

**MICROSCAN®**

# MicroHAWK ID-20, ID-30, ID-40 User Manual



*P/N 84-9007234-02 Rev A*

---

## *Copyright and Disclaimer*

Copyright ©2015  
Microscan Systems, Inc.  
Tel: +1.425.226.5700 / 800.762.1149  
Fax: +1.425.226.8250

All rights reserved. The information contained herein is proprietary and is provided solely for the purpose of allowing customers to operate and/or service Microscan manufactured equipment and is not to be released, reproduced, or used for any other purpose without written permission of Microscan.

Throughout this manual, trademarked names might be used. We state herein that we are using the names to the benefit of the trademark owner, with no intention of infringement.

## ***Disclaimer***

The information and specifications described in this manual are subject to change without notice.

## ***Latest Manual Version***

For the latest version of this manual, see the Download Center on our web site at: [www.microscan.com](http://www.microscan.com).

## ***Technical Support***

For technical support, e-mail: [helpdesk@microscan.com](mailto:helpdesk@microscan.com).

## ***Warranty***

For current warranty information, see: [www.microscan.com/warranty](http://www.microscan.com/warranty).

## **Microscan Systems, Inc.**

**United States Corporate Headquarters**  
+1.425.226.5700 / 800.762.1149

**United States Northeast Technology Center**  
+1.603.598.8400 / 800.468.9503

**European Headquarters**  
+31.172.423360

**Asia Pacific Headquarters**  
+65.6846.1214

# Table of Contents

Chapter 1	Quick Start	
	MicroHAWK ID and WebLink Quick Start.....	1-2
	Step 1 Check Hardware and Connect the System .....	1-2
	Step 2 Mount and Position the Reader.....	1-3
	Step 2 Install WebLink Drivers (ID-20 and ID-30) .....	1-4
	Step 4 Connect to WebLink .....	1-5
	Step 5 Explore the Start View.....	1-6
	Step 6 Create a New Setup or Load an Existing Setup .....	1-7
	Step 7 Explore the Setup View.....	1-8
	Step 8 Configure Read Cycle Settings .....	1-9
	Step 9 Configure Acquire Settings .....	1-10
	Step 10 Configure Symbology Settings .....	1-11
	Step 11 Format Output and Configure Match String .....	1-12
	Step 12 Configure Output 1, 2, and 3 Settings .....	1-13
	Step 13 Run the Application .....	1-14
	MicroHAWK ID and ESP Software Quick Start .....	1-15
	Step 1 Check Hardware and Connect the System .....	1-15
	Step 2 Mount and Position the Reader.....	1-16
	Step 3 Install ESP .....	1-17
	Step 4 Select Model .....	1-18
	Step 5 Connect.....	1-19
	Step 6 Locate the Symbol in the Field of View .....	1-20
	Step 7 Calibrate .....	1-21
	Step 8 Test Read Rate .....	1-22
	Step 9 Configure the Reader in ESP .....	1-23
	Step 10 Save Configuration in ESP .....	1-24
	MicroHAWK Part Numbers .....	1-25
Chapter 2	Using ESP	
	EZ Mode .....	2-2
	Application Mode .....	2-3
	Menu Toolbar .....	2-4
	View .....	2-13
	Navigating in ESP.....	2-14
	Send/Receive Options .....	2-15
Chapter 3	Communications	
	Communications by ESP.....	3-2
	Communications Serial Commands .....	3-3
	Host Port Connections.....	3-4
	Host Port Protocol .....	3-5
	ACK/NAK Options .....	3-6
	Polling Mode Options .....	3-7

---

## Table of Contents

Ethernet .....	3-8
Response Timeout .....	3-14
LRC Status .....	3-15
Protocol Configuration Examples .....	3-16
ASCII Character Entry Modifier .....	3-17
Preamble .....	3-18
Postamble .....	3-19
<b>Chapter 4      Calibration</b>	
Calibration Serial Commands.....	4-2
Calibration Overview .....	4-2
Calibration Options .....	4-3
Calibration by ESP .....	4-10
Initiating Calibration .....	4-12
Additional Notes about Calibration .....	4-13
<b>Chapter 5      Read Cycle</b>	
Read Cycle by ESP .....	5-2
Read Cycle Serial Commands .....	5-3
Read Cycle Setup .....	5-4
Multisymbol .....	5-5
Trigger Mode and Filter Duration .....	5-6
External Trigger Polarity .....	5-11
Serial Trigger .....	5-12
Serial Trigger Start Character (Non-Delimited) .....	5-13
Serial Trigger Stop Character (Non-Delimited).....	5-14
End of Read Cycle .....	5-15
Capture Mode.....	5-17
Capture Timing .....	5-21
Image Processing Timeout.....	5-23
Image Storage.....	5-24
Minimum Good Reads.....	5-26
<b>Chapter 6      Symbolologies</b>	
Symbolologies by ESP .....	6-2
Symbolologies Serial Commands .....	6-3
Data Matrix .....	6-4
Aztec .....	6-6
QR Code .....	6-7
Micro QR Code.....	6-8
Code 39 .....	6-9
Code 128/EAN 128 .....	6-12
BC412 .....	6-15
Interleaved 2 of 5.....	6-16
Code 93.....	6-19
Codabar.....	6-20

UPC/EAN .....	6-23
Pharmacode .....	6-27
Postal Symbolologies .....	6-29
GS1 DataBar .....	6-34
PDF417 .....	6-36
MicroPDF417 .....	6-37
Composite .....	6-38
<b>Chapter 7      I/O Parameters</b>	
I/O Parameters by ESP .....	7-2
I/O Parameters Serial Commands .....	7-3
Symbol Data Output .....	7-4
No Read Message .....	7-7
Read Duration Output .....	7-8
Output Indicators .....	7-9
Beeper .....	7-13
LED Configuration .....	7-14
Serial Verification .....	7-15
Configurable Output 1 .....	7-17
Trend Analysis Output 1 .....	7-20
ISO/IEC 16022 Symbol Quality Output 1 .....	7-23
Diagnostic Output 1 .....	7-26
Configurable Output 2 .....	7-27
Trend Analysis Output 2 .....	7-27
ISO/IEC 16022 Symbol Quality Output 2 .....	7-27
Diagnostic Output 2 .....	7-27
Configurable Output 3 .....	7-28
Trend Analysis Output 3 .....	7-28
ISO/IEC 16022 Symbol Quality Output 3 .....	7-28
Diagnostic Output 3 .....	7-28
Power-On/Reset Counts .....	7-29
Time Since Reset .....	7-30
Service Message .....	7-31
Frame Information .....	7-32
Image Output .....	7-33
Database Identifier Output .....	7-36
Quality Output .....	7-37
<b>Chapter 8      Symbol Quality</b>	
Symbol Quality by ESP .....	8-2
Symbol Quality Serial Commands .....	8-3
Overview of Symbol Quality .....	8-4
Symbol Quality Separator/Data Matrix Output Mode .....	8-8
ISO/IEC 16022 Symbol Quality Output .....	8-10
ISO/IEC 16022 Symbol Quality Output by ESP .....	8-12
Microscan Symbol Quality Output .....	8-13

---

## Table of Contents

Microscan Symbol Quality Output by ESP .....	8-16
<b>Chapter 9 Matchcode</b>	
Matchcode by ESP .....	9-2
Matchcode Serial Commands .....	9-3
Overview of Matchcode .....	9-4
Matchcode Type .....	9-5
Match Replace .....	9-10
Mismatch Replace .....	9-11
New Master Pin .....	9-12
<b>Chapter 10 Camera and IP Setup</b>	
Camera and IP Setup by ESP .....	10-2
Camera and IP Setup Serial Commands .....	10-3
Video .....	10-4
Evaluation .....	10-5
Calibration .....	10-8
Window of Interest .....	10-9
Configuration Database .....	10-13
Dynamic Setup .....	10-14
Pixel Binning .....	10-15
Camera Settings .....	10-16
Color Filter .....	10-17
Illumination Brightness .....	10-18
Morphological Pre-Processing .....	10-19
Morphological Operation and Operator Size .....	10-20
<b>Chapter 11 Configuration Database</b>	
Configuration Database Serial Commands .....	11-2
Number of Active Indexes .....	11-3
Configuration Database Status .....	11-4
Database Mode .....	11-9
Save Current Settings to Configuration Database .....	11-14
Load Current Settings from Configuration Database .....	11-15
Request Selected Index Settings .....	11-16
Request All Configuration Database Settings .....	11-16
<b>Chapter 12 Terminal</b>	
Terminal .....	12-2
Find .....	12-3
Send .....	12-4
Macros .....	12-5
Terminal Menus .....	12-6
<b>Chapter 13 Utilities</b>	
Serial Utility Commands .....	13-2
Read Rate .....	13-4
Counters .....	13-6

Device Control .....	13-8
Differences from Default .....	13-9
Master Database .....	13-10
Firmware .....	13-16
Default/Reset/Save .....	13-19
Reader Status Requests .....	13-21
Other Operational Serial Commands .....	13-23
<b>Chapter 14      Output Format</b>	
Output Format Serial Commands .....	14-2
Output Format Status .....	14-3
Format Assign .....	14-4
Format Extract .....	14-5
Format Insert .....	14-7
Output Filter Configuration .....	14-9
Number of Filters .....	14-14
<b>Appendices</b>	
Appendix A General Specifications .....	A-2
Appendix B Electrical Specifications .....	A-6
Appendix C Serial Commands .....	A-8
Appendix D Communications Protocol .....	A-14
Appendix E ASCII Table .....	A-15
Appendix F Glossary of Terms .....	A-16

## About MicroHAWK ID Readers

The key features of the MicroHAWK ID-20, ID-30, and ID-40 Readers are:

- World's smallest industrial barcode readers
- Able to read any symbol on any surface
- Intuitive WebLink user interface
- Customizable hardware options
- Customizable software options

## MicroHAWK ID Communications

There are three ways to configure and test the MicroHAWK ID-20, ID-30, and ID-40:

- Microscan's browser-based **WebLink** user interface, which enables you to access, configure, and test your reader without having to install software or access files on a host system;
- Microscan's **ESP Software**, which allows you to configure your reader using simple tree controls and UI elements;
- Serial commands that can be sent from the **Terminal** in WebLink or ESP, or from another terminal program.

## About This Manual

The purpose of this user manual is to document the hardware and firmware of the MicroHAWK ID series of readers, as well as the software used to configure and operate those readers. The browser-based WebLink interface is the preferred configuration tool for MicroHAWK ID readers, but ESP Software is useful for certain operations, such as:

- Device discovery to find the reader IP address;
- If you only have an RS-232 (serial) connection;
- Updating MicroHAWK ID firmware;
- Using the Configuration Database;
- Creating barcodes for reader configuration;
- Generating Symbol Quality reports.

Parts of this manual introduce the basic functionality of WebLink, but *WebLink Help* in the WebLink user interface provides more detailed information. This manual's primary focus is MicroHAWK ID firmware and ESP Software.



## Statements of Compliance



### ***FCC***

Pending.

### ***CE***

Pending.

### ***UL***

Pending.

## ***Restricted Substances***

See the following link for Microscan compliance statements related to all applicable restricted substances:

<http://www.microscan.com/en-us/AboutUs/quality-environmental-management-system.aspx>

---

## Warning and Caution Summary

**CAUTION:** Viewing the MicroHAWK ID-20, ID-30, or ID-40's LED output with optical instruments such as magnifiers, eye loupes, or microscopes within a distance of 100 mm could cause serious eye injury.

**CAUTION:** Use of controls or adjustments or performance of procedures other than those specified herein may result in hazardous radiation exposure.

# 1 Quick Start

## Contents

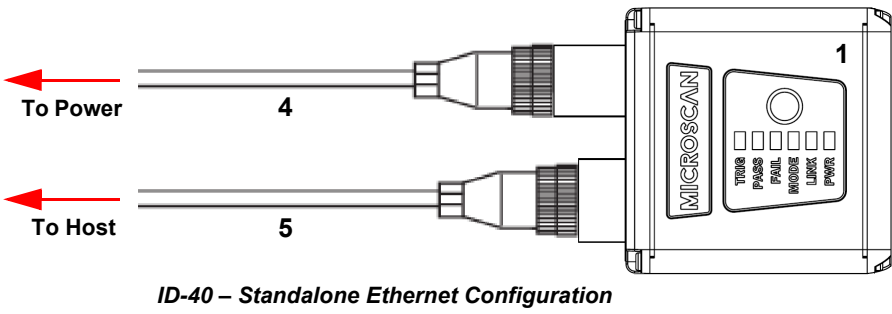
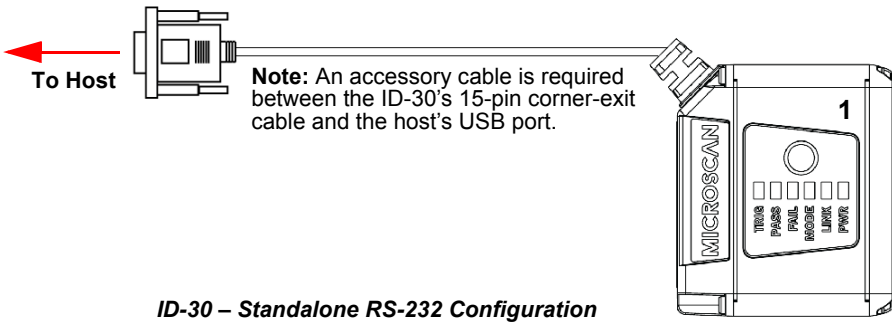
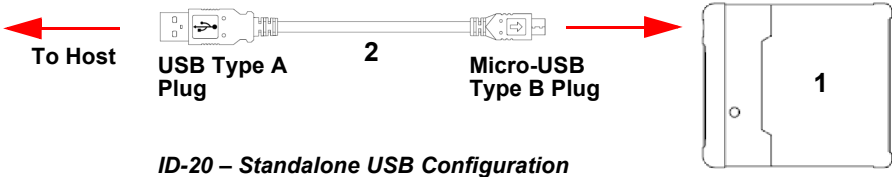
MicroHAWK ID and WebLink Quick Start .....	1-2
Step 1 Check Hardware and Connect the System .....	1-2
Step 2 Mount and Position the Reader .....	1-3
Step 3 Install WebLink Drivers (ID-20 and ID-30) .....	1-4
Step 4 Connect to WebLink .....	1-5
Step 5 Explore the Start View .....	1-6
Step 6 Create a New Setup or Load an Existing Setup .....	1-7
Step 7 Explore the Setup View .....	1-8
Step 8 Configure Read Cycle Settings .....	1-9
Step 9 Configure Acquire Settings .....	1-10
Step 10 Configure Symbology Settings .....	1-11
Step 11 Format Output and Configure Match String .....	1-12
Step 12 Configure Output 1, 2, and 3 Settings .....	1-13
Step 13 Run the Application .....	1-14
 MicroHAWK ID and ESP Software Quick Start .....	 1-15
Step 1 Check Hardware and Connect the System .....	1-15
Step 2 Mount and Position the Reader .....	1-16
Step 3 Install ESP .....	1-17
Step 4 Select Model .....	1-18
Step 5 Connect .....	1-19
Step 6 Locate the Symbol in the Field of View .....	1-20
Step 7 Calibrate .....	1-21
Step 8 Test Read Rate .....	1-22
Step 9 Configure the Reader in ESP .....	1-23
Step 10 Save Configuration in ESP .....	1-24
 MicroHAWK Part Numbers .....	 1-25

This section is designed to get your MicroHAWK up and running quickly with **WebLink** or **ESP Software**. Following these steps will allow you to get a sense of the reader's capabilities and to test sample symbols.

# MicroHAWK ID and WebLink Quick Start

## Step 1 — Check Hardware and Connect the System

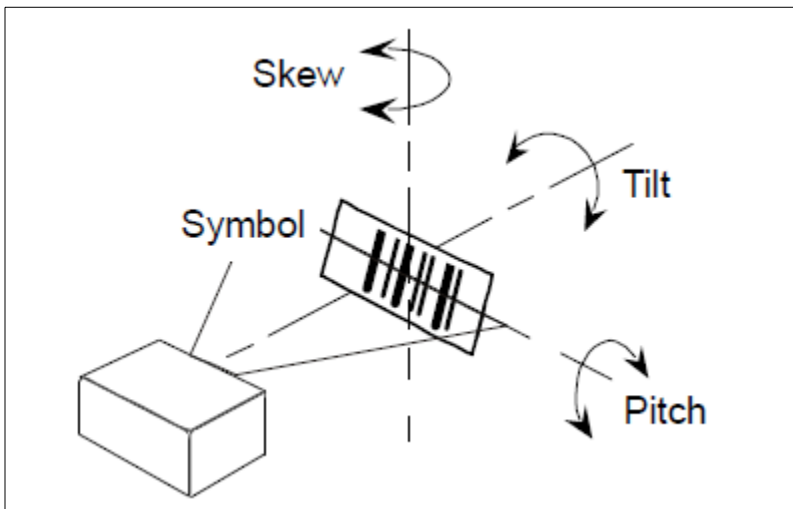
Item	Description	Part Number
1	MicroHAWK ID-20, MicroHAWK ID-30, or MicroHAWK ID-40	7AXY-YZZZ-LPPP
2	Micro-USB Type B Plug to USB Type A Plug, 6 ft. (High Speed USB 2.0)	61-9000034-01
3	Ferrite Core Snap-On Kit for USB Cable (for Class B Emissions; not shown)	98-9000035-01
4	Power Supply, 100-240VAC, +24VDC, M12 12-pin Socket	97-000012-01
5	Host Ethernet Cordset, M12 8-pin Plug (Screw-On) to RJ45, 1 m	61-000160-02



## Step 2 — Mount and Position the Reader

- Position the reader several inches from the symbol. You may need to reposition the reader a few times to find the ideal distance.
- Tip the reader relative to the symbol to avoid the glare of direct (specular) reflection.
- Symbols can be rotated (tilted) at any angle; however, for best results symbols should be aligned with the field of view. In the case of linear symbols, aligning the bars in the direction of their movement (ladder orientation) will minimize the chances of blurring and will result in more consistent decodes.

**Important:** Avoid excessive skew or pitch. Maximum skew is  $\pm 30^\circ$ ; maximum pitch is  $\pm 30^\circ$ . The illustration below shows approximate skew axis, pitch axis, and tilt axis.



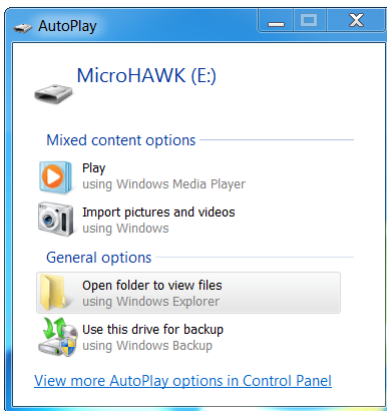
**Reader and Symbol Orientation**

## Step 3 — Install WebLink Drivers (ID-20 and ID-30)

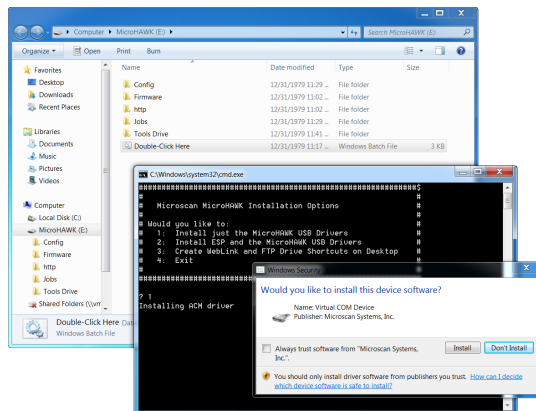
1. Plug the reader into a USB port and wait for the **AutoPlay** dialog to appear.
2. Click **Open folder to view files** and double-click the **Double-Click Here.bat** batch file.
3. At the command prompt, select **option 1** and then type **Enter**. **VCOM** and **USBLAN** drivers are installed.
4. At the command prompt, select **option 3** to install the WebLink and FTP drive shortcuts. WebLink and MicroHAWK FTP drive shortcut icons will appear on the desktop.
5. When installation of the drivers and shortcuts is complete, unplug the reader from the USB port.
6. Plug the the reader back into the USB port and wait for the reader to reboot and to enter read mode (LEDs **ON**).
7. Double-click the WebLink desktop shortcut. WebLink will load and start.
8. Double-click the FTP drive shortcut and log in with username: **target** and password: **password**.
9. The FTP drive is opened so you can access additional resources and installers in the **Tools and Documentation** folder.

You are now ready to use the MicroHAWK ID-20 or ID-30 with WebLink.

### Step 1



### Step 3



## Step 4 — Connect to WebLink

### MicroHAWK ID-20 and MicroHAWK ID-30

When you double-click the WebLink desktop shortcut or enter the reader's IP address directly in the address bar of your web browser, WebLink will load and start.

**Note:** WebLink is the preferred user interface for MicroHAWK ID Readers, but Microscan's **ESP Software** can also be used for configuration and testing. ESP is useful in the following circumstances:

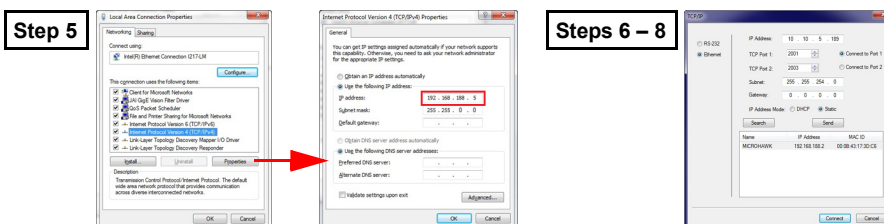
- Device discovery to find the reader IP address
- If you only have an RS-232 (serial) connection
- Updating MicroHAWK ID firmware
- Using the Configuration Database
- Creating barcodes for reader configuration
- Generating Symbol Quality reports



### MicroHAWK ID-40

A slightly different method is required to connect to the MicroHAWK ID-40:

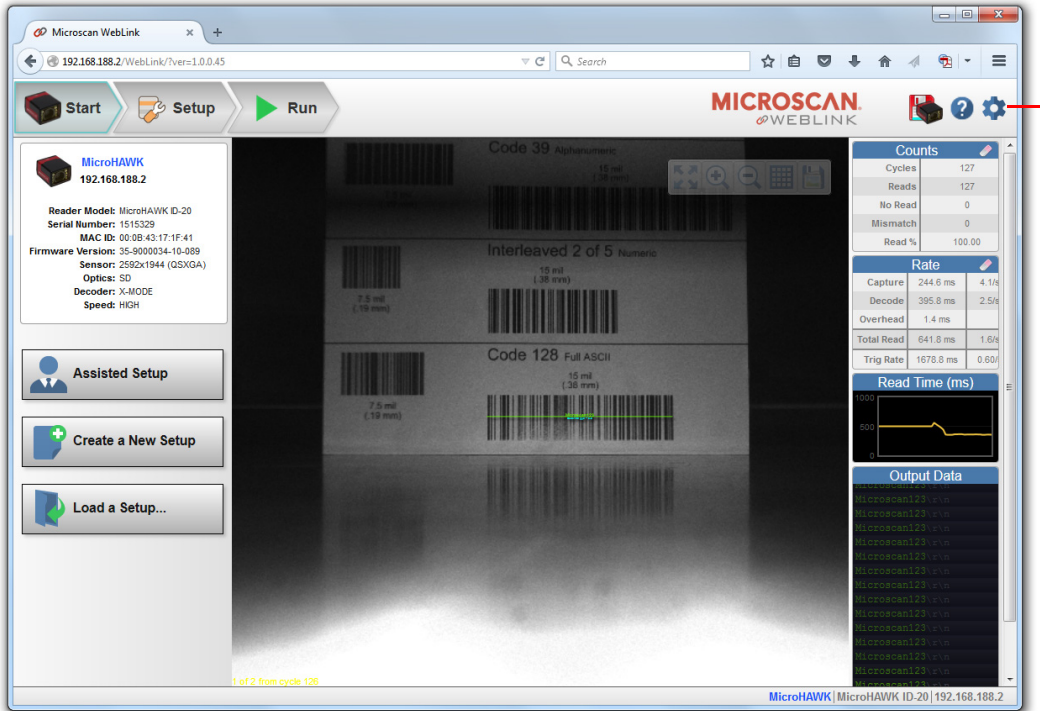
1. Plug your reader into the appropriate connector and open the web browser of your choice.
2. Type **http://192.168.188.2** (the default IP address) in the web browser's address bar.
3. Navigate to **Control Panel > Network and Sharing Center** on your PC.
4. Click **Local Area Connection 4**. In the **Status** dialog, click **Properties**.
5. In the **Local Area Connection Properties** dialog, select **Internet Protocol Version 4 (TCP/IPv4)** and click **Properties** again. Set your PC to a **192.168.188.x** address (192.168.188.5, for example).
6. Open **ESP Software**, connect to the MicroHAWK ID-40 via Ethernet TCP/IP, and click **Search** to find the reader. When the reader appears in the field below the Search and Send buttons, select it.
7. Set the ID-40's IP address to match that of the host PC.
8. Click **Send**. The camera will reboot and ESP will search for the reader again. When the ID-40 is found, you can use it with WebLink.



