuration Block #2 – Camera Definition

Remote I/O & RS–232 Word #*	Bit #	Definition	Usage	Notes
0	0–15	Block Transfer Signature		
1	0–9 10–15	Reserved High Reference of Red Converter	0 = Minimum Value . . . 63 = Maximum Value	
2	0–9 10–15	Reserved High Reference of Green Converter	0 = Minimum Value . . . 63 = Maximum Value	
3	0–9 10–15	Reserved High Reference of Blue Converter	0 = Minimum Value . . . 63 = Maximum Value	
4	0–7	Light Probe Status	0 = Disabled 1 = Same Field 2 = Next Field	
4	8–15	Reserved		
5	0–15	Light Probe X Location	16 = Minimum Value . . . 504 = Maximum Value	
6	0–15	Light Probe Y Location	8 = Minimum Value . . . 232 = Maximum Value	
7–16	0–15	Reserved		
17	0–15	Fail Range High for Red Channel (Integer)		Words 17 and 18 represent a 16(bit).16(bit) fixed point decimal value.
18	0–15	Fail Range High for Red Channel (Fraction)		
19	0–15	Fail Range Low for Red Channel (Integer)		Words 19 and 20 represent a 16 (bit). 16 (bit) fixed point decimal value.
20	0–15	Fail Range Low for Red Channel (Fraction)		
21	0–15	Warning Range High for Red Channel (Integer)		Words 21 and 22 represent a 16 (bit).16 (bit) fixed point decimal value.
22	0–15	Warning Range High for Red Channel (Fraction)		
23	0–15	Warning Range Low for Red Channel (Integer)		Words 23 and 24 represent a 16 (bit). 16 (bit) fixed point decimal value.

Configuration Blocks 2 & 3 (cont'd)

Table D.2
Configuration Block #2 – Camera Definition (cont'd)

Remote I/O & RS–232 Word #*	Bit #	Definition	Usage	Notes
24	0–15	Warning Range Low for Red Channel (Fraction)		
25–28	0–15	Reserved		
29	0–15	Fail Range High for Green Channel (Integer)		Words 29 and 30 represent a 16(bit).16(bit) fixed point decimal value.
30	0–15	Fail Range High for Green Channel (Fraction)		
31	0–15	Fail Range Low for Green Channel (Integer)		Words 31 and 32 represent a 16 (bit). 16 (bit) fixed point decimal value.
32	0–15	Fail Range Low for Green Channel (Fraction)		
33	0–15	Warning Range High for Green Channel (Integer)		Words 33 and 34 represent a 16 (bit).16 (bit) fixed point decimal value.
34	0–15	Warning Range High for Green Channel (Fraction)		
35	0–15	Warning Range Low for Green Channel (Integer)		Words 35 and 36 represent a 16 (bit). 16 (bit) fixed point decimal value.
36	0–15	Warning Range Low for Green Channel (Fraction)		
37–40	0–15	Reserved		
41	0–15	Fail Range High for Blue Channel (Integer)		Words 41 and 42 represent a 16(bit).16(bit) fixed point decimal value.
42	0–15	Fail Range High for Blue Channel (Fraction)		
43	0–15	Fail Range Low for Blue Channel (Integer)		Words 43 and 44 represent a 16 (bit). 16 (bit) fixed point decimal value.
44	0–15	Fail Range Low for Blue Channel (Fraction)		
45	0–15	Warning Range High for Blue Channel (Integer)		Words 45 and 46 represent a 16 (bit).16 (bit) fixed point decimal value.
46	0–15	Warning Range High for Blue Channel (Fraction)		
47	0–15	Warning Range Low for Blue Channel (Integer)		Words 47 and 49 represent a 16 (bit). 16 (bit) fixed point decimal value.
48	0–15	Warning Range Low for Blue Channel (Fraction)		
49	0–15	Reserved		

* Refer to Chapter 6 for Pyramid Integrator long word descriptions.

Configuration Blocks 2 & 3 (cont'd)

Table D.3
Configuration Block #3 – Camera Definition

Remote I/O & RS–232 Word #*	Bit #	Definition	Usage	Notes
0	0–15	Block Transfer Signature		
1–35	0–15	Reserved		

* Refer to Chapter 6 for Pyramid Integrator long word descriptions.

Configuration Blocks 4–35

Table D.4 shows the function of each word in the color image definition for colors 1 to 32.

Table D.4
Configuration Blocks #4–35 – Color Definition

Remote I/O & RS–232 Word #*	Bit #	Definition	Usage	Notes
0	0–15	Block Transfer Signature		
1–12		Description of Image #1		
1	0	Status	0 = Disable 1 = Enable	
	1–15	Reserved		
2	0	Segmentation Type	0 = HSI 1 = RGB	
	1–15	Reserved		
3–6		Teaching Window of Color #1		
3	0–15	Window X Location		
4	0–15	Window Y Location		
5	0–15	Window Width		
6	0–15	Window Height		
7	0–15	High Threshold of H/R	0 to 255	
8	0–15	Low Threshold of H/R	0 to 255	
9	0–15	High Threshold of S/G	0 to 255	
10	0–15	Low Threshold of S/G	0 to 255	
11	0–15	High Threshold of I/B	0 to 255	
12	0–15	Low Threshold of I/B	0 to 255	
13	0–15	Reserved		

Note: The words 1 through 12 contain the definition of the color:

#1	in block	#4
#2	in block	#5
#3	in block	#6
. . .		
#32	in block	#35

Configuration Block 36

Table D.5 shows the function of each word in the Gray Scale Images definition.

Table D.5
Configuration Block 36 – Gray Scale Definition

Remote I/O & RS–232 Word #*	Bit #	Definition	Usage	Notes
0	0–15	Block Transfer Signature		
1	0–15	Hue Status (#33)	0 = Disable 1 = Enable	
2	0–15	Saturation Status (#34)	0 = Disable 1 = Enable	
3	0–15	Intensity Status (#35)	0 = Disable 1 = Enable	
4	0–15	Red Status (#36)	0 = Disable 1 = Enable	
5	0–15	Green Status (#37)	0 = Disable 1 = Enable	
6	0–15	Blue Status (#38)	0 = Disable 1 = Enable	
7	0–15	Reserved		

Configuration Blocks 37–39

Table D.6 shows the function of each word in the reference line 1–3 configuration blocks.

Table D.6
Configuration Blocks #37–39 – Reference Lines 1–3

Remote I/O & RS–232 Word #	Bit #	Definition	Usage	Notes
0	0–15	Block Transfer Signature		
1	0–7	Reserved		
1	8	Enable	0 = Disabled 1 = Enabled	
1	9–15	Reserved		
2–7	0– 15	Reserved		
8	0–7	X–Line High Threshold/ Gradient Threshold	0–197	
8	8–15	Image number	0–37	
9	0–7	Reserved		
9	8–15	X–Line Low Threshold/Scale Factor		
10	0–15	X–Line Head X Position		From upper left corner.
11	0–15	X–Line Head Y Position		From upper left corner.
12	0–15	X–Line Tail X Position		From upper left corner.
13	0–15	X–Line Tail Y Position		From upper left corner.
14	0–15	Reserved		
15	0–7	Y–Line High Threshold/ Gradient Threshold	0–197	
15	8–15	Image number	0–37	
16	0–7	Reserved		
16	8–15	Y–Line Low Threshold/ Scale Factor		
17	0–15	Y–Line Head X Position		From upper left corner.
18	0–15	Y–Line Head Y Position		From upper left corner.
19	0–15	Y–Line Tail X Position		From upper left corner.
20	0–15	Y–Line Tail Y Position		From upper left corner.
21	0–15	Reserved		

Configuration Blocks 37–39 (cont'd)

Table D.6
Configuration Blocks #37–39 – Reference Lines 1–3 (cont'd)