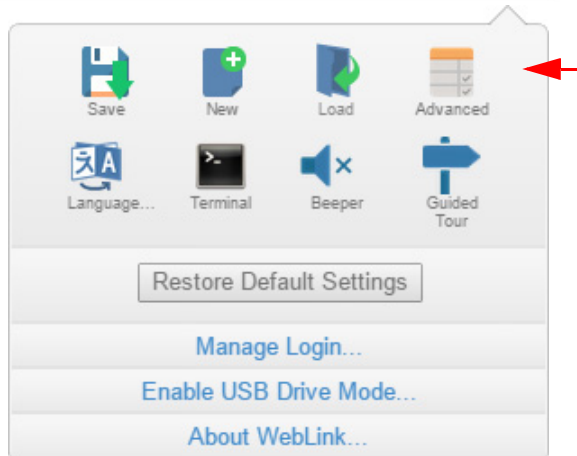
**To create an FTP drive shortcut:** right-click on your desktop and select **New > Shortcut**; enter **%windir%\explorer.exe ftp://192.168.188.2** as the target path; name your FTP drive; click **Finish**. Double-click the FTP drive shortcut on the desktop; log in with the username: **target** and password: **password** credentials to navigate the reader's file system.

## Step 5 — Explore the Start View

The **Start View** is the initial view you will see when the session begins. The connected reader is shown, along with its **user-defined name**, **IP address**, **Reader Model**, **Serial Number**, **MAC ID**, **Firmware Version**, **Sensor**, **Optics**, **Decoder**, and **Speed**. This view allows you to choose **Assisted Setup**, to **Create a New Setup**, or to **Load a Setup**.



Click the gear icon to show **Save**, **New**, **Load**, **Advanced**, **Language**, **Terminal**, **Beeper**, **Guided Tour**, **Restore Default Settings**, **Manage Login**, **Enable USB Drive Mode**, and **About WebLink**.





## Step 6 — Create a New Setup or Load an Existing Setup

### Assisted Setup

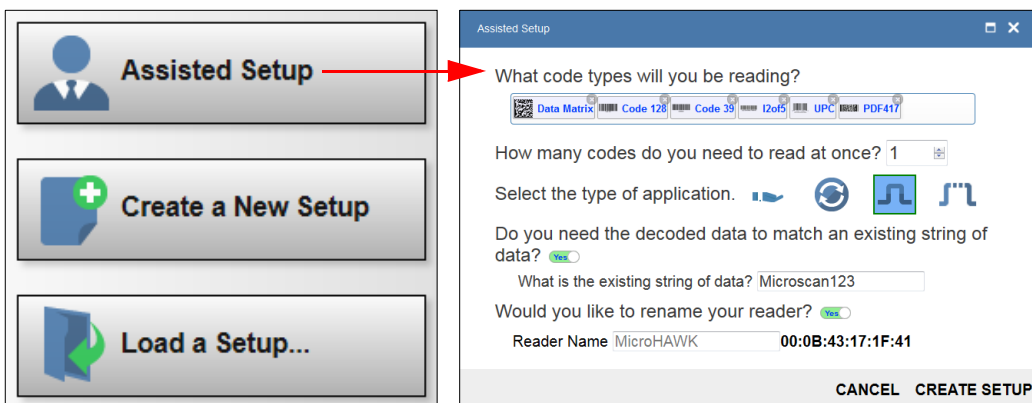
When you click the **Assisted Setup** button in the Start View, a dialog will appear asking you a series of application-based questions. Based on your answers, WebLink generates your initial setup automatically. Once the setup is created, you can fine-tune its parameters in the **Setup View**.

### New Setup

The Start View also allows you to create a **New Setup** without using Assisted Setup. When you click the New Setup button, WebLink searches for any differences from default in the reader parameters. If no differences from default are found, you will see the **Setup View**. If differences from default are found, an alert will appear asking if you want to restore the reader to factory defaults.

### Load Setup

Select **Load Setup** to load an existing .json WebLink setup file.



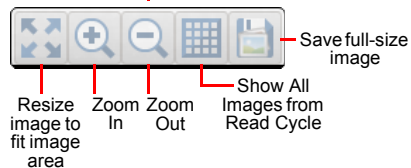
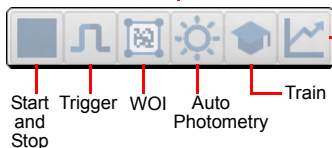
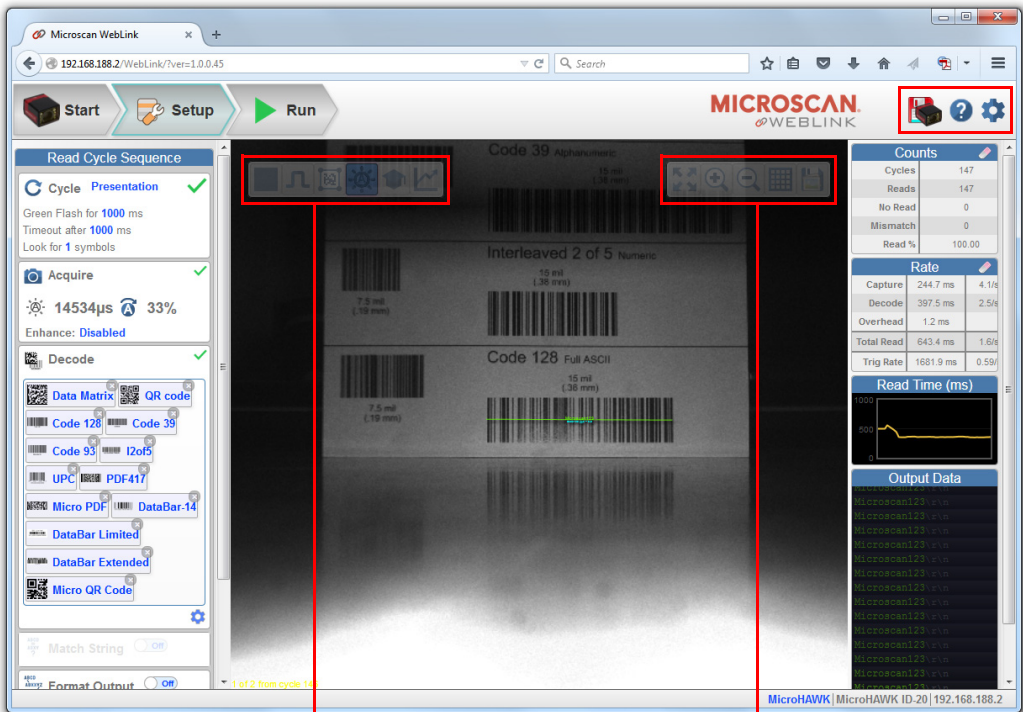
## Step 7 — Explore the Setup View

The **Setup View** allows you to configure all aspects of a setup. The left pane of the UI gives you the ability to configure **Cycle type**, **Acquire**, **Decode**, **Match String**, **Format Output**, and **Outputs 1, 2, and 3** parameters.

Clicking the **Save** icon in the upper right of the interface saves current settings to the reader's flash memory so the settings will be available when the reader is rebooted.

The **question mark** icon in the upper right of the interface opens **WebLink Help**.

The **gear** icon in the upper right of the interface brings up **Application Settings**.



## Step 8 — Configure Read Cycle Settings

The **Cycle** section of the **Setup View** allows you to modify the trigger, determine the number of symbols for the reader to expect, and set Read Cycle Timeout. A dropdown menu of various Cycle types provides a variety of options, each with configurable parameters.

The screenshot shows the 'Cycle' section of the 'Setup View' with three cycle types and a 'Start/Stop' section. Each cycle type has a green checkmark icon.

- Cycle Presentation**
  - Green Flash for 1000 ms
  - Timeout after 1000 ms
  - Look for 1 symbols
- Cycle Continuous**
  - Timeout after 1000 ms
  - Look for 1 symbols
- Cycle Triggered**
  - Serial Trigger Character <SP>
  - Trigger Delay 32µs
  - Timeout after 1000 ms
  - Look for 1 symbols

The **Start/Stop** section includes:

- Trigger Filter Duration**
  - Leading Edge 32 us
  - Trailing Edge 32 us
- Serial Trigger (Non-Delimited):**
  - ☒ On
  - Start Character S
  - Stop Character E
- Timeout after 1000 ms
- Look for 1 symbols

### Presentation

This mode, commonly referred to as “hand presentation mode”, uses **Continuous Trigger** along with **Continuous Capture Mode** and a **Timeout at End of Read Cycle**. **Green Flash Mode** is set to **Static Presentation** and the **Green Flash Duration** is set to 1 second.

### Continuous

This mode allows you to set the **Read Cycle Timeout** and the expected number of symbols.

### Triggered

This mode sets the Read Cycle to **Serial Data and Edge** (see the **Read Cycle** section for more information about this parameter), **End of Read Cycle** is set to **Timeout or New Trigger**, and **Capture Mode** is set to **Rapid Capture** with 1 capture. You can adjust the **Serial Trigger**, **Trigger Delay**, **Timeout**, and **Number of Symbols**.

### Start / Stop

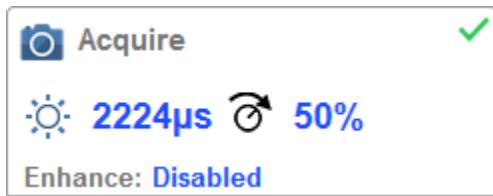
This mode uses **External Level** with a **Read Cycle Timeout** and **Continuous Capture**, allowing you to set **Leading Edge** and **Trailing Edge** as well as the **Serial Trigger** and the **Start and Stop Characters**.

### Custom

This mode allows you a wider variety of read cycle scenarios. Use this mode to select **Trigger** mode and to set **Serial Trigger Character** and **Trigger Delay**; to select **Capture Mode** and to set **Number of Captures**, **Rapid Capture Mode**, and **Delay between Images**; and to select the **End Cycle On** setting as well as **Timeout** and **Number of Symbols**.

## Step 9 — Configure Acquire Settings

**Acquire** settings allow you to set **Exposure** (signified by the sun icon) and **Gain** (signified by the dial and right-pointing arrow icon) in real time. Clicking any of these settings will cause a control to appear, allowing you to modify that setting. Settings take effect immediately.



**Standard**

When **Auto Photometry** is enabled instead of **Standard**, Exposure and Gain are read-only. The **A** shown on the sun and dial icons signifies that **Auto Photometry** is enabled. Auto Photometry constantly determines the best Exposure and Gain settings during each read cycle.



**Auto Photometry**

# Step 10 — Configure Symbology Settings

Clicking the gear icon at the bottom of the **Decode** dialog brings up **Symbology Settings**. This allows you to configure every parameter for every available code type.

Symbology Settings

Data Matrix	☆	ECC 200 Status	Enabled
Code 128	☆	ECC 000 Status	Disabled
Code 39	☆	ECC 050 Status	Disabled
Codabar	☆	ECC 080 Status	Disabled
Code 93	☆	ECC 100 Status	Disabled
Interleaved 2 of 5	☆	ECC 140 Status	Disabled
UPC/EAN	☆	ECC 120 Status	Disabled
	☆	ECC 130 Status	Disabled
PDF417			
Micro PDF417			
BC412			
Pharmacode			
DataBar Expanded			
Postal Symbologies			

Data Matrix error correction parameters are shown in this example, but you can configure any parameter for any of the code types supported by WebLink. All parameter changes for all code types take effect immediately.

## Step 11 — Format Output and Configure Match String

**Format Output**, when enabled in the **Setup View**, allows you to determine the many ways in which barcode data can be formatted and parsed before it is output as a data string. You can also set **Preamble** and **Postamble** in this dialog.

The **Output Formatting Editor** dialog box is shown. It has a title bar with a close button. The main content is divided into two sections: **Output String** and **Symbol Parsing**.

**Output String** section:

- ☒ **Preamble**: A text field containing `<CR>`.
- Symbol 1**: A selection rule box showing a barcode icon, the text "Any Symbol", and "Data: \x00". Below it is ☒ **Parse Symbol?**.
- ☒ **Postamble**: A text field containing `<CR><LF>`.

A blue arrow points down to the **Symbol Parsing** section.

**Symbol Parsing** section:

- Extract chars**: A button with a plus icon and a range of `[1 - 10]`.
- Parsing Actions**: A dropdown menu with "Extract chars from symbol data" selected and "Insert your own text" as an option.
- Example Data:** A text field.
- Parsing Result:** A text field.

A **DONE** button is at the bottom right.

**Match Options** and **Match String Database**, accessible by clicking the **Match String** section in the **Setup View**, allow you to set the match code mode, text output, new master, and match string database.

The **Match String Editor** dialog box is shown. It has a title bar with a close button. The main content is divided into two sections: **Match Options** and **Match String Database**.

**Match Options** section:

- Mode:** **Standard**.
- Settings for Match String:**
  - Range of Characters to Match Against:**
    - ☒ **Match All**
    - ☐ **Partial Match:** Start: 0 Length: 1
- Text Output Options:**
  - ☒ **Match Replace:** **MATCH**
  - ☒ **Mismatch Replace:** **MISMATCH**
- Train Match String on New Master Input:**
  - New Master Pin:** **Enabled**

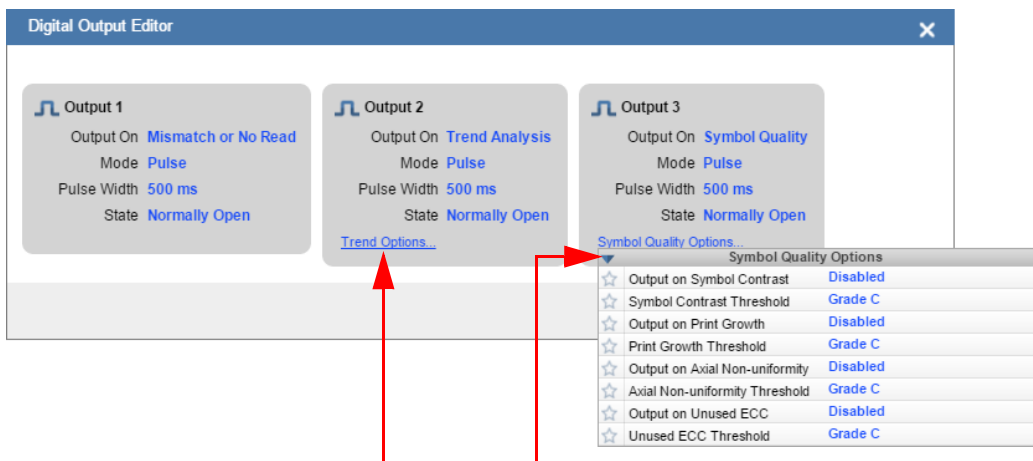
**Match String Database** section:

- A list with one entry: **1 MICROSCAN123**.
- A plus icon button to add a new entry.

A **DONE** button is at the bottom right.

## Step 12 — Configure Output 1, 2, and 3 Settings

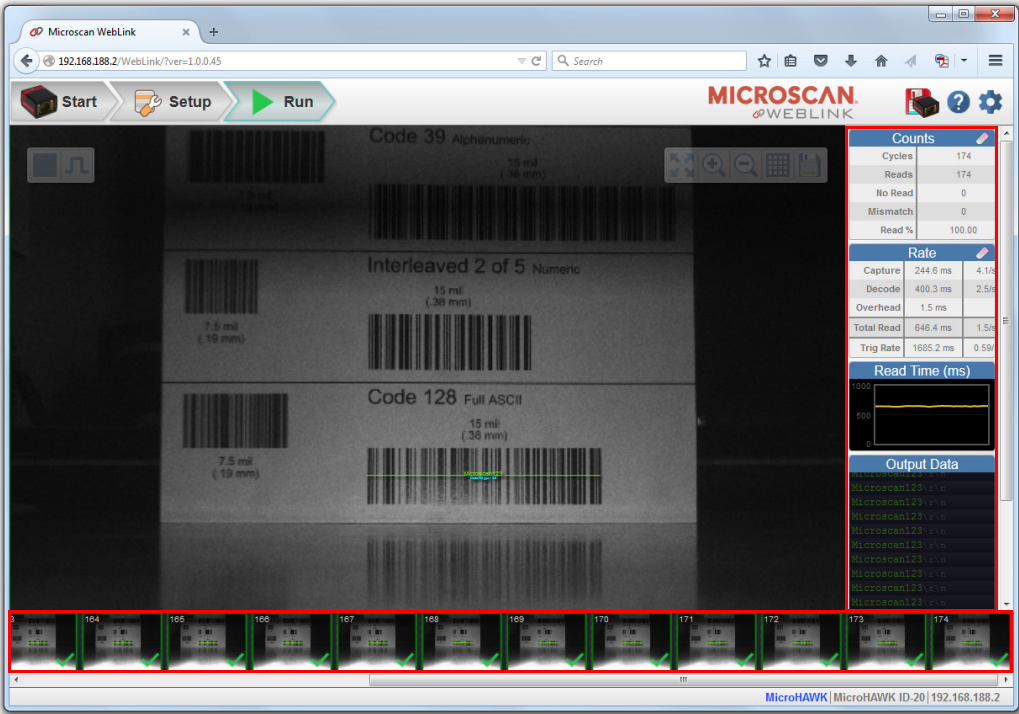
Bring up the **Output 1**, **Output 2**, and **Output 3** dialogs by clicking on the **Outputs** section at the lower left of the **Setup View**. For each output you can determine the **Output On** behavior, **Mode**, **Pulse Width**, and **State**. Outputs 2 and 3 also allow you to determine output behavior based on **Trend Options** parameters or **Verification Options** parameters.



This example shows the dialog that appears when you click **Symbol Quality Options**. A similar dialog would appear for **Trend Options** under **Output 2**.

# Step 13 — Run the Application

In the **Run** view, you can observe the progress of the setup as it follows the parameters you have defined. The right pane of the UI shows **Counts** for **Inspected**, **Passed**, **No Read**, and **Mismatch**, as well as **Read Rate**, **Reads per Second**, **Read Time** (shown numerically and visually), and **Output Data**. A “filmstrip” in the bottom pane shows each image capture with a **green check mark** for good reads and a **red x** for no reads.

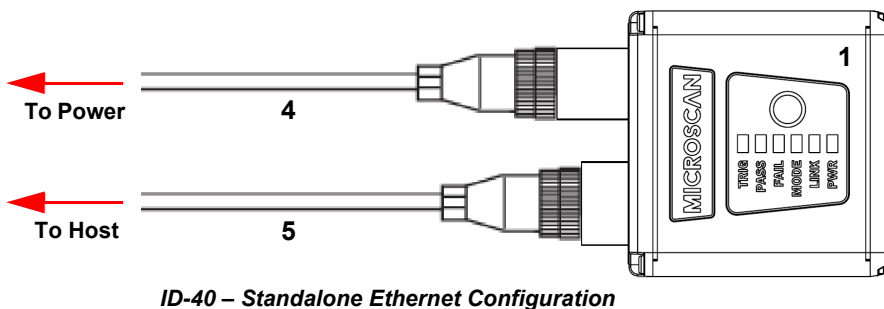
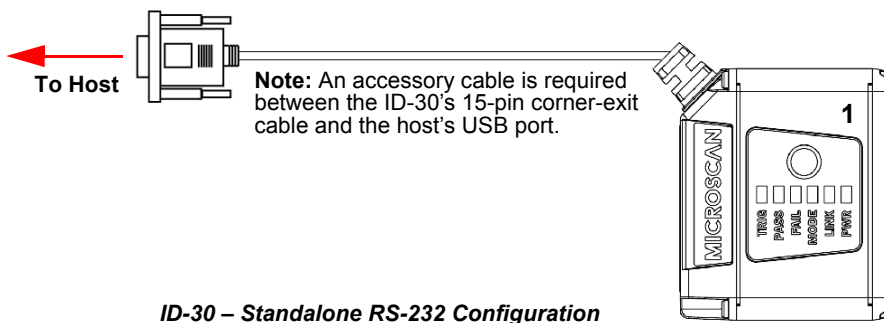
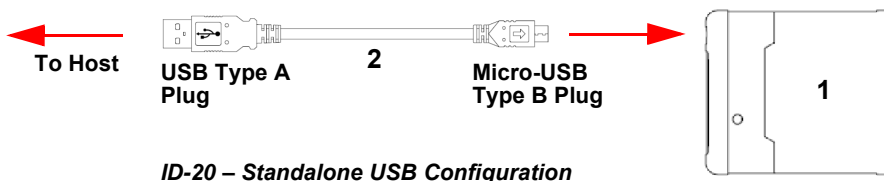




# MicroHAWK ID and ESP Software Quick Start

## Step 1 — Check Hardware and Connect the System

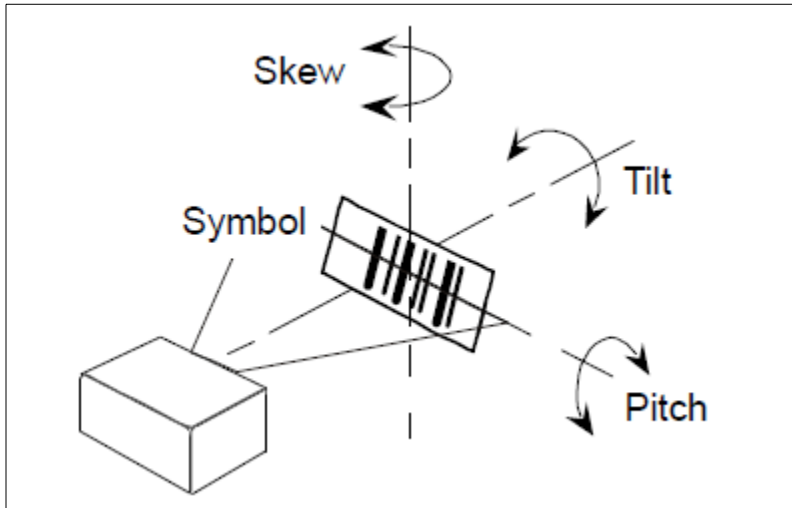
Item	Description	Part Number
1	MicroHAWK ID-20, MicroHAWK ID-30, or MicroHAWK ID-40	7AXY-YZZZ-LPPP
2	Micro-USB Type B Plug to USB Type A Plug, 6 ft. (High Speed USB 2.0)	61-9000034-01
3	Ferrite Core Snap-On Kit for USB Cable (for Class B Emissions; not shown)	98-9000035-01
4	Power Supply, 100-240VAC, +24VDC, M12 12-pin Socket	97-000012-01
5	Host Ethernet Cordset, M12 8-pin Plug (Screw-On) to RJ45, 1 m	61-000160-02



## **Step 2 — Mount and Position the Reader**

- Position the reader several inches from the symbol. You may need to reposition the reader a few times to find the ideal distance.
- Tip the reader relative to the symbol to avoid the glare of direct (specular) reflection.
- Symbols can be rotated (tilted) at any angle; however, for best results symbols should be aligned with the field of view. In the case of linear symbols, aligning the bars in the direction of their movement (ladder orientation) will minimize the chances of blurring and will result in more consistent decodes.

**Important:** Avoid excessive skew or pitch. Maximum skew is  $\pm 30^\circ$ ; maximum pitch is  $\pm 30^\circ$ . The illustration below shows approximate skew axis, pitch axis, and tilt axis.



***Reader and Symbol Orientation***

## Step 3 — Install ESP

**ESP Software** can be found on the Microscan Tools Drive that is packaged with the MicroHAWK ID. ESP can also be installed from the **Download Center** at [www.microscan.com](http://www.microscan.com).

- Follow the prompts to install ESP from the Microscan Tools Drive or Download Center.
- Click on the ESP icon to run the program.

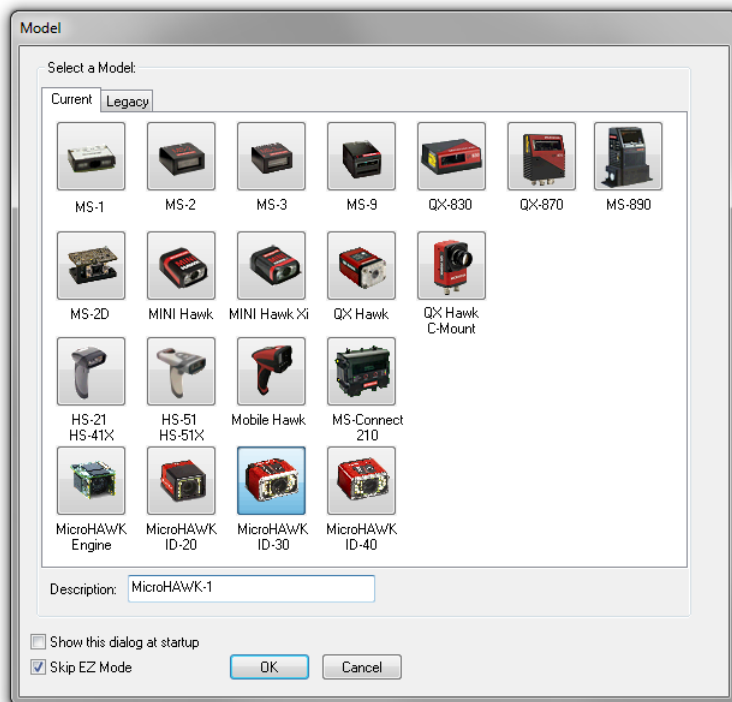


### Minimum System Requirements

- 233 MHz Pentium® PC
- Windows® 8, 7, Vista, or XP operating system (32-bit or 64-bit)
- Internet Explorer® 6.0 or higher
- 128 MB RAM or greater
- 160 MB free disk space
- 800 x 600 256 color display (1024 x 768 32-bit color recommended)

## Step 4 — Select Model

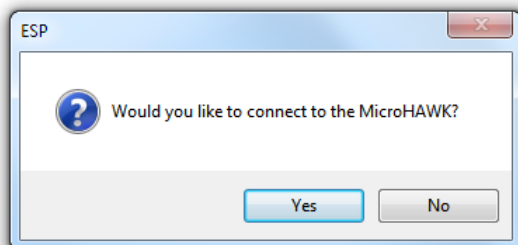
When you start **ESP**, the following menu will appear:



1. Click the button showing your version of the MicroHAWK ID.
2. Click **OK**.

**Note:** You can also double-click the MicroHAWK ID button to make your selection.

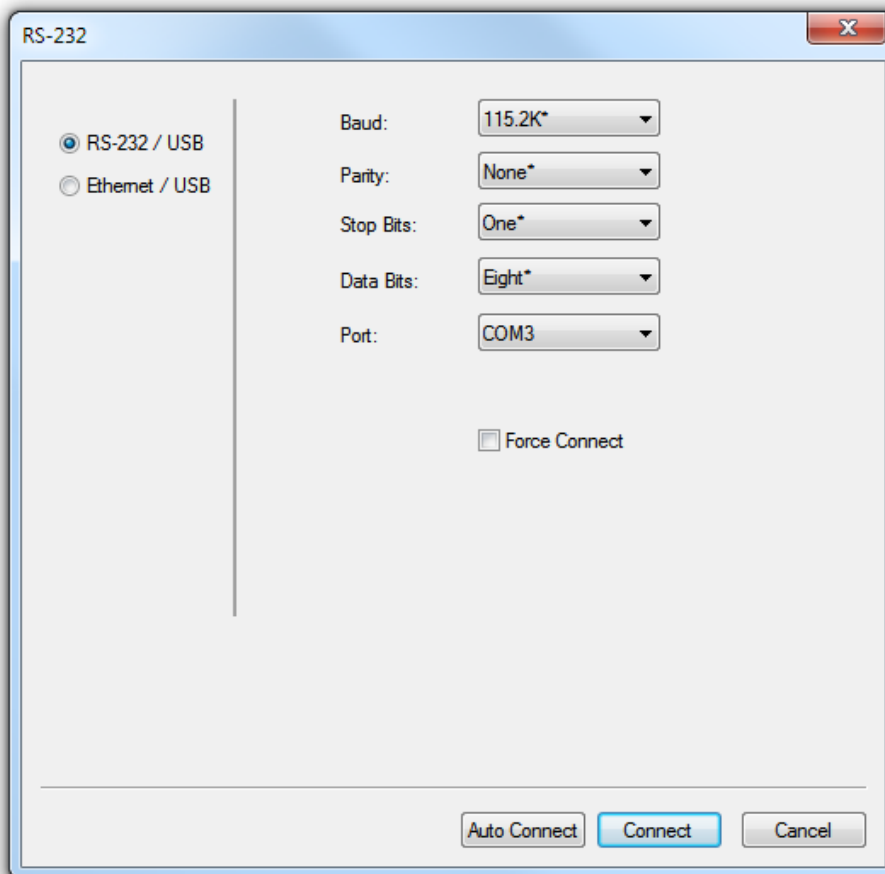
3. Click **Yes** when this dialog appears:



**Note:** If you need to select another model later, click the **Switch Model** button near the top of the screen or use **Model > New Model** in the menu toolbar.

## Step 5 — Connect

- Click **Connect** on the menu toolbar, and then select **Connection Wizard**.
- Select the communication interface required by your application.
- Configure settings as required by the application, and click **Connect**.

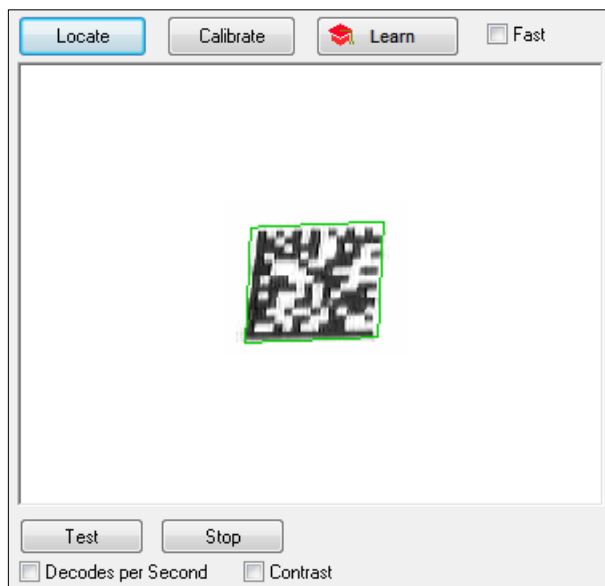


- When a connection is established, the green indicator in the status bar at the bottom right of the screen will be visible:



## Step 6 — Locate the Symbol in the Field of View

- In ESP's **EZ Mode**, click the **Locate** button to enable the blue target pattern.

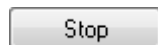


The symbol in the field of view will appear in the video view beneath the **Locate** and **Calibrate** buttons, and you will see the blue target pattern projected from the front of the reader.

- Center the target pattern on the symbol.

**Important:** The entire symbol should fall within the field of view (FOV) of the reader. The field of view is what appears in **ESP's Locate/Calibrate** window in **EZ Mode**.

- Click the **Stop** button to end the **Locate** function.



## Step 7 — Calibrate

MicroHAWK ID settings can be adjusted automatically in **ESP** for optimum symbol decoding performance.

During the calibration routine, the reader will flash its amber Read Rate percent LEDs and red illumination LEDs while searching camera settings and determining the best configuration for decoding symbol data. Upon successful completion of this routine, a green LED pattern will flash brightly and illuminate the symbol. If unsuccessful, the reader will emit 5 short beeps and stop searching.

### Calibrate by ESP

1. Click the **Calibrate** button.



2. The reader will search camera settings to determine the best configuration for decoding symbol data.

A successful calibration will display a green frame around the symbol, and the following message will appear: "Uploading all reader parameters." After a moment the symbol data will be presented in the field below the image display window.

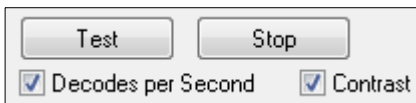
**Note:** If you choose to calibrate the reader by sending a **<@CAL>** command, a decode is also required for the calibration process to be completed successfully.

## Step 8 — Test Read Rate

**Read Rate** indicates the number of successful decodes per second achieved by the reader.

### Test Read Rate by ESP

1. Click the **Test** button to start the Read Rate test and **Stop** to end it.



If a symbol has been successfully decoded, the symbol's data and related features will be presented in the field below the image display window. Also, while the symbol is being inspected, the Read Rate LEDs will indicate the corresponding Read Rate percentage on the back of the unit.

2. To end the test, click the **Stop** button.

**Note:** Read Rate can also be tested using the **Read Rate** interface in **Utilities**.

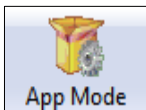
### Test Read Rate by Serial Command

You can also start a test with the **<C>** or **<Cp>** command and end it with the **<J>** command.

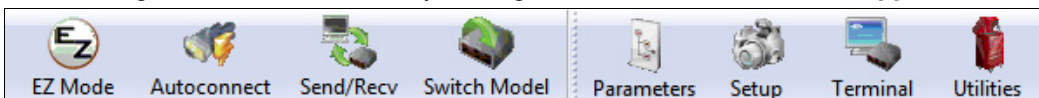


## Step 9 — Configure the Reader in ESP

Click the **App Mode** button to make setup changes to the MicroHAWK ID.

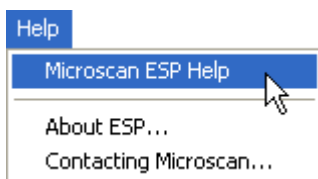


The following modes are accessible by clicking the buttons in the first row of **App Mode** icons:



- Click the **EZ Mode** button to return to EZ Mode.
- Click the **Autoconnect** button to establish communication.
- Click the **Send/Recv** button to send or receive commands.
- Click the **Switch Model** button to open the model menu, or to return to a previous model.
- Click the **Parameters** button to show the tabbed tree controls for Communication, Read Cycle, Symbolologies, I/O Parameters, Symbol Quality, Matchcode, and Diagnostics.
- Click the **Setup** button to access a Camera Setup tree control and Video view, and to Evaluate image captures, Calibrate the reader, set the Window of Interest, fine-tune capture settings and processing settings in the Configuration Database, set up output filters and parse symbol data in Ordered Output and Output Format, and control multiple read cycle functions in Dynamic Setup.
- Click the **Terminal** button to display tube or cap detection data or decoded symbol data, and to send serial commands to the reader using text or macros.
- Click the **Utilities** button to test Read Rate, request or clear Counters, enable or disable the reader or send output pulses in Device Control, determine the Differences from Default in the current settings, add or remove master symbol data in Master Database, and verify or update the reader's Firmware.

For further details, see **ESP Help** in the dropdown Help menu.



# Step 10 — Save Configuration in ESP

To make changes to a configuration setting:

- 1. **Left-click** on the **+** to expand the desired tree.
- 2. **Double-click** on the desired parameter and click once in the selection box to view options.

Parameters	ESP Values
Communications	
RS232	
Baud Rate	
Parity	
Stop Bits	
Data Bits	
Host Protocol	
Ethernet	
Preamble	
Postamble	Enabled*
Postamble Characters	Disabled
Response Timeout	Enabled*
LRC Status	Disabled
Search and Configure Mode	Enabled

- 3. Place your cursor in the selection box, scroll down to the setting you want to change, and **click** once on the setting.
- 4. **Left-click** again on the open screen to complete your selection.
- 5. **Right-click** on the open screen and select **Save to Reader** to implement the command in the reader.

## Saving Options

- **Send, No Save.** Changes will be lost when power is re-applied to the reader.
- **Send and Save.** This activates all changes in current memory *and* saves to the reader for power-on.

## MicroHAWK ID Part Numbers

MicroHAWK ID part numbers follow the format **7ABX-YZZZ-LPPP**.

**7 = MicroHAWK.**

### Example Part Number: 7312-2102-1005

**Description:** MicroHAWK ID-30, Auto ID, SXGA 1.2 Megapixel, Mono, High Density, 102 mm Focus Distance, Red Outer LEDs, High Speed X-Mode Decoder.

#### (A) Model

- 1: Engine, No Case, USB
- 2: ID-20, IP40 Case, USB
- 3: ID-30, IP54 Case, 5V, USB
- 4: ID-40, IP65 Case, 24V, Ethernet

#### (B) Software

- 1: Auto ID

#### (X) Sensor

- 1: WVGA 0.3 Megapixel, Mono
- 2: SXGA, 1.2 Megapixel, Mono
- 3: QSXGA, 5 Megapixel, Color

#### (Y) Optics

- 0: Custom
- 1: Standard Density
- 2: High Density

#### (ZZZ) Focus Distance

- 050:** 50 mm = 1.96 in.
- 064:** 64 mm = 2.51 in.
- 081:** 81 mm = 3.18 in.
- 102:** 102 mm = 4.02 in.
- 133:** 133 mm = 5.23 in.
- 190:** 190 mm = 7.48 in.
- 300:** 300 mm = 11.81 in.

#### (L) Outer LED Color

- 0: N/A (Engine and ID-20)
- 1: Red
- 2: White

#### (PPP) Speed and Decoder

- 000:** Standard Speed
- 001:** High-Speed
- 002:** Plus (1D/2D Decoder)
- 003:** X-Mode Decoder
- 004:** High-Speed Plus (1D/2D Decoder)
- 005:** High-Speed X-Mode Decoder

#### Notes:

- **(L) Outer LED Color:** Outer LEDs provide extra illumination. Base level illumination included with all readers.
- **(PPP) Speed and Decoder:** 1D omnidirectional decoding functionality is included with all readers. The "Plus" option includes both 1D and 2D, and is useful for printed labels. X-Mode is the full decoder package for 1D, 2D, DPM, and damaged symbols.
- **Field Upgrades:** Not available for optics or illumination due to factory settings for optical alignment, LED balancing, and sealing for IP enclosure rating. However, the reader's speed and decoder version are field-upgradeable via licenses.
- **(A) Model:** MicroHAWK Engine is available for OEM-certified partners only.

**Sensor Table**

	<b>Pixels (H x V)</b>	<b>Shutter</b>	<b>Frames per Second (Standard / High)</b>
<b>WVGA</b>	752 x 480, 0.3 MP, Mono	Global	10 fps / 60 fps
<b>SXGA</b>	1280 x 960, 1.2 MP, Mono	Global	10 fps / 42 fps
<b>QSXGA</b>	2592 x 1944, 5 MP, Color	Rolling	5 fps





# 2 Using ESP

## Contents

EZ Mode.....	2-2
Application Mode.....	2-3
Menu Toolbar.....	2-4
View .....	2-13
Navigating in ESP .....	2-14
Send/Receive Options .....	2-15

This section is designed to help you understand the basic structure and elements of **ESP** (Easy Setup Program).

When you open **ESP**, unless otherwise specified in the **ESP Preferences** dialog accessible from the **Options** heading on the menu toolbar, you will enter **EZ Mode** for initial setup. From there, you can enter **Application Mode (App Mode)** and access several configuration menus (**Communications**, **Read Cycle**, **Symbolologies**, **I/O Parameters**, **Symbol Quality**, **Matchcode**, and **Diagnostics**), a **Camera** setup interface, a **Terminal** interface, a **Utilities** interface, and an **Output Format** interface.

**ESP** can be used to configure the MicroHAWK ID in three different ways:

- **Tree Controls:** Each configuration menu contains a list of all option settings that pertain to that specific element of reader operation. For example, the **Communications** menu shows a **Host Port Connections** option, and then a list of the sub-options **Baud Rate**, **Parity**, **Stop Bits**, and **Data Bits**. Each of these sub-options is configurable by using dropdown menus.
- **Graphic User Interfaces:** Reader settings can be configured using such point-and-click tools as radio buttons, zoom in/zoom out sliders, spin boxes, check boxes, and drag-and-drop functions.
- **Terminal:** **ESP's Terminal** allows you to send serial configuration and utility commands directly to the reader by typing them in the provided text field.

Information about using **ESP** in specific applications is provided in subsequent sections.

For **ESP** system requirements, see **Minimum System Requirements** in **Quick Start**.

## EZ Mode

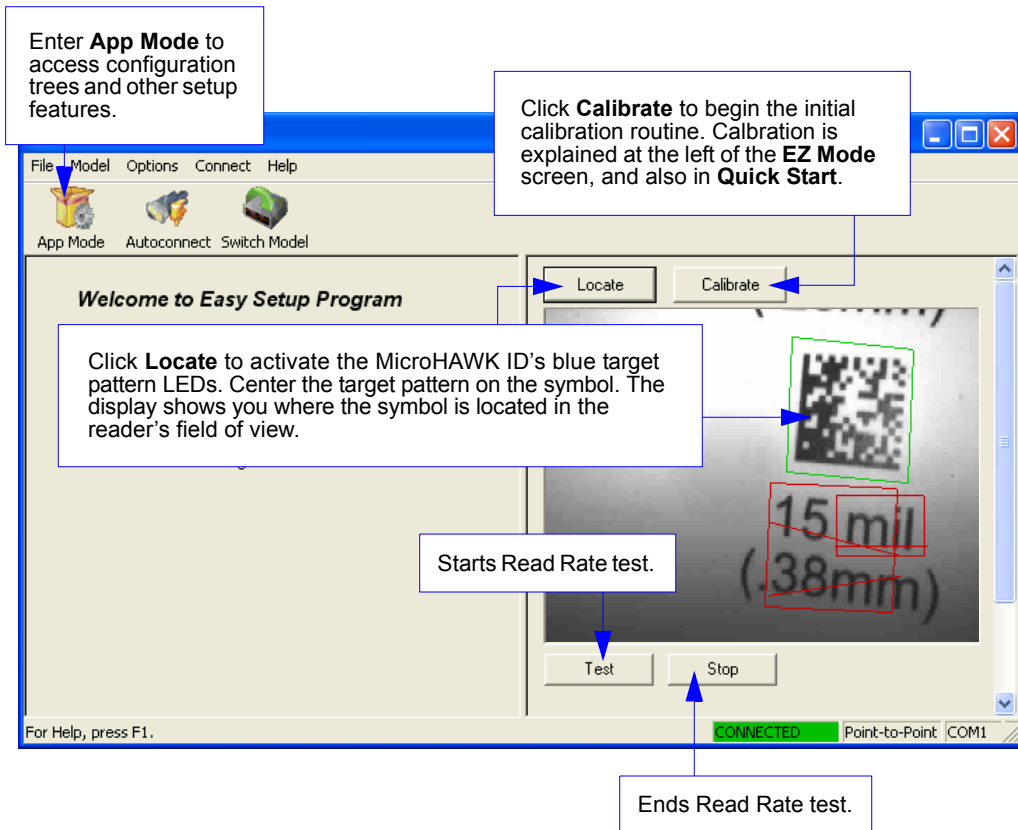
In **EZ Mode** you are presented with the **Locate**, **Calibrate**, and **Test** options. After connecting to your reader, **EZ Mode** is the screen you will see. You will be provided with on-screen instructions that will help you with positioning, testing, and calibration.

### Test

Click the **Test** button to start the Read Rate test for a quick indication of the reader's read capabilities and the limits of your application. When **Decodes per Second** is unchecked, the test will count the percentage of decodes relative to the number of actual scans. Click **Stop** to end the test.

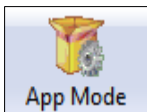
### Calibrate

The calibration routine that will optimize the reader by comparing Read Rates at various camera and image processing settings.

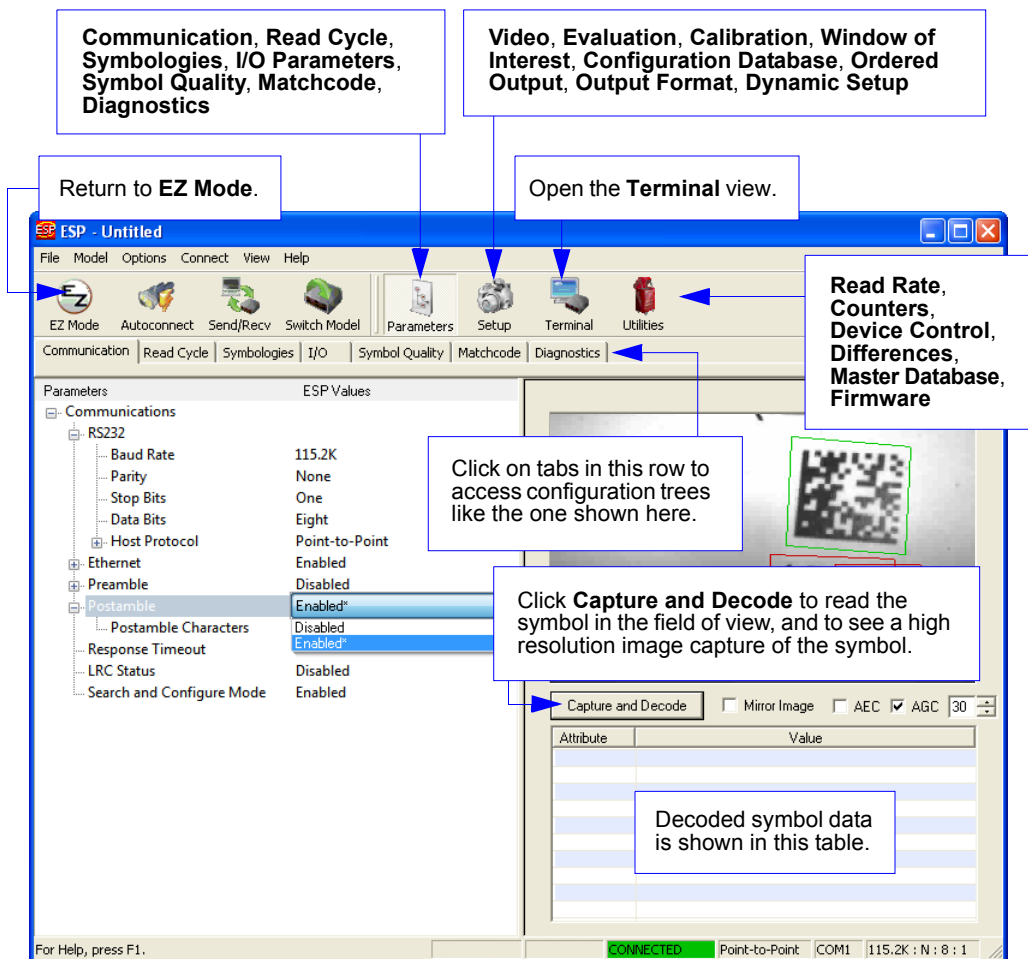


# Application Mode

From **EZ Mode**, you can click on the **App Mode** button to access specific configuration menus, **Utilities** tools, **Camera** setup, **Output Format** options, and a **Terminal** window where serial commands can be entered.