Remote I/O & RS–232 Word #	Bit #	Definition	Usage	Notes
22	0–7	X/Y–Line High Threshold/ Gradient Threshold	0–197	
22	8–15	Image Number	0–37	
23	0–7	Reserved		
23	8–15	X/Y–Line Low Threshold/ Scale Factor		
24	0–15	X/Y–Line Head X Position		From upper left corner.
25	0–15	X/Y–Line Head Y Position		From upper left corner.
26	0–15	X/Y–Line Tail X Position		From upper left corner.
27	0–15	X/Y–Line Tail Y Position		From upper left corner.
28–29	0–15	Reserved		

Configuration Blocks 40–45

Table D.7 shows the function of each word in the reference window 1–3 configuration blocks.

Table D.7
Configuration Blocks #40, 42, and 44 – Reference Windows 1–3

Remote I/O & RS–232 Word #*	Bit #	Definition	Usage	Notes
0	0–15	Block Transfer Signature		
1	0–7	Reserved		
1	8	Enable	0 = Disabled 1 = Enabled	
1	9–15	Reserved		
2–8	0– 15	Reserved		

* Refer to Chapter 6 for Pyramid Integrator long word descriptions.

Note: There are two Configuration Blocks per Reference Window:

Ref. Window 1 (Part 1)	in block	#40
Ref. Window 1 (Part 2)	in block	#41
Ref. Window 2 (Part 1)	in block	#42
Ref. Window 2 (Part 2)	in block	#43
Ref. Window 3 (Part 1)	in block	#44
Ref. Window 3 (Part 2)	in block	#45

Configuration Blocks 40–45 (cont’d)

Table D.8
Configuration Blocks #41, 43, & 45 – Reference Windows 1–3

Remote I/O & RS–232 Word #	Bit #	Definition	Usage	Notes
0	0–15	Block Transfer Signature		
1	0–15	Feature 1 Search Window X Location		Relative to the upper left corner.
2	0–15	Feature 1 Search Window Y Location		Relative to the upper left corner.
3	0–15	Feature 1 Search Window Width		
4	0–15	Feature 1 Search Window Height		
5–8	0–15	Reserved		
9	0–7	Reserved		
9	8–15	Feature 1 Score	0 = Minimum Value 100 = Maximum Value	
10–19	0–7	Reserved		
20	0–15	Feature 2 Search Window X Location		Relative to the upper left corner.
21	0–15	Feature 2 Search Window Y Location		Relative to the upper left corner.
22	0–15	Feature 2 Search Window Width		
23	0–15	Feature 2 Search Window Height		
24–27	0–15	Reserved		
28	0–7	Reserved		
28	8–15	Feature 2 Score	0 = Minimum Value 100 = Maximum Value	
29–38	0–15	Reserved		
39	0–15	Feature 3 Search Window X Location		Relative to the upper left corner.
40	0–15	Feature 3 Search Window Y Location		Relative to the upper left corner.
41	0–15	Feature 3 Search Window Width		

Configuration Blocks 40–45 (cont'd)

Table D.8
Configuration Blocks #41, 43, & 45 – Reference Windows 1–3 (cont'd)

Remote I/O & RS–232 Word #	Bit #	Definition	Usage	Notes
42	0–15	Feature 3 Search Window Height		
43–46	0–15	Reserved		
47	0–7	Reserved		
47	8–15	Feature 3 Score	0 = Minimum Value 100 = Maximum Value	
48–58	0–15	Reserved		

Configuration Blocks 46–109

Table D.9 shows the function of each word in the gage 1–64 configuration blocks.