**Note:** The **App Mode** and **EZ Mode** buttons appear in the same position to allow easy switching between these primary modes.



**Note:** For specific information on any of the icons shown above in the operations bar or configuration bar, see the corresponding sections of this manual.

# Menu Toolbar

## File > New

Whenever **New** is selected, the default configuration of **ESP** is loaded.

## Open/Save

When **Save** or **Save As** is selected, the **ESP** configuration is saved to the host computer's hard drive and available whenever the same file is selected under **Open**.

**Important:** When you save menu changes to your hard drive, these changes are not saved to your reader. The illustration below shows how settings can be saved and received between **ESP** and the reader, and **ESP** and the host hard drive.

File	
New	Ctrl+N
Open...	Ctrl+O
Save	Ctrl+S
Save As...	
Print...	Ctrl+P
Import...	
Export...	
Recent File	
Exit	

## Import/Export

**Import** converts the ASCII settings from a text file to **ESP** configuration settings.

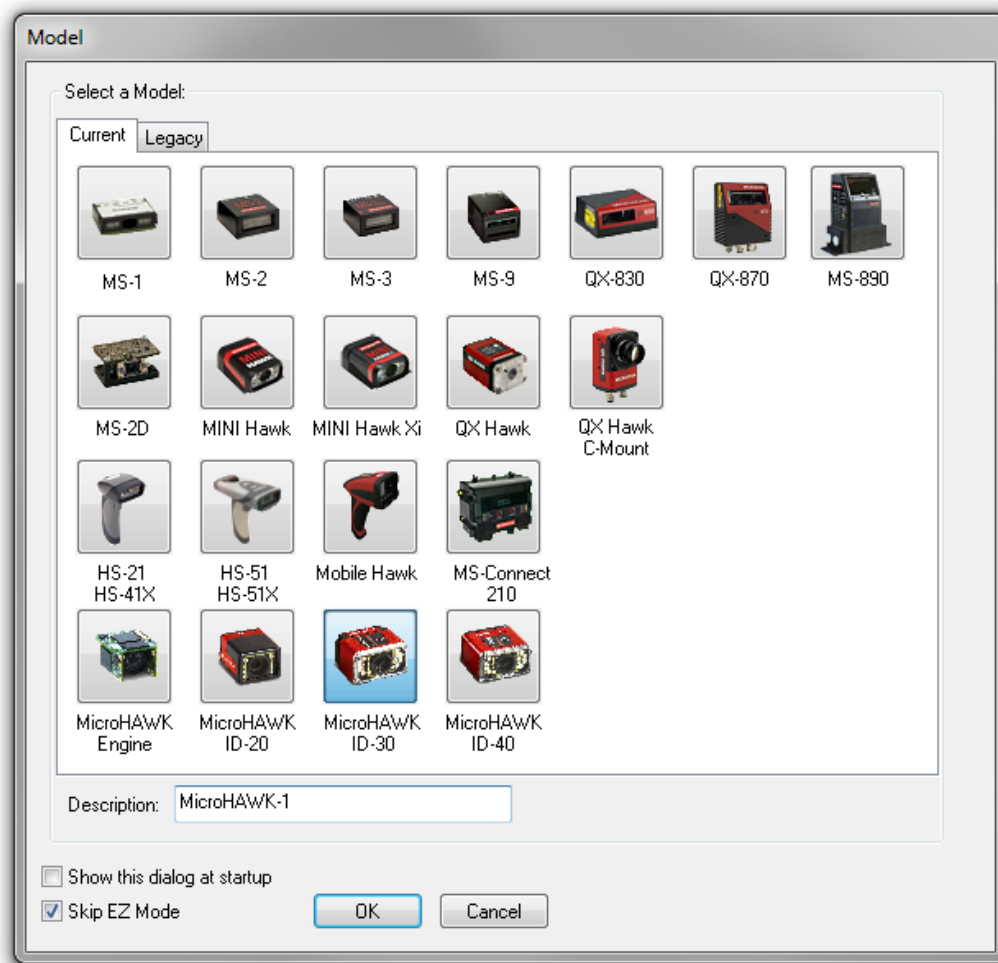
**Export** converts the active **ESP** configuration settings to an ASCII text file.



## Model

In the **Model** menu you can select any of the models supported by **ESP**. When you choose a different model, the connection to your present model will be terminated.

To connect to another model, select **New Model**, choose a new model from the **pop-up menu** that appears, and click **OK**.

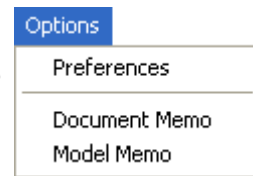


**Note:** When you save an **ESP** file, you are saving the settings of all the models defined in that file.

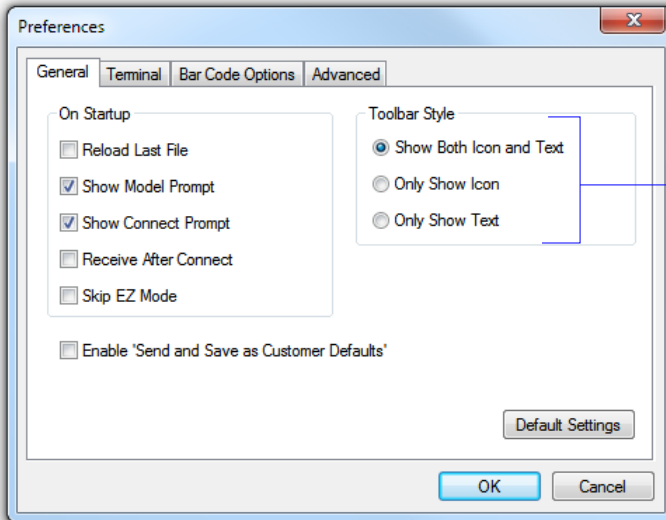
## Options

The **Options** menu allows you to save memos and set up **ESP Preferences**.

**Note:** Preferences will be saved and loaded into **ESP** whenever **ESP** is opened next, whether or not you save the **ESP** file.



### Preferences > General Tab



The **Toolbar Style** options allow you to determine how **ESP** will display the mode options in the two rows at the top of the screen.

#### ***Reload Last File***

At startup, reloads the last file saved to the host computer's hard drive.

#### ***Show Model Prompt***

At startup, shows the model menu displaying all supported readers.

#### ***Show Connect Prompt***

At startup, displays the **Would you like to connect to the MicroHAWK ID?** prompt.

#### ***Receive After Connect***

At startup, loads the reader's settings into **ESP**. (This is not recommended if you want to preserve your **ESP** settings for future use.)

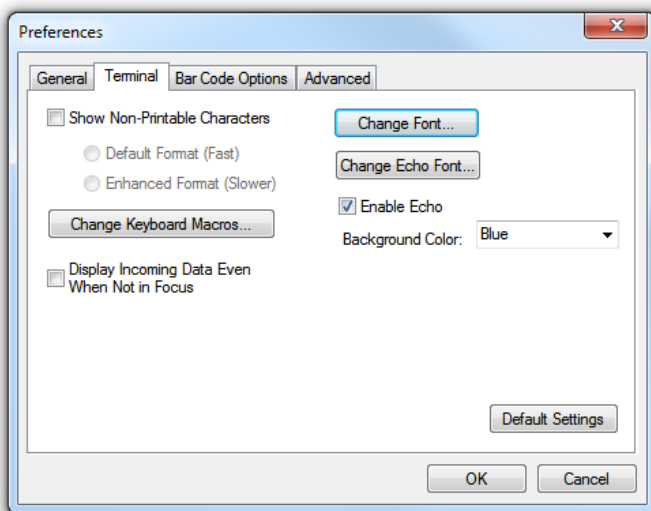
#### ***Skip EZ Mode***

At startup, skips **EZ Mode** and opens directly in **App Mode**.

#### ***Enable 'Send and Save as Customer Defaults'***

At startup, enables the **Send and Save as Customer Defaults** option in the **Send/Recv** command.

## Preferences > Terminal Tab



### Show Non-Printable Characters

When **Show Non-Printable Characters** is enabled, characters such as “CRLF” will be displayed in the **Terminal** window. When **Enhanced Format** is checked, the characters are displayed with more detailed formatting.

### Change Keyboard Macros

Clicking the **Change Keyboard Macros** button brings up the **Function Keys** dialog. In this dialog you can select the desired function key and then enter your macro keystrokes in the associated key map. For example, to make **Ctrl-F2** the keystroke to send a trigger character, select **F2**, then in the **Ctrl** row, enter **<trigger character>** and click **OK**. Then whenever the **Ctrl-F2** keystroke is pressed, the trigger character will start the read cycle.

**Note:** The **F1** key is reserved for opening **ESP** Help and the **F3** key is reserved for the **Find Next** function.

### Change Font

Allows you to modify the font used for decode data received from the reader on the **Terminal** screen.

### Change Echo Font

Allows you to modify the font used for command characters typed into the **Terminal** view.

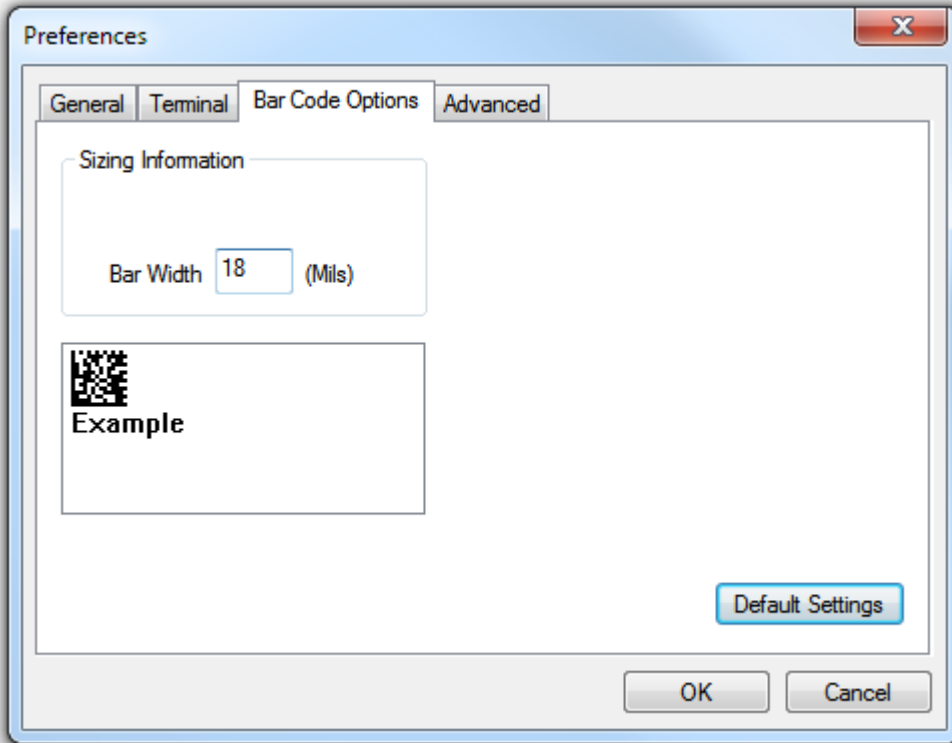
### Enable Echo

Allows you to enter command characters in **Terminal**.

### Display Incoming Data Even When Not in Focus

When **Display Incoming Data Even When Not in Focus** is enabled, data from the reader will continue to appear in the terminal even when **ESP** is not the top window.

## Preferences > Bar Code Options Tab



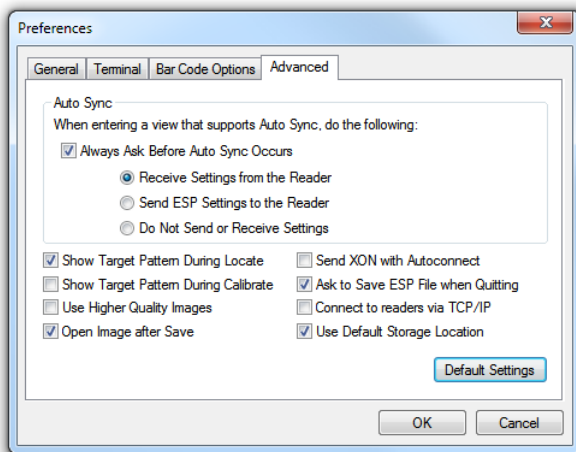
The **Bar Code Options** dialog allows you to set the size of user-created symbols.

### ***Sizing Information***

Sets the bar width or module width (in **mils**, or thousandths of an inch) of user-created symbols.

*Example:* A bar width of 18 is 0.018 inches.

## Preferences > Advanced Tab



The **Auto Sync** options at the top of the **Advanced** tab allow you to determine whether Auto Sync will be enabled automatically in sections of **ESP** where it is used, or if it will ask you before it enables Auto Sync functions.

### ***Always Ask Before Auto Sync Occurs***

If you check this option box, you are then able to determine what specific Auto Sync functions will be enabled. **Receive Settings from the Reader** will automatically send the reader's settings to **ESP** when Auto Sync is enabled. **Send ESP Settings to the Reader** will automatically send all reader configuration settings chosen in **ESP** to the reader. **Do Not Send or Receive Settings** creates a condition in which Auto Sync will not automatically send reader settings to **ESP**, or send **ESP** settings to the reader.

### ***Show Target Pattern During Locate***

Allows you to determine whether the blue LED target pattern will be on or off during the **Locate** routine.

### ***Show Target Pattern During Calibrate***

Allows you to determine whether the blue LED target pattern will be on or off during the **Calibrate** routine.

### ***Use Higher Quality Images***

Sets **ESP** to output images at a higher resolution than standard JPEG format.

### ***Open Image after Save***

When **Open Image after Save** is enabled, **ESP** automatically opens saved image captures. Images can be saved from the **Evaluation** tab in the **Camera** view, or by right-clicking an image in any other image capture view and then saving.

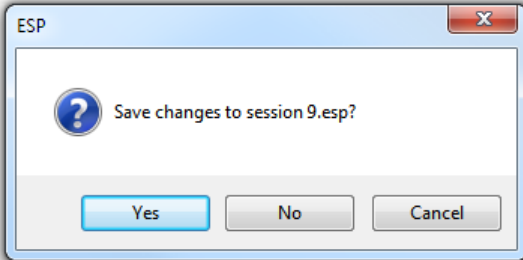
### ***Send XON with Autoconnect***

Sends an **XON (Begin Transmission)** command to the reader before starting **Autoconnect**.

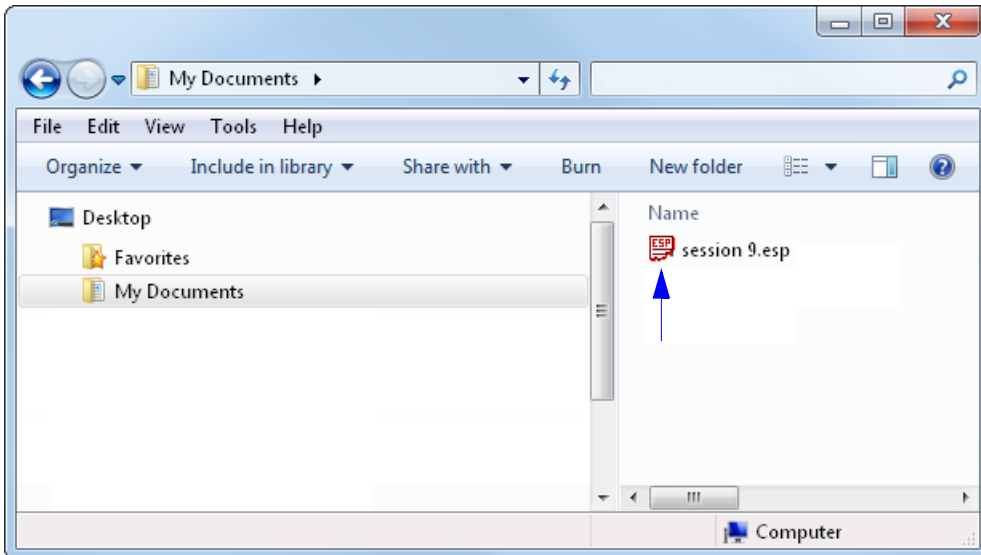
## Preferences > Advanced Tab (cont.)

### **Ask to Save ESP File when Quitting**

When enabled, prompts the user to save a **.esp** file when ending a session.



The **.esp** file will be saved in the location of your choice.



### **Connect to Readers via TCP/IP**

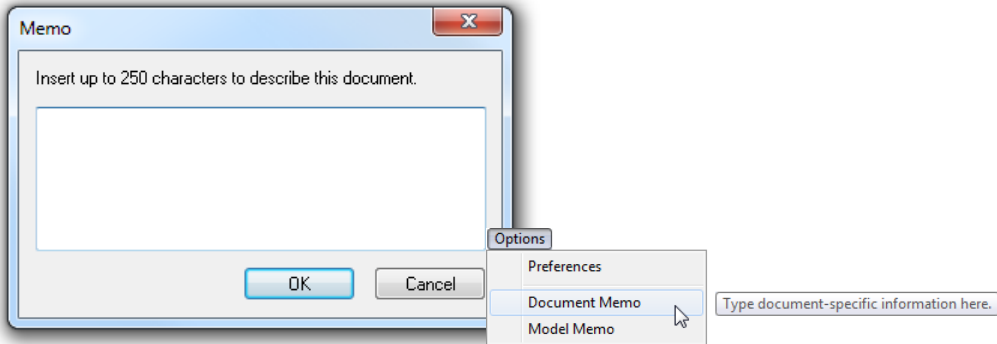
When enabled, shows a **TCP/IP** option on the **Connection Wizard**.

### **Use Default Storage Location**

When enabled, automatically stores data in **ESP's** Application Data folder.

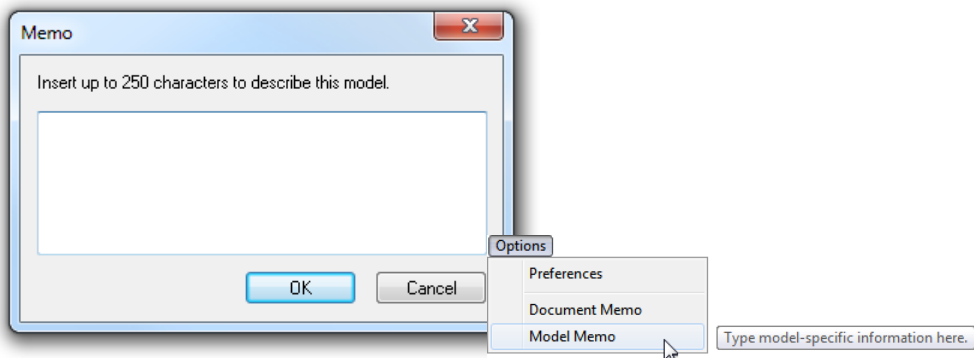
## Document Memo

The information you type in the **Document Memo** field will appear in a context-sensitive text box whenever your cursor hovers over the **Document Memo** item on the **Options** menu.



## Model Memo

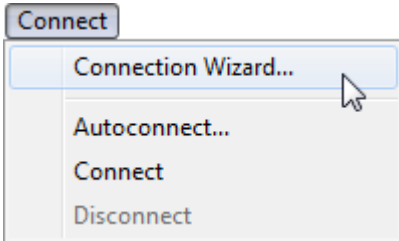
Similar to **Document Memo**, the information you type in the **Model Memo** field will appear in a context-sensitive text box whenever your cursor hovers over the **Model Memo** item on the **Options** menu. Memos created in **Model Memo** are specific to the model enabled when the message was created.



**Note:** Memos must be saved in a **.esp** file if you want them to be available in your next session. If you do not save your current session, any memos that you have entered during the session will be discarded, and will be unavailable in your next session.

## Connect

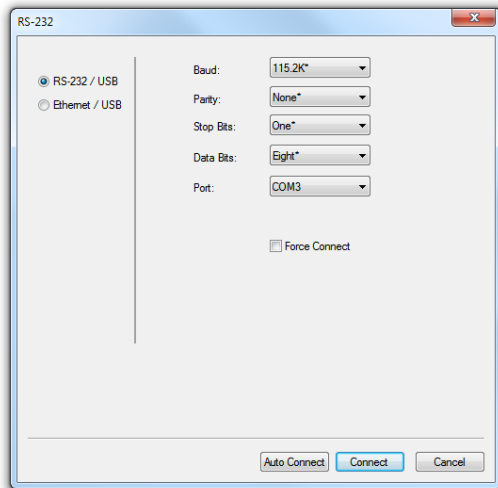
The **Connect** dropdown menu allows the user to access the **Connection Wizard**, as well as the **Autoconnect** dialog. **Connect** and **Disconnect** can also be performed directly from the dropdown menu without opening a dialog.



## Connection Wizard

To connect using the **Connection Wizard**:

- Click **Connect** on **ESP**'s menu toolbar, and then select **Connection Wizard**.
- Select the communication interface required by your application.
- Configure settings as required by the application, and click **Connect**.



- When a connection is established, the green indicator in the status bar at the bottom right of the screen will be visible.



- If your RS-232 connection attempt fails, click the **Auto Connect** button to establish a connection between the MicroHAWK ID and the host.