Table D.9
Configuration Blocks #46–109 – Gages 1–64

Remote I/O & RS–232 Word #*	Bit #	Definition	Usage	Notes
0	0–15	Block Transfer Signature		
1	0	Enable	0 = Disabled 1 = Enabled.	
1	1–15	Reserved		
2–3	0–15	Reserved		
4	0–7	X–Line High Threshold/ Scale Factor	0–197	
4	8–15	Image Number	0–37	
5	0–7	Reserved		
5	8–15	X–Line Low Threshold/ Scale Factor		
6	0–15	Gage Head X Position		
7	0–15	Gage Head Y Position		
8	0–15	Gage Tail X Position		
9	0–15	Gage Tail Y Position		
10	0–15	Gage X Center Position (Circular Gage)		
11	0–15	Gage Y Center Position (Circular Gage)		
12	0–15	Reserved		
13–16	0–15	Reserved		

* Refer to Chapter 6 for Pyramid Integrator long word descriptions.

Configuration Blocks 46–109 (cont’d)

Table D.9
Configuration Blocks #46–109 – Gages 1–64 (cont’d)

Remote I/O & RS–232 Word #*	Bit #	Definition	Usage	Notes
17	0–15	Fail Range High (Integer)		Words 17 and 18 represent a 16 (bit).16 (bit) fixed point decimal value or 32 bit integer Refer to Appendix A.
18	0–15	Fail Range High (Fraction)		
19	0–15	Fail Range Low (Integer)		Words 19 and 20 represent a 16 (bit).16 (bit) fixed point decimal value or 32 bit integer. Refer to Appendix A.
20	0–15	Fail Range Low (Fraction)		
21	0–15	Warning Range High (Integer)		Words 21 and 22 represent a 16 (bit).16 (bit) fixed point decimal value or 32 bit integer. Refer to Appendix A.
22	0–15	Warning Range High (Fraction)		
23	0–15	Warning Range Low (Integer)		Words 23 and 24 represent a 16 (bit).16 (bit) fixed point decimal value or 32 bit integer. Refer to Appendix A.
24	0–15	Warning Range Low (Fraction)		
25–27	0–15	Reserved		

* Refer to Chapter 6 for Pyramid Integrator long word descriptions.

Note: Block #46 describes gage #1
Block #47 describes gage #2
. . .
Block #109 describes gage #64

Configuration Blocks 110–205

Tables D.10 and D.11 show the function of each word in the window 1– 48 configuration blocks.

Table D.10
Configuration Blocks #110, 112 . . . & 204 – Windows 1–48

Remote I/O & RS–232 Word #*	Bit #	Definition	Usage	Notes
0	0–15	Block Transfer Signature		
1	0	Enable	0 = Disabled 1 = Enabled	
1	1–15	Reserved		
2	0–7	Threshold for the Gray Scale Images		
2	8–15	Image Number	0–37	
3	0–15	Window X Location (Bounding Box)		
4	0–15	Window Y Location (Bounding Box)		
5	0–15	Window Width (Bounding Box)		
6	0–15	Window Height (Bounding Box)		
7	0–15	Mask X Location (Bounding Box)		
8	0–15	Mask Y Location (Bounding Box)		
9	0–15	Mask Width (Bounding Box)		
10	0–15	Mask Height (Bounding Box)		
11–14	0–15	Reserved		

Configuration Blocks 110–205 (cont'd)

Table D.11
Configuration Blocks #110, 112 . . . & 204 – Windows 1–48

Remote I/O & RS–232 Word #	Bit #	Definition	Usage	Notes
15	0–15	Fail Range High (Integer)		Words 15 and 16 represent a 16 (bit).16 (bit) fixed point decimal value or 32 bit integer.
16	0–15	Fail Range High (Fraction)		
17	0–15	Fail Range Low (Integer)		Words 17 and 18 represent a 16 (bit).16 (bit) fixed point decimal value or 32 bit integer.
18	0–15	Fail Range Low (Fraction)		
19	0–15	Warning Range High (Integer)		Words 19 and 20 represent a 16 (bit). 16 (bit) fixed point decimal value or 32 bit integer.
20	0–15	Warning Range High (Fraction)		
21	0–15	Warning Range Low (Integer)		Words 21 and 22 represent a 16 (bit).16 (bit) fixed point decimal value or 32 bit integer.
22	0–15	Warning Range Low (Fraction)		
23	0–15	Reserved		

Note: Each window needs two blocks for the description:

Block #110 describes window #1 (Part 1)
Block #111 describes window #1 (Part 2)
Block #112 describes window #2 (Part 1)
. . .
Block #204 describes window #48 (Part 1)
Block $205 describes window #48 (Part 2)

Table D.12
Configuration Block #111, 113 . . . & 205 – Windows 1–48

Remote I/O & RS–232 Word #*	Bit #	Definition	Usage	Notes
0	0–15	Block Transfer Signature		
1–55	0–15	Reserved		

Configuration Blocks 206–213

Table D.12 shows the function of each word in the polygon configuration blocks.

Table D.13
Polygon Configuration Words

Remote I/O & RS–232 Word #	Bit #	Definition	Usage	Notes
0	0–15	Block Transfer Signature		
1–34	0–15	Reserved		

Appendix E

ASCII Conversion Table

ASCII or Control Char.	Decimal Value	Hex Value	ASCII or Control Char.	Decimal Value	Hex Value	ASCII or Control Char.	Decimal Value	Hex Value	ASCII or Control Char.	Decimal Value	Hex Value
NUL	0	0	[Space]	32	20	@	64	40	’	96	60
SOH	1	1	!	33	21	A	65	41	a	97	61
STX	2	2	”	34	22	B	66	42	b	98	62
ETX	3	3	#	35	23	C	67	43	c	99	63
EOT	4	4	$	36	24	D	68	44	d	100	64
ENQ	5	5	%	37	25	E	69	45	e	101	65
ACK	6	6	&	38	26	F	70	46	f	102	66
BEL	7	7	’	39	27	G	71	47	g	103	67
BS	8	8	(	40	28	H	72	48	h	104	68
HT	9	9	)	41	29	I	73	49	i	105	69
LF	10	A	*	42	2A	J	74	4A	j	106	6A
VT	11	B	+	43	2B	K	75	4B	k	107	6B
FF	12	C	,	44	2C	L	76	4C	l	108	6C
CR	13	D	–	45	2D	M	77	4D	m	109	6D
SO	14	E	.	46	2E	N	78	4E	n	110	6E
SI	15	F	/	47	2F	O	79	4F	o	111	6F
DLE	16	10	0	48	30	P	80	50	p	112	70
DC1	17	11	1	49	31	Q	81	51	q	113	71
DC2	18	12	2	50	32	R	82	52	r	114	72
DC3	19	13	3	51	33	S	83	53	s	115	73
DC4	20	14	4	52	34	T	84	54	t	116	74
NAK	21	15	5	53	35	U	85	55	u	117	75
SYN	22	16	6	54	36	v	86	56	v	118	76
ETB	23	17	7	55	37	W	87	57	w	119	77
CAN	24	18	8	56	38	X	88	58	x	120	78
EM	25	19	9	57	39	Y	89	59	y	121	79
SUB	26	1A	:	58	3A	Z	90	5A	z	122	7A
ESC	27	1B	;	59	3B	[	91	5B	{	123	7B
FS	28	1C	<	60	3C	\	92	SC	|	124	7C
GS	29	1D	=	61	3D	]	93	5D	}	125	7D
RS	30	1E	>	62	3E	^	94	5E	~	126	7E
US	31	1F	?	63	3F	_	95	5F			

A

ACK
An abbreviated term for Positive Acknowledgment. A control code that indicates that the previous transmission block was received.

address
A character or group of characters that identifies a register, a particular part of storage, or some other data source or destination. To refer to a device or an item of data by its address.

ASCII
The character set and code described in American National Standard Code for Information Interchange, ANSI X3.4–1977. Each ASCII character is encoded with 8–bits including parity check.

B

backplane
A printed circuit card located in the back of a rack, which has sockets into which specific boards fit for interconnection.