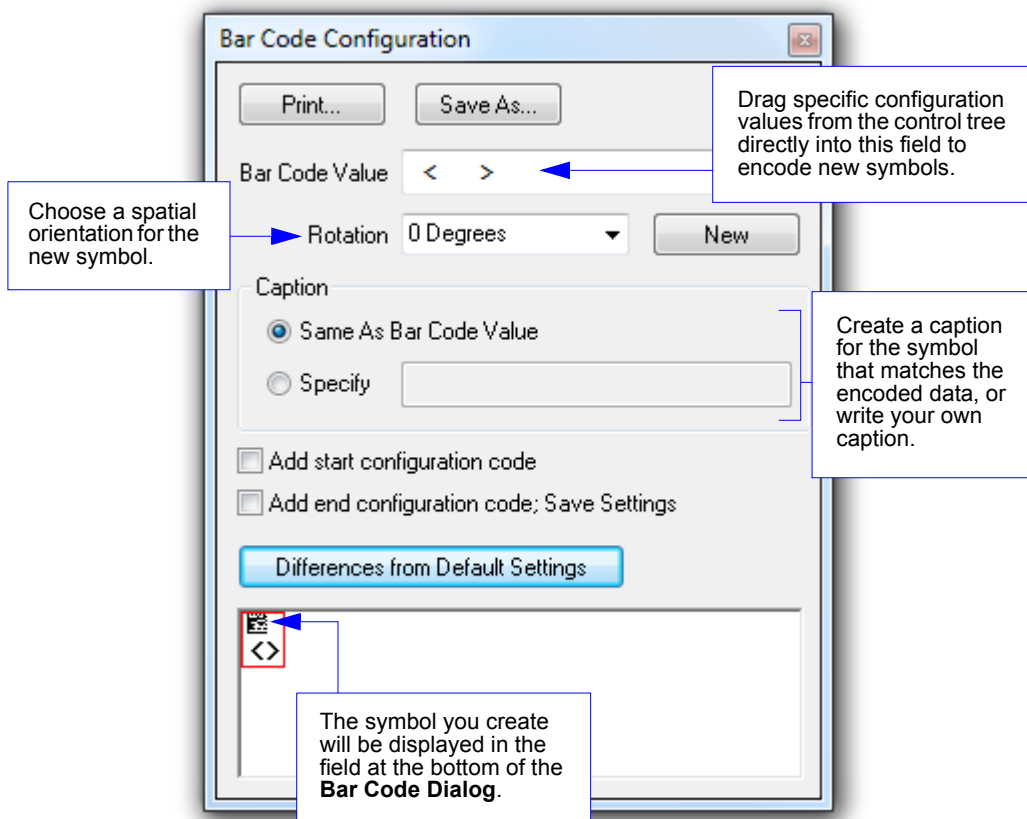


## View

The **View** menu allows you to move quickly between interfaces without using the icon buttons on the **App Mode** toolbar. It also allows you to access the **Bar Code Dialog**.

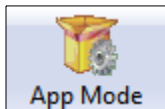
### Bar Code Dialog

In the **Bar Code Dialog** you can create symbols by typing the text you wish to encode. This is a useful tool for creating configuration symbols, allowing you to configure your reader by reading the symbols you create.

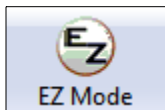


# Navigating in ESP

To change reader settings, or to access the **Utilities**, **Camera**, **Terminal**, or **Output Format** views, click the **App Mode** button.

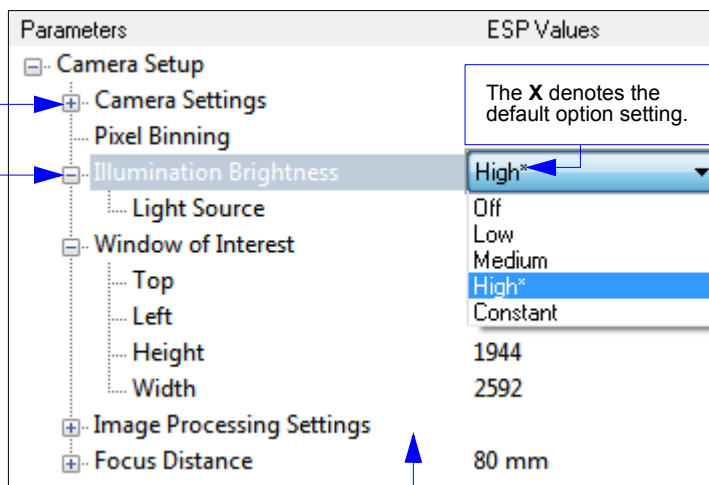


To return to **EZ Mode**, click the **EZ Mode** button.

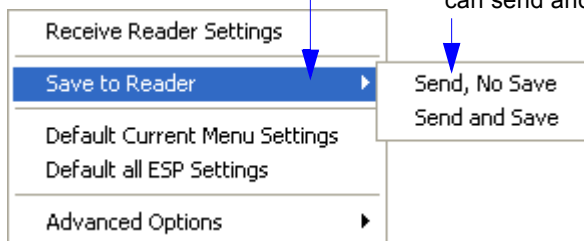


**To make changes to configuration settings in the control trees:**

1. **Left-click** on the + to expand menu items.
2. **Double-click** the desired parameter and **single-click** in the selection box to view options.
3. Place your cursor in the selection box, scroll down to the setting you want to change, and **single-click** the setting.



4. **Left-click** again on the open screen to complete the selection.
5. **Right-click** on the open screen and select **Save to Reader** to implement the command in the reader. You can send the command without saving it, or you can send and save the command simultaneously.



## Send/Receive Options

To access **Receive**, **Save**, and **Default** options, click the **Send/Recv** button. You can also access these options by right-clicking in any of the configuration views.



### Receiving

From the **Send/Recv** menu, select **Receive Reader Settings**.

**Caution:** Do not select this option if you do not want to upload the reader's settings. For example, if your **ESP** file has a number of custom settings that you want to maintain and download into the reader, these settings would be lost by choosing **Yes**.

This is useful if you want to receive (upload) the reader's settings and save them as a file for future use. For example, if your reader has settings that you do not want to change, choosing **Yes** would allow you to load those settings to **ESP** and save them in an **ESP** file for later retrieval.

Receiving the reader's settings will also assure that you will not be subsequently saving any unwanted changes that you or someone else has made previously in **ESP**.

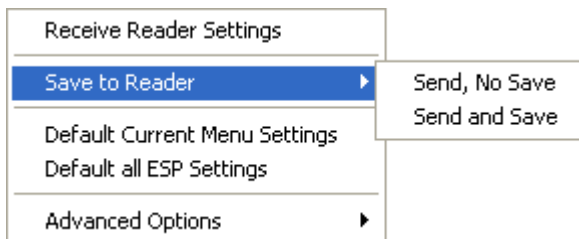
### Saving

#### Send, No Save (<A>)

Saves **ESP** settings to current memory.

#### Send and Save (<Z>)

Activates all changes in current memory *and* saves to the reader for power-on.



#### Send and Save as Customer Defaults (<Zc>)

Saves your default settings for quick retrieval.

This option will be visible only if you have checked **Enable 'Send and Save as Customer Defaults'** in **ESP Preferences**.

# Defaulting

When you select **Default Current Menu Settings** or **Default all ESP Settings**, you are only defaulting the **ESP** settings.

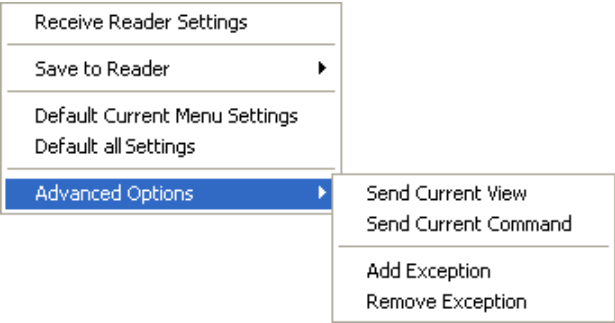
# Advanced Options

## Send Current View

This is the same as **Save to Reader** > **Send No Save** except that only the commands in the current configuration tree are sent.

## Send Current Command

This is the same as **Send Current View**, except that it saves only the command that is currently selected.



## Add/Remove Exception

After you perform a **Receive Reader Settings** command<sup>1</sup> and you click on the **Add Exception** option, you may see a list of serial commands. These are commands that may be in your reader's firmware, but not included in, or different from, your current version of **ESP**.

You can edit these commands by double-clicking on them and changing them as needed.

It is important to note that these commands will be saved to your reader whenever you send a **Save to Reader** command, or an **<A>** or a **<Z>** command.

Also, if there is a corresponding **ESP** menu item, the **ESP Value** column for that item will be blank following a **Receive Reader Settings** command.

1. From the **Send/Recv** button or by right-clicking in any blank section of a tree control view.



# 3 *Communications*

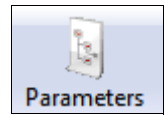
## **Contents**

Communications by ESP .....	3-2
Communications Serial Commands.....	3-3
Host Port Connections .....	3-4
Host Port Protocol.....	3-5
ACK/NAK Options.....	3-6
Polling Mode Options.....	3-7
Ethernet .....	3-8
Response Timeout.....	3-14
LRC Status.....	3-15
Protocol Configuration Examples.....	3-16
ASCII Character Entry Modifier.....	3-17
Preamble.....	3-18
Postamble .....	3-19

This section explains how to set up communications parameters with the host.

With Microscan's **ESP** (Easy Setup Program), configuration changes can be made in the **ESP** menus and then sent and saved to the reader. The user can also send serial commands to the reader via **ESP's Terminal** window.

# Communications by ESP



Parameters

Click the **Parameters** button and then the **Communication** tab.

Parameters	ESP Values
[-] Communications	
[-] RS232	
Baud Rate	115.2K
Parity	None
Stop Bits	One
Data Bits	Eight
+ Host Protocol	Point-to-Point
+ Ethernet	Enabled
+ Preamble	Disabled
[-] Postamble	Enabled*
Postamble Characters	Disabled
Response Timeout	Enabled*
LRC Status	Disabled
Search and Configure Mode	Enabled

To open nested options, **single-click** the +.

To change a setting, **double-click** the setting and use your cursor to scroll through the options.

# Communications Serial Commands

Host Port Connections	<K100,baud rate,parity,stop bits,data bits>
Ethernet	<K126,status,IP address,subnet,gateway,IP address mode>
Ethernet TCP Ports	<K127,TCP Port 1,TCP Port 2>
Search and Configure Mode	<K128,status,timed window>
EtherNet/IP	<K129,status>
Host Protocol	<K140,protocol,address>
Preamble	<K141,status,preamble characters>
Postamble	<K142,status,postamble characters>
Response Timeout	<K143,response timeout>
LRC Status	<K145,status>
ACK/NAK Options	<K147,RES,REQ,STX,ETX,ACK,NAK>
Polling Mode Options	<K148,RES,REQ,STX,ETX,ACK,NAK>

The following settings define the basic transmission speeds and digital standards that ensure common formatting.

## Baud Rate, Host Port

*Options:*      0 = 600                      1 = 1200                      2 = 2400  
                     3 = 4800                      4 = 9600                      5 = 19.2K  
                     6 = 38.4K                      7 = 57.6K                      **8 = 115.2K**  
                     9 = 230K

## Parity, Host Port

*Options:*      **0 = None**                      1 = Even                      2 = Odd

## Stop Bits, Host Port

*Options:*      **0 = One**                      **1 = Two**

### ***Data Bits, Host Port***

*Options:*      0 = Seven                      **1 = Eight**



# Host Port Protocol

- Usage:* In general, the point-to-point protocols will work well in most applications. They require no address and must use RS-232 communications standards.
- Definition:* Protocols define the sequence and format in which information is transferred between the reader and the host.
- Serial Cmd:* <K140,protocol,address>
- Default:* **Point-to-Point**
- Options:* **0 = Point-to-Point**  
 4 = ACK/NAK  
 5 = Polling Mode

**Note:** In all protocol modes, the preamble <K141> and postamble <K142> character strings can be used to frame the decode data, and both are included in calculating the **LRC** (Longitudinal Redundancy Check).

## Point-to-Point (Standard)

- Usage:* Used only with RS-232.
- Definition:* Standard **Point-to-Point** requires no address and sends the data to the host whenever it is available, without a request or handshake from the host.
- Serial Cmd:* <K140,0>

## ACK/NAK

- Definition:* See the **ACK/NAK Options** command <K147>.
- Serial Cmd:* <K140,4>

## Polling Mode

- Definition:* See the **Polling Mode Options** command <K148>.
- Serial Cmd:* <K140,5>

## Poll Address

- Serial Cmd:* <K140,protocol,address>
- Default:* 1
- Options:* 1 to 50  
 1 = Poll address 0x1C, Select address 0x1D  
 2 = Poll address 0x1E, Select address 0x1F  
 ...  
 50 = Poll address 0x7E, Select address 0x7F

## ACK/NAK Options

**Definition:** These parameters take effect for **ACK/NAK <K140,4>** on the main RS-232 port and are completely independent of the **Polling Mode Options <K148>**. The reader always follows the protocol in both directions (to and from the host). There is no option to disable it from either direction.

**Serial Cmd:** <K147,RES,REQ,STX,ETX,ACK,NAK>

### RES-NAK Defaults

**RES: (Reset)** 00 (disabled)

**REQ: (Request)** 00 (disabled)

**STX: (Start of Text)** 00 (disabled)

**ETX: (End of Text)** 00 (disabled)

**ACK: (Acknowledge)** 06

**NAK: (Negative Acknowledge)** 15

The following are general outlines of the **ACK/NAK** protocol. Items that are framed by brackets ( [ ] ) can either be disabled or enabled. LRC does not include STX, but it does include preamble, postamble, and ETX.

### Symbol Data Output

**TX to host:** [STX] [preamble] SYMBOL DATA [postamble] [ETX] [LRC]

**Response from host:** ACK/NAK. Sent when LRC, ETX, postamble, or timeout (waiting for more data) are detected (if REQ is disabled) depending on what is enabled.

### Commands from Host to Reader

**TX to Reader:** [STX] <command> [ETX] [LRC]

**Response from Reader:** ACK/NAK. Sent when LRC, ETX, or command-ending angle bracket '>' are received, depending on what is enabled.

### Command Response from Reader to Host

**TX to host:** [STX] [preamble] COMMAND RESPONSE DATA [postamble] [ETX] [LRC]

**Response from host:** ACK/NAK. Sent when LRC, ETX, postamble, command-ending angle bracket '>', or timeout (waiting for more data) are detected, depending on what is enabled.

As with **Polling Mode <K140,5>**, the reader can optionally perform the REQ and RES event sequences in ACK/NAK mode. If the sender does not receive an ACK or NAK, it will send REQ to request such a response (if enabled). When the sender receives an ACK, too many NAKs, or times out (if already enabled), it will send a RES (if enabled) to terminate the transaction.

**Note:** See **ACK/NAK Data Flow Examples** for sample ACK/NAK communication scenarios.

# Polling Mode Options

**Definition:** These parameters only take effect for **Polling Mode <K140,5>** on the main RS-232 port and are completely independent of the **ACK/NAK Options <K147>**.

The values of protocol characters can be changed, but the protocol events cannot be disabled. The polling mode address is configured in the **<K140>** command (see **Poll Address**).

If RS-232 is enabled, **<K102,0>**, **Polling Mode** will operate as a **Point-to-Point** polling protocol. This is because the RS-232 transmitter is always left on when enabled.

**Serial Cmd:** **<K148,RES,REQ,STX,ETX,ACK,NAK>**

## RES-NAK Defaults

<i>RES: (Reset)</i>	04
<i>REQ: (Request)</i>	05
<i>STX: (Start of Text)</i>	02
<i>ETX: (End of Text)</i>	03
<i>ACK: (Acknowledge)</i>	06
<i>NAK: (Negative Acknowledge)</i>	15

**Note:** See **Polling Mode Data Flow Examples** for sample **Polling Mode** communication scenarios.

# Ethernet

Enables or disables Ethernet connectivity in the reader. This corresponds to the <K126> command. It requires a <Zrdall> to return to default settings.

Ethernet	Enabled
IP Address	192.168.0.100
Subnet	255.255.0.0
Gateway	0.0.0.0
IP Address Mode	Static
TCP Port 1	2001
Symbol Data Output	Enabled
Extra Symbol Information	Enabled
Diagnostics Output	Enabled
External Source Processing Mode	Command
TCP Port 2	2003
Symbol Data Output	Enabled
Extra Symbol Information	Enabled
Diagnostics Output	Enabled
External Source Processing Mode	Command

## IP Address

This is the IP address of the reader when it is in Static IP Address Mode.

Ethernet	Enabled
IP Address	192.168.0.100
Subnet	255.255.0.0
Gateway	0.0.0.0
IP Address Mode	Static
TCP Port 1	2001
Symbol Data Output	Enabled
Extra Symbol Information	Enabled
Diagnostics Output	Enabled
External Source Processing Mode	Command
TCP Port 2	2003
Symbol Data Output	Enabled
Extra Symbol Information	Enabled
Diagnostics Output	Enabled
External Source Processing Mode	Command

## Subnet

This is the subnet of the reader when it is in Static IP Address Mode.

[-] Ethernet	Enabled
IP Address	192.168.0.100
Subnet	255.255.0.0
Gateway	0.0.0.0
IP Address Mode	Static
[-] TCP Port 1	2001
Symbol Data Output	Enabled
Extra Symbol Information	Enabled
Diagnostics Output	Enabled
External Source Processing Mode	Command
[-] TCP Port 2	2003
Symbol Data Output	Enabled
Extra Symbol Information	Enabled
Diagnostics Output	Enabled
External Source Processing Mode	Command

## Gateway

This is the gateway IP address of the reader when it is in Static IP Address Mode.

[-] Ethernet	Enabled
IP Address	192.168.0.100
Subnet	255.255.0.0
Gateway	0.0.0.0
IP Address Mode	Static
[-] TCP Port 1	2001
Symbol Data Output	Enabled
Extra Symbol Information	Enabled
Diagnostics Output	Enabled
External Source Processing Mode	Command
[-] TCP Port 2	2003
Symbol Data Output	Enabled
Extra Symbol Information	Enabled
Diagnostics Output	Enabled
External Source Processing Mode	Command

# IP Address Mode

Determines how the reader’s IP address will be defined.

[-] Ethernet	Enabled
IP Address	192.168.0.100
Subnet	255.255.0.0
Gateway	0.0.0.0
IP Address Mode	Static*
[-] TCP Port 1	Static*
Symbol Data Output	DHCP
Extra Symbol Information	Enabled
Diagnostics Output	Enabled
External Source Processing Mode	Command
[-] TCP Port 2	2003
Symbol Data Output	Enabled
Extra Symbol Information	Enabled
Diagnostics Output	Enabled
External Source Processing Mode	Command

## Static

In **Static Mode**, the reader uses the user-defined IP address entered in ESP. This is the default state for an Ethernet-enabled MicroHAWK.

## DHCP

In **DHCP Mode**, the reader automatically acquires the IP address, Subnet, and Gateway addresses from a DHCP or BOOTP server. This is the default state for a USB MicroHAWK.

## TCP Port 1

One of two TCP ports for Ethernet communication with the reader. The default setting is **2001**.

<input checked="" type="checkbox"/> Ethernet	Enabled
IP Address	192.168.0.100
Subnet	255.255.0.0
Gateway	0.0.0.0
IP Address Mode	Static
<input checked="" type="checkbox"/> TCP Port 1	2001 <span style="border: 1px solid black; padding: 2px;">1024 - 65535</span>
Symbol Data Output	Enabled
Extra Symbol Information	Enabled
Diagnostics Output	Enabled
External Source Processing Mode	Command
<input checked="" type="checkbox"/> TCP Port 2	2003
Symbol Data Output	Enabled
Extra Symbol Information	Enabled
Diagnostics Output	Enabled
External Source Processing Mode	Command

### Symbol Data Output

Enables or disables decoded symbol data output from the reader.

### Extra Symbol Information

Enables or disables extra symbol information output from the reader.

### Diagnostics Output

Enables or disables diagnostics output from the reader.

### External Source Processing Mode

Enables or disables processing of commands or data from sources external to the reader.

#### **Command**

**Command** enables command processing in the reader.

#### **Data**

**Data** enables Ethernet TCP Port 1 as a data source port.

**Note:** The data path between **in** the source port and **out** the source port is always two-way. Data is copied from source data ports and all those source ports' data is transmitted to the destination port, and from the destination port to the source port.

# TCP Port 2

One of two TCP ports for Ethernet communication with the reader. The default setting is **2003**.

[-] Ethernet	Enabled
IP Address	192.168.0.100
Subnet	255.255.0.0
Gateway	0.0.0.0
IP Address Mode	Static
[-] TCP Port 1	2001
Symbol Data Output	Enabled
Extra Symbol Information	Enabled
Diagnostics Output	Enabled
External Source Processing Mode	Command
[-] TCP Port 2	2003 1024 - 65535
Symbol Data Output	Enabled
Extra Symbol Information	Enabled
Diagnostics Output	Enabled
External Source Processing Mode	Command

## Symbol Data Output

Enables or disables decoded symbol data output from the reader.

## Extra Symbol Information

Enables or disables extra symbol information output from the reader.

## Diagnostics Output

Enables or disables diagnostics output from the reader.

## External Source Processing Mode

Enables or disables processing of commands or data from sources external to the reader.

### Command

**Command** enables command processing in the reader.

### Data

**Data** enables Ethernet TCP Port 2 as a data source port.

**Note:** The data path between **in** the source port and **out** the source port is always two-way. Data is copied from source data ports and all those source ports' data is transmitted to the destination port, and from the destination port to the source port.



## Search and Configure Mode

Search and Configure Mode

Enabled\*

Disabled

Enabled\*

Timed Window

**Search and Configure Mode** is intended primarily for initial setup of an reader in a network. This parameter controls whether or not the reader will respond to ESP's **Search** function in the Ethernet TCP/IP connect dialog section of the **Connection Wizard**:

**TCP/IP**

☐ RS-232

☒ Ethernet

IP Address: 10 . 10 . 4 . 46

TCP Port 1: 2001 ☒ Connect to Port 1

TCP Port 2: 2003 ☐ Connect to Port 2

Subnet: 255 . 255 . 254 . 0

Gateway: 10 . 10 . 4 . 1

IP Address Mode: ☒ DHCP ☐ Static

Name	IP Address	MAC ID
	10.10.4.132	00:0B:43:0F:6B:6B

After the reader is in full use in an application, you may want the reader to continue to appear in the **Search** response list for future searches. This can help prevent disruption of the system that may occur is ESP were to show only unconfigured units.

Changes to this parameter are saved to NOVRAM and are set to default on power on.

A Reset **<A>** is required for settings to take effect.

**Important:** Once this setting is Disabled, ESP will only be able to connect to the reader if you know the IP address and enter it in the IP Address field of the Ethernet TCP/IP connect dialog. Search and Configure Mode can be re-enabled by defaulting the reader.

### Enabled

When enabled, **Search and Configure Mode** will find the reader and settings can be changed.

### Timed Window

When **Timed Window** is selected, Search and Configure Mode will find the reader and settings can be changed, but only 60 seconds from the last reset. After 60 seconds, Search and Configure Mode will be disabled.

## Response Timeout

*Usage:* Only used when a response is required from the host. The reader can be set to wait indefinitely by setting **Response Timeout** to zero.

*Definition:* The time that the reader will wait before timing out if **ACK**, **NAK**, and **ETX** are enabled, and a host response is expected.

*Serial Cmd:* **<K143,response timeout>**

*Default:* **12** (in 1 ms increments)

*Options:* 0 to 255 (A zero (0) setting causes an indefinite wait.)

# LRC Status

## *(Longitudinal Redundancy Check)*

**Usage:** Used when extra data integrity is required.

**Definition:** An error-checking routine that verifies the accuracy of transmissions. It is the exclusive OR of all characters following the **STX** (start of text) up to and including the **ETX** (end of text). What this means is that the binary representation of all the characters in a transmission are cumulatively added in a column and each resulting odd integer is assigned a 1 and each even integer a 0 (two 1s = 0, two 0s = 0, a 1 and a 0 = 1). The extra LRC character is then appended to the transmission, and the receiver (usually the host) performs the same addition and compares the results.