BASIC
Acronym for Beginner's All–Purpose Symbolic Instruction Code. A problem solving, algebra–like programming language.

block
A group of words considered as a unit.

bit
An acronym for Binary Digit. The smallest unit of information in the binary numbering system. Represented by the digits 0 and 1.

byte
A unit of data that contains 8 bits

C

centroid
Midpoint of x and y axis of an object.

CFG host
The device that is allowed to read/write configurations. These configurations can be stored on the ram card or in the CFG host memory.

CVIM
Allen–Bradley trademark for Configurable Vision Input Module. Pronounced as "See VIM".

D

data link
The communication(s) lines, related controls, and interface(s) for the transmission of data between two or more devices.

F

fixed point
A number system in which the position of the decimal point is fixed in respect to one end of a string of numbers.

flag
An indicator. A single bit of a memory location, used to detect and remember the occurrence of some event.

floating point
A system of representing numerical quantities with a variable number of places in which the location of the point does not remain fixed.

G

gray scale
In monochromatic displays, variations in brightness level used to enhance the contrast among the displayed features.

H

handshaking
Two–way communication between two devices to effect a data transfer. Handshaking operations are based on a Data–Ready/Data–Received signal scheme that assures orderly data transfer.

hex
Abbreviated form of the word hexadecimal.

hexadecimal
A base 16 numbering system.

hexadecimal numbering system
A numbering system using the equivalent of the decimal number 16 as a base. Because only a single character is allowed for each absolute value, the hexadecimal numbering system uses the 10 symbols of the decimal system for values 0 through 9, and the first six letters of the alphabet to represent values 10 through 15 (a through F). The positional significance of the hexadecimal symbols is based upon the progression of powers of 16. The highest number that can be represented in the units position is 15.

I

Image
A photographic picture, e.g., as being picked up by a TV camera. Mathematically, an image can be described by a function of 2 variables f(x,y), usually defined over a rectangular region. X and y are the region coordinates, and f(x,y) represents the gray scale value of the point (x,y) in the region.

I/O
Acronym for Input/ Output.

L

left justified
A field of numbers (decimal, binary, etc) with no zeros or spaces to the left.

lightpen
A hand held photosensitive input device used to designate a location on a display screen.

N

NAK
An abbreviated term for Negative Acknowledgment. A control code that indicates the previous transmission block was not received correctly.

P

parity bit
A parity bit is added to a binary array to make the sum of all the bits always odd or always even; a fundamental check.

pixel
An element of a picture. In order for a computer to analyze a picture, the picture is broken up into a series of picture elements called pixels. Each pixel is assigned a brightness level which is the average of the area in the pixel. In computer vision systems, the pixels is the smallest area of resolution in a picture.

PLC
Allen–Bradley trademark for programmable logic controller.

Q

Q–bus
A set of electrical conductors that carry specific signals to several other circuits.

R

RS–232
Standard electrical interface.

S

standard deviation
A measure of the dispersion around a mean value.

serial port
A communications connector through which data is transmitted one bit at a time.

space character
A graphic character that is usually represented by a blank site in a series of graphics. The space character, though not a control character, has a function equivalent that of a formal effect or that causes the printer or display to move one position forward without producing the printing or display of any graphics.

string
A sequence of ASCII characters.

subroutine
A series of computer instructions which perform a specific task for other routines. It is distinguishable from from a main routine in that it requires as one of its parameters a location specifying where to return to the main program after its function has been accomplished.

SYS host
The device that controls the Color CVIM's display and the lightpen's permission to enter configuration changes. The SYS host may also trigger the Color CVIM

T

TTL
A signal processing system in which data in the form of low level electrical signals is processed through circuits either discretely or through integrated circuits comprised primarily of transistors.

W

word
A unit of data which contains two bytes (16 bits).