**Serial Cmd:** <**K145,status**>

**Default:** **Disabled**

**Options:** **0 = Disabled**                      1 = Enabled

## Protocol Configuration Examples

### ***Point-to-Point (Main Port)***

<K100,8,0,1,1> Baud Rate: **115.2K**; Parity: **None**; Stop Bits: **2**; Data Bits: **8**  
<K140,0> Point-to-Point  
<K102,0> RS-232 enabled

### ***Polling Mode (Main Port)***

<K100,4,0,1,1> Baud Rate: **9600**; Parity: **None**; Stop Bits: **2**; Data Bits: **8**  
<K140,5,23> Polling Mode; Address: **23**  
<K102,0> RS-232 Point-to-Point polling  
<K143,30> 30 ms Response Timeout

### ***“User-Defined” Polling Mode (Main Port)***

<K100,4,0,1,1> Baud Rate: **9600**; Parity: **None**; Stop Bits: **2**; Data Bits: **8**  
<K140,5,12> Polling Mode; Address: **12**  
<K148,,08,09,18,0B,0C,0D> Default RES (0x04), REQ=0x08; EOT=0x09; STX=0x18;  
ETX=0x0B; ACK=0x0C; NAK=0x0D  
<K102,0> RS-232 Point-to-Point polling  
<K143,40> 40 ms Response Timeout

### ***ACK/NAK (Main Port)***

<K100,9,0,1,1> Baud Rate: **230K**; Parity: **None**; Stop Bits: **2**; Data Bits: **8**  
<K140,4> ACK/NAK  
<K147,,,01,1B,2E,1F> Default RES and REQ (00, disabled); STX=0x01; ETX=0x1B;  
ACK=0x2E; NAK=0x1F  
<K102,0> RS-232 enabled  
<K143,50> 50 ms Response Timeout

# ASCII Character Entry Modifier

Commands that require ASCII text fields, such as **Preamble** and **Postamble** commands, can be sent to the reader as hex pairs.

*Serial Cmd Format:* <**Knnnh,00-FF**>

To enter ASCII fields as hex values (00 to FF), add a lower-case **h** directly after the command's **K** number, and then enter the hex value that corresponds with the desired ASCII character.

*Example:*

Consider the **Postamble** command:

*Serial Cmd:* <**K142,status,postamble character(s)**>

Imagine that your application requires the ASCII character > to be the postamble in your symbol decode output.

The ASCII characters <, >, and , can only be entered as hex pairs. So, to make > the postamble in your symbol decode output, enter the **Postamble** command as follows:

**<K142h,,3E>**

Note that the “status” field contains only a , . This is because the only field that is being changed is the “postamble character(s)” field. (See **Serial Configuration Command Conventions** for a more detailed explanation of this command shortcut.)

# Preamble

## ***Preamble Status***

- Usage:** Useful for identifying and controlling incoming data. For example, defining the preamble as a carriage return and a line feed causes each decoded message to be displayed on its own line.
- Definition:** Defines a one to four character data string that can be added to the front of the decoded data.
- Serial Cmd:** <**K141,status,preamble character(s)**>
- Default:** **Disabled**
- Options:** **0 = Disabled**                      1 = Enabled (within any protocol)

## ***Preamble Characters***

- Serial Cmd:** <**K141,status,preamble character(s)**>
- Default:** **^M** corresponds to: **carriage return**.
- Options:** To enter control characters within a serial command, hold down the **Ctrl** key while typing the desired character.

*Example:* <**K141,1,CNTL-m**> to enter the control character **^M**.

**Important:** The ASCII characters <, >, and , can only be entered as hex pairs:

For <: <**K141h,,3C**>

For >: <**K141h,,3E**>

For , : <**K141h,,2C**>

# Postamble

## ***Postamble Status***

**Usage:** Useful for identifying and controlling incoming data. For example, defining the postamble as a carriage return and a line feed causes each decoded message to be displayed on its own line.

**Definition:** Allows the user to enable or disable up to four postamble characters that can be added to the end of the decoded data.

**Serial Cmd:** <**K142,status,postamble character(s)**>

**Default:** **Enabled**

**Options:** 0 = Disabled                      **1 = Enabled (within any protocol)**

## ***Postamble Characters***

**Serial Cmd:** <**K142,status,postamble character(s)**>

**Default:** **^M^J** corresponds to: **carriage return/line feed**.

**Options:** To enter control characters within a serial command, hold down the control key while typing the desired character.

**Example:** <**K142,1,CNTL-m CNTL-j**> to enter **^M^J**.

**Important:** The ASCII characters <, >, and , can only be entered as hex pairs:

For <: <**K142h,,3C**>

For >: <**K142h,,3E**>

For , : <**K142h,,2C**>







# 4 Calibration

**Contents**

Calibration Serial Commands ..... 4-2

Calibration Overview ..... 4-2

Calibration Options ..... 4-3

Calibration by ESP..... 4-10

Initiating Calibration ..... 4-12

Additional Notes about Calibration..... 4-13

This section shows the MicroHAWK ID’s calibration options, and explains the different ways that those options can be configured.

# Calibration Serial Commands

Calibration Options	<i>&lt;K529,gain,exposure,unused,symbol type,WOI framing,WOI margin, line scan height,processing&gt;</i>
---------------------	--

## Calibration Overview

**Calibration** is one of the most powerful features of the MicroHAWK ID. The calibration process can be initiated by serial command or in the **ESP** user interface.

When the reader enters calibration, it runs through an optimization cycle that sets the ideal parameters for reading symbols at the highest possible level of performance. Calibration can be specially configured to optimize specific parameters, such as gain, exposure, and symbol type.

# Calibration Options

This command specifies the operation of the calibration feature. The default configuration performs calibration on gain and symbol type. The calibration process optimizes the gain setting for the configured exposure.

## Gain

- Definition:** When enabled, gain is calibrated to provide the best available image quality and performance.  
When disabled, gain is fixed and is not part of the calibration process.
- Serial Cmd:** `<K529,gain,exposure,unused,symbol type,WOI framing,WOI margin, line scan height,processing>`
- Default:** **Enabled (Decode Required)**
- Options:** 0 = Disabled  
1 = **Enabled (Decode Required)**  
2 = Quick Calibrate

**Note:** If you choose to calibrate the reader by sending a `<@CAL>` command, a decode is also required for the calibration process to be completed successfully.

## Disabled

When disabled, gain is fixed and is not part of the calibration process.

## Enabled

When enabled, gain is calibrated to provide the best image quality and performance for the symbol present in the field of view. Calibration requires that a decodable symbol be placed in the field of view, as feedback from the symbol decoding process is used to select the best gain setting. If a symbol is not decoded the process will result in a fail condition. Gain is enabled by default.

## Quick Calibrate

**Quick Calibrate** uses the automatic gain control (AGC) feature of the image sensor to adjust the gain value such that the current image falls into a desirable region of the image sensor's sensitivity range, ensuring optimal image luminance. The image sensor converges on the optimal gain value within a few image frames.

## Exposure

**Definition:** Unless the application is static, exposure should be configured based on the application's line speed. The table below is a general guideline for exposure configurations at various line speeds.

**Serial Cmd:** <K529,gain,**exposure**,unused,symbol type,WOI framing,WOI margin, line scan height,processing>

**Default:** **Enabled (Decode Required)**

**Options:** 0 = Disabled

**1 = Enabled (Decode Required)**

2 = Quick Calibrate

**Note:** If you choose to calibrate the reader by sending a <@CAL> command, a decode is also required for the calibration process to be completed successfully.

**Note:** This table shows guidelines for exposure settings based on various line speeds. The settings shown depend on the reader's optical configuration and on symbol element size.

Exposure	Line Speed
100,000 – 4,000	Static
4,000 – 1,250	5" / sec
1,250 – 700	10" / sec
700 – 500	15" / sec
500 – 400	20" / sec

### Disabled

When disabled, exposure is fixed and is not part of the calibration process.

### Enabled

When enabled, exposure is calibrated to provide the best image quality and performance for the symbol present in the field of view. Calibration requires that a decodable symbol be placed in the field of view, as feedback from the symbol decoding process is used to select the best exposure setting. If a symbol is not decoded the process will result in a fail condition. Exposure is enabled by default.

### Quick Calibrate

**Quick Calibrate** uses the automatic exposure control feature of the image sensor to adjust the exposure value such that the current image falls into a desirable region of the image sensor's sensitivity range, ensuring optimal image luminance. The image sensor converges on the optimal exposure value within a few image frames.

## Symbol Type

*Serial Cmd:*     <**K529**,gain,exposure,unused,**symbol type**,WOI framing,WOI margin,  
line scan height,processing>

**Default:** Enabled

Options: 0 = Disabled 1 = Enabled

## Disabled

When this feature is disabled, only the current enabled symbologies will be considered during the calibration process.

## Enabled

When this feature is enabled, autodiscrimination is in effect during the calibration process. All supported symbologies except PDF417 and Pharmacode will be attempted during calibration. Any new symbologies successfully decoded during calibration will remain enabled at the end of the process. All enabled symbologies will remain enabled. For example, assume that only Code 39 is enabled at the beginning of calibration. If a Code 128 symbol is decoded during calibration, then Code 128 as well as Code 39 will be enabled.

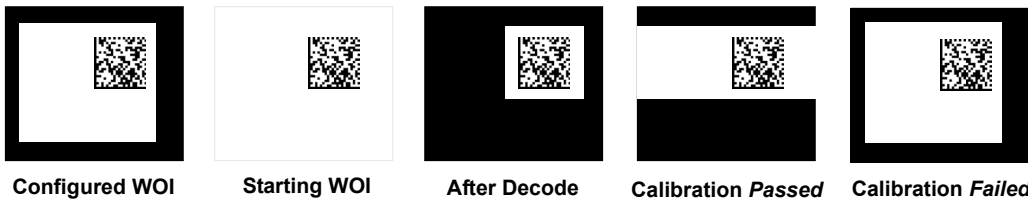
# Window of Interest (WOI) Framing

**Definition:** If a WOI framing mode is enabled, calibration will set the camera's WOI to a full-size image. When a symbol is decoded, the camera's WOI will be zoomed in both vertically and horizontally (regardless of which WOI mode is enabled) to include the symbol plus an additional margin. This is done to speed up the calibration process. When calibration is successfully completed, the camera's WOI will be adjusted according to the mode enabled. Otherwise, the original WOI configuration will be retained.

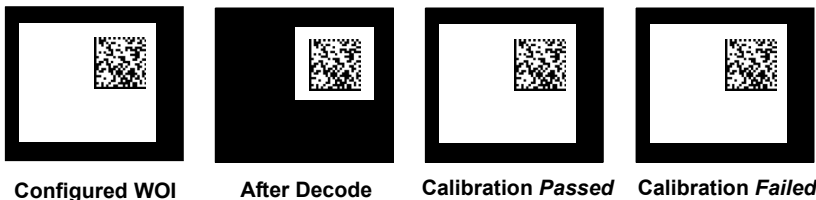
**Serial Cmd:** <K529,gain,exposure,unused,symbol type,**WOI framing**,WOI margin, line scan height,processing>

**Default:** Disabled

**Options:** **0 = Disabled**  
1 = Row and Column  
2 = Row  
3 = Column  
4 = Straight Line  
5 = Straight Line Framed



If WOI framing is not enabled, the current WOI configuration will be used until a symbol has been decoded. After a symbol is decoded, the WOI is framed exactly as it would be if a WOI framing mode was enabled. When calibration completes, the original WOI configuration is restored. Refer to the flow diagram below:



## Disabled

When this feature is disabled, the Window of Interest is not modified after the calibration process is complete.

## Row and Column

If the calibration process is successful, the Window of Interest will be modified to frame the symbol as well as an additional margin area around the symbol, determined by the WOI Margin parameter. Refer to the diagram below:



## Column

If the calibration process is successful, the Window of Interest columns will be modified to vertically frame the symbol, plus an additional margin area around the symbol, determined by the WOI Margin parameter. Refer to the diagram below:



## Row

If the calibration process is successful, the Window of Interest rows will be modified to horizontally frame the symbol, plus an additional margin area around the symbol, determined by the WOI Margin parameter. Refer to the diagram below:



## **Straight Line**

This feature is intended for use with linear symbologies. If the calibration process is successful, the orientation of the symbol is determined and the Window of Interest is modified according to the symbol orientation. The scan line orientation is determined to be vertical if the symbol tilt is between 225° and 315°, or between 45° and 135°. Otherwise, the scan line will be horizontal.

If the symbol is vertical, the image column size will be set by the scan height parameter, and will be configured for full row resolution. If the symbol is horizontal, the image row size will be set by the scan height parameter, and will be configured for full column resolution. The scan line will be centered in the middle of the symbol. If the symbol is tilted such that the scan line will not pass completely through the symbol, the scan width will be adjusted to include the entire symbol. Refer to the diagram below:



## **Straight Line Framed**

This parameter is the same as “Straight Line”, except that the Window of Interest will also frame the scan line on the symbol length as well. The scan line includes the symbol plus an additional margin area determined by the WOI Margin parameter.



## Window of Interest (WOI) Margin

**Definition:** Sets the margin size that is applied to the calibrated symbol. This parameter is expressed in number of pixels. If the margin causes the image to exceed the maximum image size, it will be reduced accordingly.

**Serial Cmd:** <**K529**,gain,exposure,unused,symbol type,WOI framing,**WOI margin**,line scan height,processing>

**Default:** **75 (pixels)**

**Options:** 20 to 1280

## Line Scan Height

**Definition:** This parameter is only used with the Straight Line modes. It sets the scan height of the straight-line image, and it is expressed in number of pixels.

**Serial Cmd:** <**K529**,gain,exposure,unused,symbol type,WOI framing,WOI margin, **line scan height**,processing>

**Default:** **64 (pixels)**

**Options:** 3 to 1280

## Processing

**Definition:** This setting defines the amount of time and effort the reader will spend attempting to decode a symbol for each parameter configuration.

**Serial Cmd:** <**K529**,gain,exposure,unused,symbol type,WOI framing,WOI margin, **line scan height**,**processing**>

**Default:** **Medium**

<b>Options:</b>	0 = Low	<b>1 = Medium</b>
	2 = High	<b>3 = Definable</b>

### Low

The reader will spend a low amount of effort attempting to decode the given symbol for each parameter configuration.

### Medium

The reader will spend a medium amount of effort attempting to decode the given symbol for each parameter configuration.

### High

The reader will spend a high amount of effort attempting to decode the given symbol for each parameter configuration.

### Definable

The processing time for each image frame is defined by **Image Processing Timeout**.

## Calibration by ESP

ESP's **Calibration** view incorporates all the functionality of the **<K529> (Calibration Options)** command in a single easy-to-use, intuitive interface. This calibration process allows the user much finer control of individual parameters than the calibration routine in **EZ Mode** or on the **Video** tab.

### Before/During Calibration

The screenshot shows the ESP Calibration interface with several callouts explaining its features:

- Before and After tabs** give the user an authentic representation of the difference in what the reader "sees" before and after the calibration routine.
- Exposure, Gain, and Image Quality** values are all tracked in real-time during the calibration process.
- Gain and Shutter Speed** are configurable in the **Calibration** interface.
- Processing** dropdown menu determines the amount of time and effort the reader will spend attempting to decode a symbol for each parameter configuration. (Low, Medium, High, Definable).
- Calibrate** button initiates the calibration routine.
- Cancel** button cancels the calibration process, if necessary.
- Save** button saves the calibration settings.
- Window of Interest** section allows the user to make precise adjustments to **WOI Framing**, **WOI Margin** (in pixels), and the scan height of the straight-line image (in pixels).

The interface includes a **Before** and **After** tab, a video feed, and a bar chart showing **Shutter** (250), **Gain** (32), and **Quality** (8792). The **Processing** dropdown is set to **Medium\***. The **Calibrate** button is highlighted.

Gain and Shutter Speed are configurable in the **Calibration** interface.

Gain: 32  
Shutter Speed: 250

Processing: Medium\*

Calibrate

Cancel

Save

Window of Interest

WOI Framing: Disabled\*

WOI Margin: 75

Linescan Height: 64

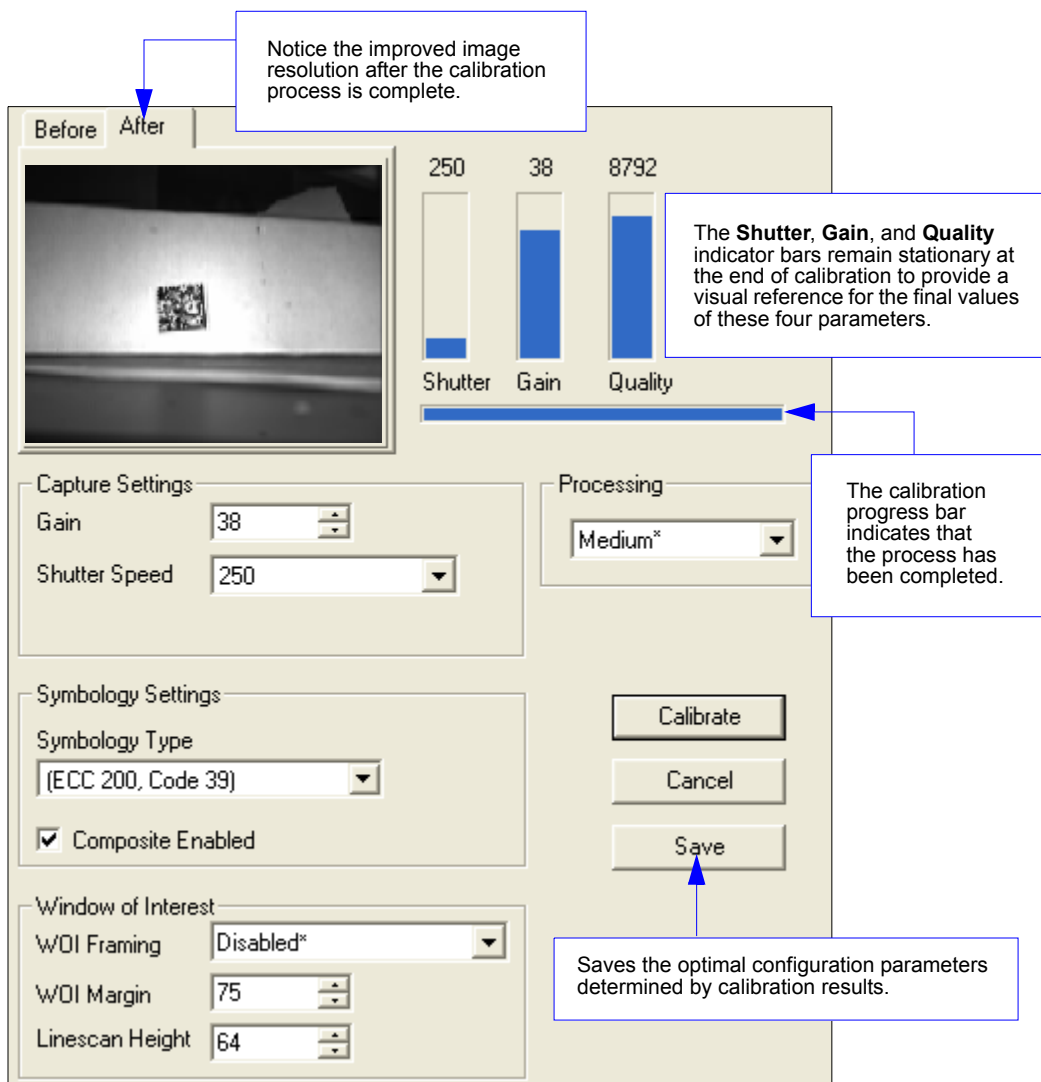
Dropdown menu of 1D and 2D symbologies, and a check box for enabling or disabling **Composite** symbologies.

(ECC 200, Code 39)

☒ Composite Enabled

The **Window of Interest** section of the **Calibration** interface allows the user to make precise adjustments to **WOI Framing**, **WOI Margin** (in pixels), and the scan height of the straight-line image (in pixels).

## After Calibration



## Initiating Calibration

Calibration can be initiated by serial command or from the **Calibration** interface in **ESP**. A symbol must be in the reader's field of view during the calibration process.

In **ESP**, the **Calibrate** button starts the calibration routine.



**Note:** If you choose to calibrate the reader by sending a **<@CAL>** command, a decode is also required for the calibration process to be completed successfully.

## Additional Notes about Calibration

The following conditions apply to the reader's calibration process. Some of these items are noted at various points throughout this section, or in other sections of the reader's documentation.

1. A symbol must be in the reader's field of view during the calibration process.
2. The **Illumination Brightness** parameter **<K514>** is not part of the calibration process.
3. If **Window of Interest Framing** is enabled, the WOI will be set to full frame when calibration begins. If **WOI Framing** is disabled, the current WOI configuration will be used for the **Search Pass**.
4. If **Symbol Type** is enabled for calibration (**Autodiscriminate**), then Interleaved 2 of 5 **Range Mode Status (Interleaved 2 of 5) <K472>** will be enabled. This allows variable-length Interleaved 2 of 5 symbols to be decoded. If an Interleaved 2 of 5 symbol is decoded during calibration, then code length # 1 will be set to the decoded symbol length at the end of calibration. Otherwise, the symbol lengths will be restored to their original configurations.
5. **Pharmacode** is not calibrated.
6. All symbol types that were enabled before calibration will still be enabled after calibration. For example, if Data Matrix ECC 200 was enabled before calibration, and the calibration routine was performed on a Code 128 symbol, then after calibration is completed both Data Matrix ECC 200 and Code 128 will be enabled.
7. The calibration process does not modify the global **Composite** status **<K453>**. The global Composite status must be configured properly before calibration.
8. If the user requires calibration of a **stacked** symbology **<K482>**, **<K483>**, or **<K484>**, that symbology must be enabled and configured appropriately before calibration.
9. **UPC Supplementals** will be disabled during calibration.
10. The **Search** process will use the configurable Window of Interest for image captures. However, when the search process is completed, the WOI will be reduced to include only the symbol of interest and some additional boundary area.



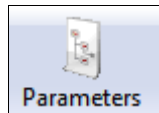
# 5 Read Cycle

## **Contents**

Read Cycle by ESP .....	5-2
Read Cycle Serial Commands .....	5-3
Read Cycle Setup .....	5-4
Multisymbol .....	5-5
Trigger Mode and Filter Duration .....	5-6
External Trigger Polarity .....	5-11
Serial Trigger .....	5-12
Start Trigger Character (Non-Delimited) .....	5-13
Stop Trigger Character (Non-Delimited) .....	5-14
End of Read Cycle .....	5-15
Capture Mode .....	5-17
Capture Timing .....	5-21
Image Processing Timeout .....	5-23
Image Storage .....	5-24
Minimum Good Reads .....	5-26

After you've established communications and completed basic read rate testing, you will need to address the spatial and timing parameters associated with your application. This section explains those parameters.

# Read Cycle by ESP



Click the **Parameters** button and then the **Read Cycle** tab.

Parameters	ESP Values
<b>Read Cycle</b>	
+ Multisymbol	
- Trigger	
Mode	Continuous Read Auto
+ Trigger Filter Duration	
External Trigger State	Active Closed
+ Serial Trigger	
Decodes Before Output	1
- End of Read Cycle	
Mode	Timeout*
Read Cycle Timeout	
+ Capture Mode	
+ Capture Time	
+ Store No Read Image	
+ Image Processing Settings	

Timeout\*

Timeout\*

New Trigger

Timeout or New Trigger

Last Frame

Last Frame or New Trigger

To open nested options, **single-click** the +.

To change a setting, **double-click** the setting and use your cursor to scroll through the options.