16 point 16, 4-18

2801-JMB interface board, 3-14

6008-SI, Sample program, 4-27

A

Accessing results data, 4-21

ACK/NAK characters, 5-35

Addressing, 2-5

ASCII
  Character set, 5-7
  Command structure, 5-7
  Command summary (table), 5-27
  Field types, 5-8
  Protocol, 5-2, 5-6
  Sample program, 5-29

ATTENTION, 3-13, 3-14, 4-16, 4-17, 4-20

B

Backplane, 2-3, 6-1
  Configuration, 6-2

Baud rate, 4-1

BCC (block check character), 5-36

Bit numbering, 2-6

Block check character, 5-36

Block transfers, 4-15

C

Cables, Communication, 2-5, 4-1, 5-3

CFG host, 2-7, 4-15, 4-20, 5-4, 5-5, 6-2

Command(s)
  ASCII structure, 5-7
  Configure statistics block, 5-23
  Deactivate force, 5-8, 5-37
  DF1 structure, 5-34
  Disable outputs, 5-9, 5-39
  Display object, 5-21, 5-51
  Download configuration, 5-25, 5-57
  Echo, 5-9, 5-38
  Enable outputs, 5-9, 5-39
  Force local I/O, 5-10, 5-39
  Load configuration, 5-11, 5-41
  Lock, 5-12, 5-42
  Read configuration, 5-14, 5-43
  Read data, 5-13, 5-42
  Read inspection results, 5-16, 5-47
  Read output status, 5-13, 5-42
  Read results, 5-16, 5-47
  Read statistics block, 5-23, 5-54
  Save (configuration), 5-19, 5-50
  Select image, 5-21, 5-51
  Set configurable results, 5-22, 5-53
  Set results block 6, 5-22, 5-53
  Set statistics block, 5-23, 5-54
  Summary of ASCII (table), 5-27
  Summary of DF1 (table), 5-60
  Trigger operation, 5-25, 5-56
  Unlock, 5-13, 5-56
  Write configuration, 5-25, 5-57

Configurable results, Set, 5-22, 5-53

Configuration
  Accessing, 4-24
  Backplane, 6-2
  Block transfers, 4-20
  Data, 2-5, 4-15, 4-20, 6-4
  Download, 5-57
  Host, 2-7
  Instructions, 4-7, 5-5, 6-4
  Load, 5-11, 5-41
  Read, 5-14, 5-43
  Save, 5-19, 5-50
  Write, 5-25, 5-57

Configuration block, Command bytes, 5-44

Configure, Statistics block, 5-23

Connections, 3-11, 3-13, 5-3
  Diagrams, 4-2
  Interface board, 3-15

Converting results data, 4-18

D

Data
  Configuration, 2-5
  Read, 5-42
  Results, 2-5, 4-7
  Transmission speed, 4-1

Data Field, 5-8

DF1 protocol, 5-2, 5-33
  Character set, 5-33
  Command structure, 5-34
  Command summary (table), 5-60
  Field types, 5-34
  Sample program, 5-62

Discrete bits information, Accessing, 4-8

Download configuration, 5-25

E

EEPROM, 4-6, 5-4

F

Field
  Data, 5-8
  Object, 5-8
  Operation, 5-8

Firmware revision, 1-1

Flexible interface module, 5-1

Flow control, 5-8

Format
  16 point 16, 4-18
  Data, 5-5
  Statistics block, 4-17, 5-55

Function(s)
  Assignment, 3-2
  Possible on backplane, 6-2
  Possible on remote I/O network, 4-6
  Possible over RS-232 interface, 5-4

H

Host
  CFG, 4-20
  Configuration, 2-7
  Designation, 2-7
  System, 2-7

I

I/O lines, Number of, 2-1

Inspection cycle, 3-7

Interface board, 3-14

Interface box, 2-3, 3-1, 3-11, 3-13, 5-3

L

Local I/O, 3-1

M

Memory
  Addressing, 2-5
  Random access, 2-1, 4-6
  Requirements for images and configurations, 5-12, 5-20