# Read Cycle Serial Commands

Trigger Mode/Filter Duration	<K200,trigger mode,leading edge trigger filter,trailing edge trigger filter>
Serial Trigger Character	<K201,serial trigger character>
External Trigger State	<K202,external trigger state>
End of Read Cycle	<K220,end of read cycle mode,read cycle timeout>
Decodes Before Output	<K221,decodes before output>
Multisymbol	<K222,number of symbols,multisymbol separator>
Start Trigger Character	<K229,start character>
Stop Trigger Character	<K230,stop character>
Capture Mode	<K241,capture mode,number of rapid captures,rapid capture mode,number of continuous captures,images per read cycle limit,read cycle history limit>
Capture Time	<K242,time before first capture,time between captures 1 and 2,timebetween captures 2 and 3,time between captures 3 and 4,time between captures 4 and 5,time between captures 5 and 6,time between captures 6 and 7,time between captures 7 and 8>
Image Storage	<K244,image storage type,image store mode>
Image Processing Timeout	<K245,image processing timeout>

## Read Cycle Setup

Setting up read cycle and triggering parameters will involve a series of decisions based on your particular application, as follows:

1. Select the number of symbols to be read in a single cycle. The MicroHAWK ID can read multiple symbols in a single image frame.
2. Decide on the trigger type to be used: if serial, choose a serial character; if external, choose either **Level** or **Edge**.
3. Designate how the read cycle should end (**Timeout**, **New Trigger**, **Last Frame**).
4. Select **Capture Mode**, **Continuous Mode**, or **Rapid Mode**.
5. Select **Number of Captures** (if in **Rapid Capture Mode**).
6. Set the **Time Before First Capture** and **Time Between Captures**, if any.

**Note:** The capture rate increases as the frame size decreases.

# Multisymbol

- Usage:** Multisymbol is commonly used in shipping applications where a shipping symbol contains individual symbols for part number, quantity, etc. This feature allows on trigger to pick up all the symbols.
- Definition:** **Multisymbol** allows the user to define up to 100 symbols that can be read in a single read cycle.
- Conditions:** The following conditions apply:
- Each symbol must be different to be read, unless in **Rapid Capture Mode**, configured for “triggered capture”.
  - The maximum number of characters in a read cycle is 3,000 for all symbols.
  - All No Read messages are posted at the end of the data string, unless output filtering is enabled.
  - If more than one symbol is within the field of view at the same time, symbol data may not be displayed in the order of appearance.
  - If **Matchcode Type** is set to **Sequential** or if **Trigger** is set to **Continuous Read 1 Output**, the reader will behave as if **Number of Symbols** were set to 1, regardless of the user-defined configuration.

## Number of Symbols

- Definition:** **Number of Symbols** is the number of different symbols that can be read in a single read cycle.
- Serial Cmd:** `<K222,number of symbols, multisymbol separator>`
- Default:** 1
- Options:** 1 to 100

## Multisymbol Separator

- Usage:** Used to delimit or separate data fields with a user defined character.
- Definition:** Any valid ASCII character, inserted between each symbol read when Multisymbol is set to any number greater than 1.
- Serial Cmd:** `<K222,number of symbols,multisymbol separator>`
- Default:** , (comma)
- Options:** Any available ASCII character.

**Important:** The ASCII characters <, >, and , can only be entered as hex pairs:

For <: `<K222h,,3C>`

For >: `<K222h,,3E>`

For , : `<K222h,,2C>`

**Note:** If No Read messages are disabled and there are No Reads occurring, separators will only be inserted between symbol data outputs.

# Trigger Mode and Filter Duration

## Trigger Mode

- Definition:** The **Trigger** is the event that initiates a read cycle.  
**Note:** When calibrating the reader or testing read rate, the current trigger setting will be disregarded.
- Serial Cmd:** <K200,trigger mode,leading edge trigger filter,trailing edge trigger filter>
- Default:** **Continuous Read**
- Options:** **0 = Continuous Read**  
1 = Continuous Read 1 Output  
2 = External Level  
3 = External Edge  
4 = Serial Data  
5 = Serial Data and Edge

## Continuous Read

- Usage:** **Continuous Read** is useful in testing symbol readability or reader functions. It is not recommended for normal operations.
- Definition:** In **Continuous Read**, trigger input options are disabled, the reader is always in the read cycle, and it will attempt to decode and transmit every capture. If a single symbol stays within read range for multiple read cycles, its data will be transmitted repeatedly until it leaves the read range.  
The reader sends replies to serial commands that require responses when symbol data is transmitted, or read cycle timeout is enabled and a timeout occurs *and* at least one captured image has been processed. Depending on the combination of enabled symbologies, the reader may take longer than the timeout to process a captured image.  
**Note:** When to Output and No Read options have no affect on **Continuous Read**.
- Serial Cmd:** <K200,0>

## Continuous Read 1 Output

**Usage:** **Continuous Read 1 Output** can be useful in applications where it is not feasible to use a trigger and all succeeding symbols contain different information. It is also effective in applications where the objects are presented by hand.

**Definition:** In **Continuous Read 1 Output** the reader self-triggers whenever it decodes a new symbol or a timeout occurs.

If **End of Read Cycle** is set to **Timeout** and the symbol doesn't change, the output is repeated at the end of each timeout period. For example, if **Timeout** is set to one second, the reader sends the symbol data immediately and repeats the output at intervals of one second for as long as it continues to capture the symbol.

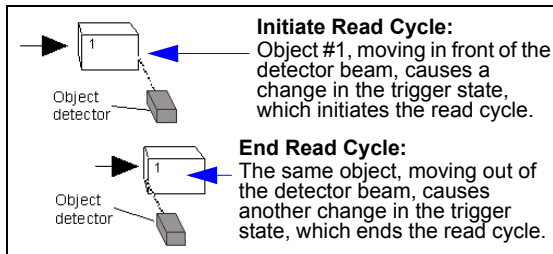
If **End of Read Cycle** is set to **New Trigger**, the reader will send the current symbol data immediately, *but only once*. A new symbol appearing in the reader's range will be read and sent immediately, provided it is not identical to the previous symbol.

**Serial Cmd:** <K200,1>

**Caution:** In automated environments, **Continuous Read 1 Output** is not recommended because there is typically no reliable way to verify that a symbol was missed.

**Note:** If **Trigger Mode** is set to **Continuous Read 1 Output**, the reader will behave as if **Number of Symbols** were set to 1, regardless of the user-defined configuration.

## External Level



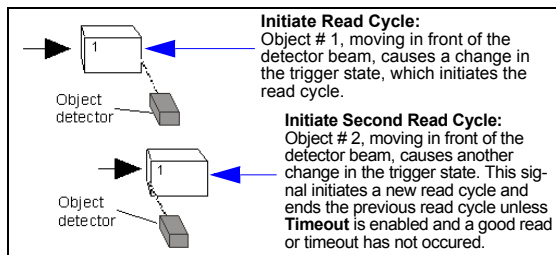
**Usage:** This mode is effective in an application where the speeds of the conveying apparatus are variable and the time the reader spends reading each object is not predictable. It also allows the user to determine if a No Read has occurred.

**Definition:** **External Level** allows the read cycle (active state) to begin when a trigger (change of state) from an external sensing device is received. The read cycle persists until the object moves out of the sensor range and the active trigger state changes again.

**Serial Cmd:** <K200,2>

**Important:** **Level** and **Edge** apply to the active logic state (**Active Open** or **Active Closed**) that exists while the object is in a read cycle, between the rising edge and the falling edge. **Rising edge** is the trigger signal associated with the appearance of an object. **Falling edge** is the trigger signal associated with the subsequent disappearance of the object. This applies both to **External Level** and **External Edge**.

## External Edge



**Usage:** This mode is highly recommended in any application where conveying speed is constant, or if spacing, object size, or read cycle timeouts are consistent.

**Definition:** **External Edge**, as with **Level**, allows the read cycle (active state) to begin when a trigger (change of state) from an external sensing device is received. However, the passing of an object out of sensor range does not end the read cycle. The read cycle ends with a good read output, or, depending on the **End of Read Cycle** setting, a timeout or new trigger occurs.

**Serial Cmd:** <K200,3>

## Serial Data

*Usage:* **Serial Data** is effective in a highly controlled environment where the host knows precisely when the object is in the field of view. It is also useful in determining if a No Read has occurred.

*Definition:* In **Serial Data**, the reader accepts an ASCII character from the host or controlling device as a trigger to start a read cycle. A **Serial Data** trigger behaves the same as an **External Edge** trigger.

Serial commands are entered inside angle brackets, as shown here: <t>.

*Serial Cmd:* <K200,4>

**Note:** In **Serial Data**, sending a non-delimited start serial character will start a read cycle; however, a non-delimited stop serial character has no effect.

## Serial Data and Edge

*Usage:* **Serial Data or External Edge** is seldom used but can be useful in an application that primarily uses an external sensing device but occasionally needs to be triggered manually.

*Definition:* In this mode the reader accepts either a serial ASCII character or an external trigger pulse to start the read cycle.

*Serial Cmd:* <K200,5>

**Note:** In **Serial Data**, sending a non-delimited start serial character will start a read cycle; however, a non-delimited stop serial character has no effect.

## ***Leading Edge Trigger Filter***

**Usage:** Used to ignore spurious triggers when **Trigger Mode** is set to External Edge or External Level.

**Definition:** To consider a change in state on the trigger input, the level must be stable for the trigger filter duration. In an edge mode, the reader will trigger a read cycle if the active state has been uninterrupted for the entire trigger filter duration. In a level mode, the leading edge is filtered such that on an active edge, the state must be held interrupted for the trigger filter duration before a trigger will occur.

**Serial Cmd:** <**K200**,trigger mode,leading edge trigger filter,trailing edge trigger filter>

**Default:** **313** (~10 ms)

**Options:** 1 to 65535 (Trigger filter range: 32.0  $\mu$ s to 2.10 seconds)

## ***Trailing Edge Trigger Filter***

**Usage:** Used to ignore spurious triggers when **Trigger Mode** is set to External Edge or External Level.

**Definition:** To consider a change in state on the trigger input, the level must be stable for the trigger filter duration. In an edge mode, the reader will trigger a read cycle if the active state has been uninterrupted for the entire trigger filter duration. In a level mode, the trailing edge is filtered such that on the falling edge, the state must be held for the trigger filter duration before the trigger will be deemed inactive.

**Serial Cmd:** <**K200**,trigger mode,leading edge trigger filter,trailing edge trigger filter>

**Default:** **313** (~10 ms)

**Options:** 1 to 65535 (Trigger filter range: 32.0  $\mu$ s to 2.10 seconds)



## External Trigger Polarity

<i>Usage:</i>	Allows users to select the trigger polarity that will be used in their application.	
<i>Definition:</i>	Determines the active state of the trigger signal applied to the cable input of the reader.	
<i>Serial Cmd:</i>	< <b>K202,active state</b> >	
<i>Default:</i>	<b>Positive</b>	
<i>Options:</i>	0 = Negative	<b>1 = Positive</b>

## Serial Trigger

**Usage:** Allows the user to define the trigger character and delimiters that start and stop the read cycle.

**Definition:** A serial trigger is considered an online host command and requires the same command format as all host commands. It must be entered within angle bracket delimiters < > or, in the case of non-delimited triggers, it must define individual start and stop characters.

### Serial Trigger Character (Delimited)

**Usage:** Allows the user to define the trigger character that initiates the read cycle.

**Definition:** A single ASCII host serial trigger character that initiates the read cycle. A delimited trigger character is one that either starts or ends the read cycle and is enclosed by delimiters such as < >.

**Serial Cmd:** <K201,**serial trigger character**>

**Default:** **Space bar**

**Options:** Any single ASCII character, including control characters, except NUL (0x00 in hex), an existing host command character, or an on-line protocol character. Control characters entered on the command line are displayed in the menu as mnemonic characters.

**Note:** **Serial Data** or **Serial Data or External Edge** triggering mode must be enabled for **Serial Trigger Character** to take effect.

**Important:** The ASCII characters <, >, and , can only be entered as hex pairs:

For <: <K201h,3C>

For >: <K201h,3E>

For , : <K201h,2C>

## Start Trigger Character (Non-Delimited)

- Usage:** Useful in applications where different characters are required to start a read cycle.
- Definition:** A single ASCII host serial trigger character that starts the read cycle and is not enclosed by delimiters such as < and >.
- Non-delimited **Start** characters can be defined and will function according to the trigger event.
- When defining **Start** trigger characters, the following rules apply:
- In **External Edge** the reader looks only for the **Start** trigger character and ignores any **Stop** trigger character that may be defined.
  - In **External Level** the **Start** trigger character begins the read cycle and the **Stop** trigger character ends it. Note that even after a symbol has been decoded and the symbol data transmitted, the reader remains in **External Level** trigger read cycle until a **Stop** character is received.
  - In **Serial Data or External Edge** trigger mode, either a **Start** trigger character or a hardware trigger can start an edge trigger read cycle.
- Serial Cmd:** <K229,start character>
- Default:** NUL (00 in hex) (disabled)
- Options:** Two hex digits representing any ASCII character except **XON** and **XOFF**.

**Important:** The ASCII characters <, >, and , can only be entered as hex pairs:

For <: <K229h,3C>

For >: <K229h,3E>

For , : <K229h,2C>

## Stop Trigger Character (Non-Delimited)

**Usage:** Useful in applications where different characters are required to end a read cycle.

**Definition:** A single ASCII host serial trigger character that ends the read cycle and is not enclosed by delimiters such as < and >.

Non-delimited **Stop** characters can be defined and will function according to the trigger event.

When defining **Stop** trigger characters, the following rules apply:

- In **External Edge** the reader looks only for the **Start** trigger character and ignores any **Stop** trigger character that may be defined.
- In **External Level** the **Start** trigger character begins the read cycle and the **Stop** trigger character ends it. Note that even after a symbol has been decoded and the symbol data transmitted, the reader remains in **External Level** trigger read cycle until a **Stop** character is received.
- In **Serial Data or External Edge** trigger mode, either a **Start** trigger character or a hardware trigger can start an edge trigger read cycle.

**Serial Cmd:** <K230,stop character>

**Default:** NUL (00 in hex) (disabled)

**Options:** Two hex digits representing an ASCII character.

**Important:** The ASCII characters <, >, and , can only be entered as hex pairs:

For <: <K230h,3C>

For >: <K230h,3E>

For , : <K230h,2C>

# End of Read Cycle

**Definition:** The read cycle is the time during which the reader will attempt to capture and decode a symbol. A read cycle can be ended by a timeout, a new trigger, or by the last frame in a capture sequence or a combination of the above.

## End of Read Cycle Mode

**Note:** When operating in Continuous Read or Continuous Read 1 Output, the reader is always in the read cycle.

**Serial Cmd:** <K220,end of read cycle,read cycle timeout>

**Default:** Timeout

**Options:** **0 = Timeout**  
 1 = New Trigger  
 2 = Timeout or new Trigger  
 3 = Last Frame  
 4 = Last Frame or New Trigger

## Timeout

**Usage:** Typically used with **Serial Data or External Edge** and **Continuous Read 1 Output**.

It is effective in highly controlled applications when the maximum length of time between objects can be predicted. It assures that a read cycle ends before the next symbol appears, giving the system extra time to decode and transmit the data to the host.

**Definition:** **Timeout** ends the read cycle, causing the reader to stop reading symbols and send the symbol data or No Read message when the time set in **Timeout** elapses (times out), *if When to Output* is set to **End of Read Cycle**.

If in **Continuous Read 1 Output**, a timeout initiates a new read cycle and allows the same symbol to be read again.

With **External Edge**, **Serial Data**, or **Serial Data or External Edge** enabled, a timeout ends the read cycle and symbol data or a No Read message is sent to the host.

With **External Level** enabled, the read cycle does not end until the falling edge trigger occurs or a timeout occurs. The next read cycle does not begin until the next rising edge trigger.

## New Trigger

*Usage:* **New Trigger** is an effective way to end a read cycle when objects move past the reader at irregular intervals (not timing-dependent).

*Definition:* **New Trigger** ends the current read cycle and initiates a new one when a new trigger occurs. **New Trigger** refers only to a rising edge trigger. With **External Edge**, **Serial Data**, or **Serial Data or External Edge** enabled, an edge or serial trigger ends a read cycle and initiates the next read cycle. In the case of **External Level**, a falling edge trigger ends the read cycle but the next read cycle does not begin until the occurrence of the next rising edge trigger.

## Timeout or New Trigger

*Usage:* Useful in applications that require an alternative way to end the read cycle. For example, if an assembly line should stop completely or the intervals between objects are highly irregular.

*Definition:* **Timeout or New Trigger** is identical to **Timeout** except that a timeout or a new trigger (whichever occurs first) ends the read cycle.

## Last Frame

*Usage:* Useful in applications in which the number of captures needed can be defined but the timeout duration varies.

*Definition:* **Last Frame** only applies to **Rapid Capture Mode**.

## Last Frame or New Trigger

*Usage:* Useful in applications in which line speeds are irregular and a new labelled object could appear before the last frame in a **Rapid Capture** sequence.

*Definition:* **Last Frame or New Trigger** is identical to **New Trigger** except that a new trigger or last frame (whichever occurs first) ends the read cycle.

## Read Cycle Timeout

*Definition:* **Read Cycle Timeout** is the duration of the read cycle.

*Serial Cmd:* <K220,end of read cycle,read cycle timeout>

*Default:* 200 (x10 ms)

*Options:* 1 to 65535

# Capture Mode

**Definition:** **Capture Mode** relates to the way that images are captured and processed.

**Serial Cmd:** <K241,capture mode,number of captures,rapid capture mode>

**Default:** **Continuous Capture**

**Options:** 0 = Rapid Capture      1 = **Continuous Capture**

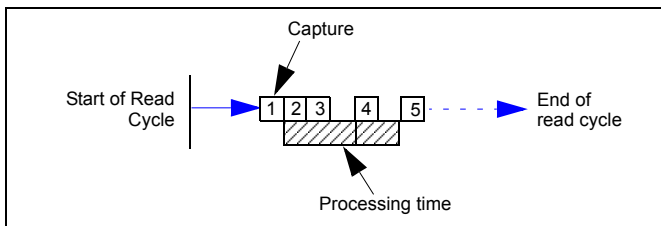
## Rapid Capture

**Definition:** In a rapid capture mode, one or multiple captures (as many as 32) can be taken at an interval specified by the time-between-captures parameter. In this mode, the only limiting time factor is integration and transfer timing.

## Continuous Capture

**Usage:** **Continuous Capture** is useful in applications with slower line speeds or where symbol spacing may be random or not time-dependent.

**Definition:** In **Continuous Capture Mode**, image captures are taken throughout the read cycle in a multi-buffered format (see diagram below). The reader begins processing the first captured image at the same time that it captures the second image. Captures will continue occurring throughout the read cycle until an end condition occurs, such as a timeout, a new trigger, the last frame in a capture sequence, or a combination of the above.



## Number of Captures

**Usage:** **Number of Captures** is used to specify the number of captures to be processed in **Rapid Capture Mode**.

**Definition:** Sets the total number of captures that are processed during a read cycle in **Rapid Capture Mode**. This feature is used in conjunction with **Capture Timing** parameters to specify the capture sequence of a rapid capture read cycle.

**Serial Cmd:** <**K241**,*capture mode*,**number of captures**,*rapid capture mode*>

**Default:** 1

**Options:** 1 to 64

**Note:** The range of maximum number of captures is dynamic. This range is dependent on the maximum image size in the system. A full-size image (1280 x 1024) reduces the maximum number of images to **6**. The smaller the image size, the greater the maximum number of captures. Once the image size is reduced to small enough dimensions, the maximum number of captures will be capped at **64**.

If a user enters a maximum capture value greater than that allowed, the value will be limited to the number of system images. This command also affects the number of stored images allowed in the system. If the maximum number of captures is selected, the number of stored images allowed will be **0**.

## Rapid Capture Mode

**Definition:** In **Rapid Capture Mode**, one or multiple captures (as many as 32) can be taken at an interval specified by the time-between-captures parameter. In this mode, the only limiting time factor is integration and transfer timing.

**Serial Cmd:** <**K241**,*capture mode*,**number of captures**,*rapid capture mode*>

**Default:** **Timed Capture**

**Options:** **0 = Timed Capture**      1 = Triggered Capture

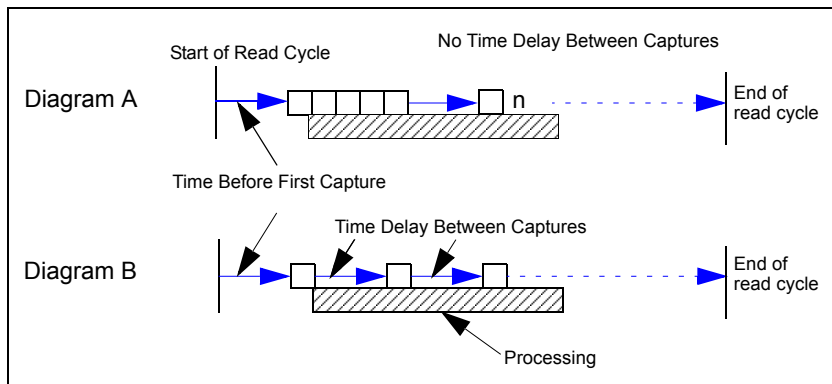


## Timed Capture

**Usage:** **Timed Rapid Capture** is useful in fast-moving applications in which symbols are only in the field of view for a short time and precise timing is required.

**Definition:** In **Timed Rapid Capture**, decoding occurs independent of and simultaneous with capturing, thus allowing precise timing or no delay at all between captures.

Also, consecutive captures are regarded as the same symbol if the output data is the same.

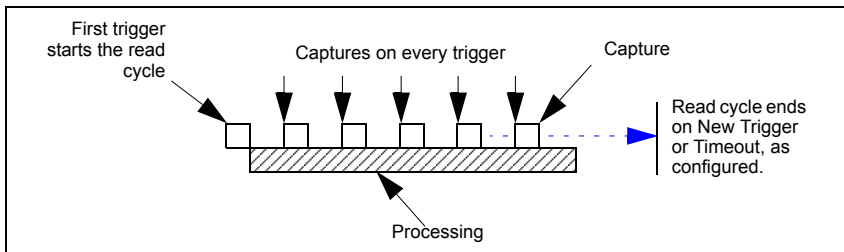


## Triggered Capture

**Usage:** Useful in applications where each decode must be treated as a discrete event, regardless of symbol data.

**Definition:** The first trigger event starts the read cycle, and subsequent triggers will continue until the predetermined **Number of Captures** is met, or until the predetermined **End of Read Cycle** condition is met-- whichever occurs first.

**Note:** If **End of Read Cycle** is set for **New Trigger** and the read cycle qualifications have not been met, the read cycle will only end *once it receives the first trigger after reaching the predetermined **Number of Captures** setting.*



# Capture Timing

**Note:** Capture Timing applies only to **Rapid Capture Mode**.

## Time Before First Capture

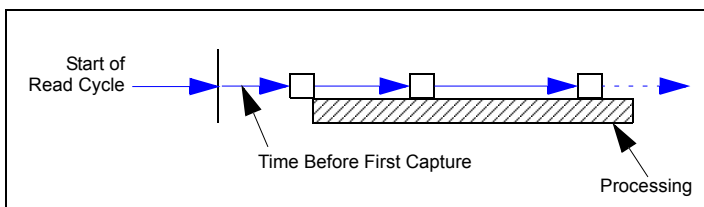
**Usage:** In almost any moving line application, a time delay is needed to ensure that a symbol will be in the reader's field of view at the beginning of the capture sequence.