MicroVAX processor, 6-13

Module I/O port, 2-3, 3-1

Multiple hosts, 2-7

N

Numbering, Bit, 2-6

O

Object field, 5-8

Operation field, 5-8

Output, Planning assignments, 3-2

Output line planning sheet
  Blank, 3-17
  Sample, 3-3

Output signal, Timing data, using, 3-7

Output status, Read, 5-42

P

PI, Defined, 1-4

Pinouts, 3-11

Planning outputs, 3-2
  Blank form, 3-17
  Sample form, 3-3

PLC, use of Color CVIM with, 1-2, 2-3, 2-6, 4-8, 6-3
  Sample program, 6-6

Port
  Module I/O, 2-3, 3-1
  Remote I/O, 2-3
  RIO, 4-1
  RS-232, 2-3

Protocol
  ASCII, 5-6
  DF1, 5-33
  Options, 5-2, 5-5

Pyramid Integrator, 1-4, 6-1
  Backplane, 2-3

R

RAM, 2-1
  Card, 4-6, 5-4

Remote I/O
  Configuration instructions, 4-7
  Functions, 4-6
  Port, 2-3, 4-1

Results
  Blocks transfer, 4-15
  Data words, 4-7
  Read, 5-16, 5-47

Results block 6, 5-22, 5-53

Results data, 2-5
  Accessing, 4-21
  Converting, 4-18

Revision number, 1-1

RIO, 2-3, 4-1

RS-232 ports, 2-3, 3-11

S

Sample program
  6008-SI, 4-27
  Accessing configuration data, 4-24
  Accessing results data, 4-21
  Accessing/Setting discrete bit data, 4-12
  ASCII, 5-29
  Converting 16 point 16, 4-18
  DF1, 5-62
  PLC, 6-6

Statistics block
  Configure, 5-23
  Format, 4-17, 5-55
  Read, 5-23, 5-54
  Set, 5-23, 5-54

SYS host, 2-7, 4-15, 5-4, 5-5, 5-54, 6-2

T

Testing, RS-232 links, 5-6

Timing data, using, 3-7

Tool(s), Assignment, 3-2

Transmission speed, 4-1

Trigger operation, 5-25, 5-56

Trigger signal, 3-7

V

Voltage, 3-14

W

Write configuration, 5-25

X

XON/XOFF flow control, 5-8

ALLEN-BRADLEY
A ROCKWELL INTERNATIONAL COMPANY

A subsidiary of Rockwell International, one of the world's largest technology companies, Allen-Bradley meets today's automation challenges with over 85 years of practical plant floor experience.Over 11,000 employees throughout the world design, manufacture and apply a wide range of control and automation products and supporting services to help our customers continuously improve quality, productivity and time to market. These products and services not only control individual machines, but also integrate the manufacturing process while providing access to vital plant floor data that can be used to support decision–making throughout the enterprise.

With offices in major cities worldwide.

WORLD HEADQUARTERS
Allen-Bradley
1201 South Second Street
Milwaukee, WI 53204 USA
Tel: (1) 414 382–2000
Telex: 43 11 016
FAX: (1) 414 382–4444

EUROPE/MIDDLE EAST/
AFRICA HEADQUARTERS
Allen–Bradley Europe B.V.
Amsterdamseweg 15
1422 AC Uithoorn
The Netherlands
Tel: (31) 2975/43500
Telex: (844) 18042
FAX: (31) 2975/60222

ASIA/PACIFIC HEADQUARTERS
Allen–Bradley (Hong Kong) Limited
Room 1006, Block B, Sea View Estate
2-8 Watson Road
Hong Kong
Tel: (852) 887-4788
Telex: (780) 64347
FAX: (852) 510-9436

CANADA HEADQUARTERS
Allen–Bradley Canada Limited
135 Dundas Street
Cambridge, Ontario N1R 5X1
Canada
Tel: (1) 519 623–1810
FAX: (1) 519 623–8930

LATIN AMERICA
HEADQUARTERS
Allen-Bradley
1201 South Second Street
Milwaukee, WI 53204 USA
Tel: (1) 414 382–2000
Telex: 43 11 016
FAX: (1) 414 382–2400

Catalog Number 5370–ND010 Series A