**Definition:** **Time Before First Capture** in a moving line application is the time between an external trigger event and the occurrence of the first capture.

**Serial Cmd:** `<K242,time before 1st capture,time between capture 1 and capture 2,,,,,,,,time between capture 7 and capture 8>`

**Default:** **0**

**Options:** 0 to 65535 (2.097 seconds, in 32  $\mu$ S increments)



## Time Between Captures

**Usage:** This is useful in applications where more than one symbol can appear during a single read cycle (multisymbol), or where line speeds are slow enough that captured frames might overlap or miss a symbol.

**Definition:** A time delay can be inserted between individual frame captures in **Rapid Capture Mode**.

**Serial Cmd:** <K242,time before 1st capture,time between captures [time 1,time 2,...time7]

Entering 0s will result in no time between captures.

Entering a different value in each field will vary the time delays accordingly.

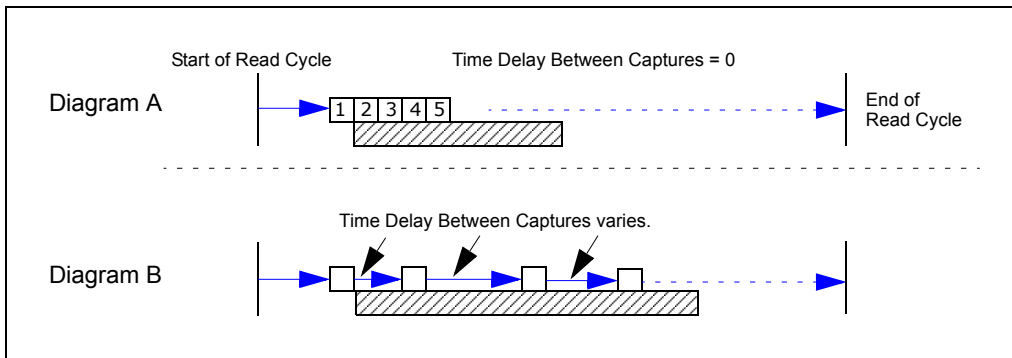
**Note:** You must enter time values along with comma separators for each field you want to change. If you omit fields, or enter only commas, the fields will remain as previously set.

**Important:** If the reader is configured to capture more than 8 images, the last (or 8th) delay value will be repeated for the remaining captures.

**Default:** 0

**Options:** 0 to 65535 (2.097 seconds, in 32  $\mu$ S increments)

**Note: Number of Captures** and number of frame delays (**Time Between Captures**) must be the same.



# Image Processing Timeout

<i>Usage:</i>	Useful in higher speed applications where image processing time is long enough that not all captures have an opportunity to be processed.
<i>Definition:</i>	Specifies the maximum amount of time to process a captured image. When the timeout expires, the image processing is aborted. This timeout works in both <b>Rapid Capture</b> and <b>Continuous Capture</b> modes, as well as with the Configuration Database.
<i>Serial Cmd:</i>	<b>&lt;K245,image processing timeout&gt;</b>
<i>Default:</i>	<b>5000 ms</b> (5 sec.)
<i>Options:</i>	1 to 65535 (in 1 ms increments)

**Notes:**

- The timeout period does not include capture time.
- If a timeout occurs during processing and no symbols in the field of view have been decoded, the image will be recorded as a No Read. For this reason, a longer timeout should be tried to ensure that the symbol is decoded successfully.

# Image Storage

## Image Storage Type

**Definition:** Allows the user to store images from separate read cycles and to retrieve them later. The number of available slots for storage depends on the mode of operation. If the reader is in **Rapid Capture Mode**, the number of images that can be stored is equivalent to the maximum number of the rapid count (the current rapid count setting). If the reader is in **Continuous Capture Mode**, a number of images equivalent to the maximum number of the rapid count minus 3.

**Serial Cmd:** <K244,image storage type,image storage mode>

**Default:** Disabled/Clear

**Options:** 0 = Disabled/Clear                      1 = Store on No Read

### Disabled/Clear

Upon selection of this option, all saved images will be cleared and the reader will not store images for later viewing.

### Store on No Read

This option will cause the reader to store an image upon exiting the read cycle for retrieval at a later time. If multiple captures are present during the duration of a read cycle, the stored image will be the last image processed for that read cycle. This image is stored in RAM and can be retrieved as long as power is cycled to the reader, and as long as the reader has not been reset via a reset/save sequence. Other commands that can initialize storage in RAM are ones that change capture modes or put the reader in a test capture mode.

## Image Storage Mode

**Serial Cmd:** <K244,image storage type,image storage mode>

**Default:** First Mode

**Options:** 0 = First Mode                      1 = Last Mode

### First Mode

This mode allows the reader to store images until the available image memory has been filled. At that point the reader will stop saving additional images. In this mode, you will always have the first image captured, because the saving process stops once memory has been filled.

### Last Mode

In this mode, image storage continues after available memory limits are reached. The oldest image in memory is overwritten, so you will always have the most recent stored image.

## ***Image Storage Example***

The following example assumes that the reader is in a rapid capture mode of 3 captures.

***Number of Symbols:*** 1  
***Frame # 1:*** No Read  
***Frame # 2:*** No Read  
***Frame # 3:*** Good Read, Symbol # 1  
***Read Cycle Result:*** Good Read  
***Stored Frame:*** **None**

***Number of Symbols:*** 1  
***Frame # 1:*** No Read  
***Frame # 2:*** No Read  
***Frame # 3:*** No Read  
***Read Cycle Result:*** No Read  
***Stored Frame:*** **Frame # 3**

***Number of Symbols:*** 2  
***Frame # 1:*** No Read  
***Frame # 2:*** No Read  
***Frame # 3:*** Good Read, Symbol # 1  
***Read Cycle Result:*** No Read  
***Stored Frame:*** **Frame # 2**

## Minimum Good Reads

*Definition:* This value specifies the number of times a symbol needs to be read to qualify as a good read.

*Serial Cmd:* **<K221,minimum good reads>**

*Default:* **1**

*Options:* 1 to 255



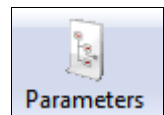
# 6 *Symbologies*

## **Contents**

Symbologies by ESP .....	6-2
Symbologies Serial Commands .....	6-3
Data Matrix .....	6-4
Aztec .....	6-6
QR Code .....	6-7
Micro QR Code.....	6-8
Code 39.....	6-9
Code 128/EAN 128 .....	6-12
BC412 .....	6-15
Interleaved 2 of 5.....	6-16
Code 93.....	6-19
Codabar.....	6-20
UPC/EAN .....	6-23
Pharmacode .....	6-27
Postal Symbologies.....	6-29
GS1 DataBar .....	6-34
PDF417 .....	6-36
MicroPDF417 .....	6-37
Composite .....	6-38

This section describes the various symbol types that can be decoded by the MicroHAWK ID.

# Symbologies by ESP



Parameters

Click the **Parameters** button and then the **Symbologies** tab.

To change a setting, **double-click** the setting and use your cursor to scroll through the options.

Parameters	ESP Values
[-] Symbologies	
[-] 2D Symbologies	
[+] Data Matrix	
QR Code	Enabled
Micro QR Code	Enabled
Aztec Code	Disabled
[-] DotCode	Disabled
Rotation Mode	No Rotation* No Rotation* Low Rotation Omni Directional Disabled
[+] 1D Symbologies	
[+] Stacked Symbologies	
[+] Composite	
Narrow Margin Status	Disabled
Damaged Symbol Status	Disabled
Symbology ID	Disabled

To open nested options, **single-click** the +.

# Symbologies Serial Commands

Composite	<K453,symbology status,separator status,separator character>
Aztec	<K458,status>
Micro QR Code	<K459,status>
Postal Symbologies	<K460,postal symbology type,POSTNET status,PLANET status,USPS4CB status>
Code 39	<K470,status,check character status,check character output status, large intercharacter gap,fixed symbol length status, fixed symbol length,full ASCII set>
Codabar	<K471,status,start/stop match,start/stop output,large intercharacter gap,fixed symbol length status,symbol length,check character type,check character output>
Interleaved 2 of 5	<K472,status,check character status,check character output status, symbol length #1, symbol length #2, guard bar status, range mode status>
UPC/EAN	<K473,UPC status,EAN status,supplemental status,separator status, separator character,supplemental type,format UPC-E as UPC-A>
Code 128/EAN 128	<K474,status,fixed symbol length status,fixed symbol length,EAN 128 status,output format,application record separator status,application record separator character,application record brackets,application record padding>
Code 93	<K475,status,fixed symbol length status,symbol length>
PDF417	<K476,status,unused,fixed symbol length status,symbol length>
Pharmacode	<K477,status,fixed bar count status,fixed bar count,minimum bar count,bar width mode,direction,fixed threshold value>
Data Matrix	<K479,ECC 200 status, ECC 000 status,ECC 050 status, ECC 080 status,ECC 100 status,ECC 140 status,ECC 120 status,ECC 130 status>
QR Code	<K480,status>
BC412	<K481,status,check character output,fixed symbol length status, fixed symbol length>
DataBar-14	<K482,status>
DataBar Limited	<K483,status>
DataBar Expanded	<K484,status,fixed symbol length status,fixed symbol length>
MicroPDF417	<K485,status,[unused],fixed symbol length status,fixed symbol length>
DotCode	<K497,status,rotation mode>

## Data Matrix

**Usage:** Very useful where information needs to be packed into a small area, and/or where symbols need to be applied directly to the substrate with laser etching, chemical etching, dot peen, or other methods.

**Definition:** Data Matrix is a type of Matrix symbology and has subsets ECC 000 - ECC 200. ECC 200 symbols have an even number of rows and an even number of columns. Most of the symbols are square with sizes from 10x10 to 144x144. Some symbols, however, are rectangular, with sizes from 8x18 to 16x48. All ECC 200 symbols can be recognized by the upper right corner module being light (binary 0) instead of dark.

### ECC 200

**Definition:** When enabled, will decode ECC 200 Data Matrix symbols.

**Serial Cmd:** <**K479,ECC 200 status,ECC 000 status,ECC 050 status,ECC 080 status,ECC 100 status,ECC 140 status,ECC 120 status,ECC 130 status**>

**Default:** **Enabled**

**Note:** This is the only symbol type enabled by default.

**Options:** 0 = Disabled                      **1 = Enabled**

### ECC 000

**Definition:** When enabled, will decode ECC 000 symbols.

**Serial Cmd:** <**K479,ECC 200 status,ECC 000 status,ECC 050 status,ECC 080 status,ECC 100 status,ECC 140 status,ECC 120 status,ECC 130 status**>

**Default:** **Disabled**

**Options:** **0 = Disabled**                      1 = Enabled

### ECC 050

**Definition:** When enabled, will decode ECC 050 symbols.

**Serial Cmd:** <**K479,ECC 200 status,ECC 000 status,ECC 050 status,ECC 080 status,ECC 100 status,ECC 140 status,ECC 120 status,ECC 130 status**>

**Default:** **Disabled**

**Options:** **0 = Disabled**                      1 = Enabled

## ECC 080

**Definition:** When enabled, will decode ECC 080 symbols.

**Serial Cmd:** <**K479**,ECC 200 status,ECC 000 status,ECC 050 status,**ECC 080 status**,  
ECC 100 status,ECC 140 status,ECC 120 status,ECC 130 status>

**Default:** **Disabled**

**Options:** **0 = Disabled**                      1 = Enabled

## ECC 100

**Definition:** When enabled, will decode ECC 100 symbols.

**Serial Cmd:** <**K479**,ECC 200 status,ECC 000 status,ECC 050 status,ECC 080 status,  
**ECC 100 status**,ECC 140 status,ECC 120 status,ECC 130 status>

**Default:** **Disabled**

**Options:** **0 = Disabled**                      1 = Enabled

## ECC 140

**Definition:** When enabled, will decode ECC 140 symbols.

**Serial Cmd:** <**K479**,ECC 200 status,ECC 000 status,ECC 050 status,ECC 080 status,  
ECC 100 status,**ECC 140 status**,ECC 120 status,ECC 130 status>

**Default:** **Disabled**

**Options:** **0 = Disabled**                      1 = Enabled

## ECC 120

**Definition:** When enabled, will decode ECC 120 symbols.

**Serial Cmd:** <**K479**,ECC 200 status,ECC 000 status,ECC 050 status,ECC 080 status,  
ECC 100 status,ECC 140 status,**ECC 120 status**,ECC 130 status>

**Default:** **Disabled**

**Options:** **0 = Disabled**                      1 = Enabled

## ECC 130

**Definition:** When enabled, will decode ECC 130 symbols.

**Serial Cmd:** <**K479**,ECC 200 status,ECC 000 status,ECC 050 status,ECC 080 status,  
ECC 100 status,ECC 140 status,ECC 120 status,**ECC 130 status**>

**Default:** **Disabled**

**Options:** **0 = Disabled**                      1 = Enabled

## Aztec

*Usage:* Used in document imaging, railway ticket validation, and some postal applications.

*Definition:* A 2D matrix symbology built on a square grid with a square “bull’s-eye” pattern at the center. **Aztec** can encode up to 3,832 numeric or 3,067 alphabetical characters, or 1,914 bytes of data.

The level of Reed-Solomon error correction used with Aztec is configurable, from 5% to 95% of the total data region. The recommended error correction level is 23% of symbol capacity plus codewords.

*Serial Cmd:* **<K458,status>**

*Default:* **Disabled**

*Options:* **0 = Disabled**                      1 = Enabled

# QR Code

- Usage:* Widely implemented in the automotive industry in Japan and throughout their worldwide supply chain.
- Definition:* QR Code is capable of handling numeric, alphanumeric, and byte data as well as kanji and kana characters. Up to 7,366 characters (numeric data) can be encoded using this symbol. Therefore, less space is required to encode the same amount of data in a QR Code symbol than in a conventional symbol, lowering the cost of labelling.  
Three Position Detection Patterns in the symbol make omnidirectional, ultra-fast reading possible.  
QR Code has error protection capability. Data can often be restored even if a part of the symbol has become dirty or damaged.
- Serial Cmd:* **<K480,status>**
- Default:* **Disabled**
- Options:* **0 = Disabled**                      1 = Enabled

# Micro QR Code

- Usage:*           Used in various applications that require higher data density than that provided by standard QR Code. Some application examples are automotive inventory, vehicle ID, and mobile phone URL encodation.
- Definition:*       **Micro QR Code** is a 2D matrix symbology that comes in 4 different symbol sizes, the largest capable of encoding 35 numeric characters.
- Serial Cmd:*       <**K459,status**>
- Default:*           **Disabled**
- Options:*           **0 = Disabled**                      1 = Enabled



## Code 39

- Usage:** **Code 39** is considered the standard for non-retail 1D symbology.
- Definition:** An alphanumeric symbology with unique start/stop code patterns, composed of 9 black and white elements per character, of which 3 are wide.
- Serial Cmd:** **<K470,status,check character status,check character output status,large intercharacter gap,fixed symbol length status,fixed symbol length,full ASCII set>**
- Default:** **Enabled**
- Options:** 0 = Disabled                      1 = Enabled

### Check Character Status (Code 39)

- Serial Cmd:** **<K470,status,check character status,check character output status,large intercharacter gap,fixed symbol length status,fixed symbol length,full ASCII set>**
- Default:** **Disabled**
- Options:** 0 = Disabled                      1 = Enabled

### Check Character Output Status (Code 39)

- Usage:** **Check Character Output Status**, added to the symbol, provides additional data security.
- Definition:** When enabled, the check character character is read and compared along with the symbol data. When disabled, symbol data is sent without the check character.
- Note:** With **Check Character Output Status** and an **External** or **Serial** trigger option enabled, an invalid check character calculation will cause a No Read message to be transmitted at the end of the read cycle.
- Serial Cmd:** **<K470,status,check character status,check character output status,large intercharacter gap,fixed symbol length status,fixed symbol length,full ASCII set>**
- Default:** **Disabled**
- Options:** 0 = Disabled                      1 = Enabled

## ***Large Intercharacter Gap (Code 39)***

**Usage:** **Large Intercharacter Gap** is helpful for reading symbols that are printed out of specification.

**Definition:** When enabled, the reader can read symbols with gaps between symbol characters that exceed three times (3x) the narrow element width.

**Serial Cmd:** *<K470,status,check character status,check character output status,**large intercharacter gap**,fixed symbol length status,fixed symbol length,full ASCII set>*

**Default:** **Disabled**

**Options:** **0 = Disabled**                      1 = Enabled

## ***Fixed Symbol Length Status (Code 39)***

**Definition:** When enabled, the reader will check the symbol length against the symbol length field. If disabled, any length will be considered valid.

**Serial Cmd:** *<K470,status,check character status,check character output status,**large intercharacter gap**,fixed symbol length status,fixed symbol length,full ASCII set>*

**Default:** **Disabled**

**Options:** **0 = Disabled**                      1 = Enabled

## ***Fixed Symbol Length (Code 39)***

**Usage:** **Fixed Symbol Length** helps prevent truncations and increases data integrity by ensuring that only one symbol length will be accepted.

**Definition:** Specifies the exact number of characters that the reader will recognize (this does not include start and stop and check character characters). The reader ignores any symbology that does not match the specified length.

**Serial Cmd:** *<K470,status,check character status,check character output status,**large intercharacter gap**,fixed symbol length status,**fixed symbol length**,full ASCII set>*

**Default:** **10**

**Options:** 1 to 64

## Full ASCII Set (Code 39)

- Usage:** Must be enabled when reading characters outside the standard character set (0-9, A-Z, etc.)  
The user must know in advance whether or not to use the **Full ASCII Set** option. Since **Full ASCII Set** requires two code words to encode one character, it is less efficient.
- Definition:** Standard Code 39 encodes 43 characters; zero through nine, capital “A” through capital “Z”, minus symbol, plus symbol, forward slash, space, decimal point, dollar sign, and percent symbol. When **Full ASCII Set** is enabled, the reader can read the full ASCII character set, from 0 to 255.
- Serial Cmd:** *<K470,status,check character status,check character output status,large intercharacter gap,fixed symbol length status,fixed symbol length,full ASCII set>*
- Default:** **Disabled**
- Options:** **0 = Disabled**                      1 = Enabled

## Code 128/EAN 128

- Usage:** **Code 128** is a smaller symbology useful in applications with tight spots and high security needs.
- Definition:** A very dense alphanumeric symbology. It encodes all 128 ASCII characters, it is continuous, has variable length, and uses multiple element widths measured edge to edge.
- Serial Cmd:** *<K474,status,fixed symbol length status,fixed symbol length,EAN 128 status,output format,application record separator status,application record separator character,application record brackets,application record padding>*
- Default:** **Enabled**
- Options:** 0 = Disabled                      1 = **Enabled**

### **Fixed Symbol Length Status (Code 128/EAN 128)**

- Definition:** When enabled, the reader will check the symbol length against the symbol length field. If disabled, any length will be considered a valid symbol.
- Serial Cmd:** *<K474,status,fixed symbol length status,fixed symbol length,EAN 128 status,output format,application record separator status,application record separator character,application record brackets,application record padding>*
- Default:** **Disabled**
- Options:** 0 = **Disabled**                      1 = Enabled

### **Fixed Symbol Length (Code 128/EAN 128)**

- Usage:** **Fixed Symbol Length** helps prevent truncations and increases data integrity by ensuring that only one symbol length will be accepted.
- Definition:** This specifies the exact number of characters that the reader will recognize (this does not include start, stop, and check character characters). The reader ignores any symbol not having the specified length.
- Serial Cmd:** *<K474,status,fixed symbol length status,fixed symbol length,EAN 128 status,output format,application record separator status,application record separator character,application record brackets,application record padding>*
- Default:** **10**
- Options:** 1 to 64

## ***EAN 128 Status (Code 128/EAN 128)***

- Definition:** When this field is disabled, the reader will not check any Code 128 labels for conformance to EAN requirements, or perform any special formatting. When enabled, the reader can read symbols with or without a function 1 character in the first position. If a symbol has a function 1 character in the first position, it must conform to EAN format. Symbols that conform to EAN format will also be subject to the special output formatting options available in this command.
- Note:** Code 128 status must be enabled for EAN status to be active. If EAN status is required, the reader will only decode symbols that have a function 1 character in the first position and that conform to EAN format. All symbols read will be subject to the special output formatting options available in this command.
- Note:** Code 128 status must be enabled for EAN status to be active.
- Serial Cmd:** *<K474,status,fixed symbol length status,fixed symbol length,EAN 128 status,output format,application record separator status,application record separator character,application record brackets,application record padding>*
- Default:** **Disabled**
- Options:** **0 = Disabled**                      1 = Enabled                      2 = Required

## ***Output Format (Code 128/EAN 128)***

- Definition:** In **Standard**, the reader will not apply special EAN output formatting options. In **Application**, the reader will apply the special EAN output formatting options to decoded EAN-conforming symbols.
- Serial Cmd:** *<K474,status,fixed symbol length status,fixed symbol length,EAN 128 status,output format,application record separator status,application record separator character,application record brackets,application record padding>*
- Default:** **Standard**
- Options:** **0 = Standard**                      1 = Application

## ***Application Record Separator Status (Code 128/EAN 128)***

- Definition:** When enabled, an EAN separator will be inserted into the output between fields whenever an EAN-conforming symbol is decoded and EAN output formatting applies.
- Serial Cmd:** *<K474,status,fixed symbol length status,fixed symbol length,EAN 128 status,output format,application record separator status,application record separator character,application record brackets,application record padding>*
- Default:** **Disabled**
- Options:** **0 = Disabled**                      1 = Enabled

## ***Application Record Separator Character (Code 128/EAN 128)***

**Definition:** This is an ASCII character that serves as an EAN separator in formatted EAN output.

**Serial Cmd:** *<K474,status,fixed symbol length status,fixed symbol length,EAN 128 status,output format,application record separator status,application record separator character,application record brackets,application record padding>*

**Default:** ,

**Options:** Any ASCII character (7 bit)

**Important:** The ASCII characters <, >, and , can only be entered as hex pairs:

For <: **<K474h,,,,,,3C>**

For >: **<K474h,,,,,,3E>**

For , : **<K474h,,,,,,2C>**

## ***Application Record Brackets (Code 128/EAN 128)***

**Definition:** If an EAN-conforming symbol is decoded and EAN formatting applies, this feature places bracket characters around the application identifiers in the formatted output.

**Serial Cmd:** *<K474,status,fixed symbol length status,fixed symbol length,EAN 128 status,output format,application record separator status,application record separator character,application record brackets,application record padding>*

**Default:** **Disabled**

**Options:** **0 = Disabled**                      1 = Enabled

## ***Application Record Padding (Code 128/EAN 128)***

**Definition:** This feature causes the reader to pad variable-length application fields with leading zeroes. This is not done for the last field of a symbol.

**Serial Cmd:** *<K474,status,fixed symbol length status,fixed symbol length,EAN 128 status,output format,application record separator status,application record separator character,application record brackets,application record padding>*

**Default:** **Enabled**

**Options:** **0 = Disabled**                      **1 = Enabled**

## BC412

- Usage:** Widely used in semiconductor manufacturing. Particularly useful where speed, accuracy, and ease of printing are required.
- Definition:** BC412 (Binary Code 412), a proprietary IBM symbology since 1988, is an alphanumeric symbol with a set of 35 characters, each encoded by a set of 4 bars in 12 module positions. All bars have a single width; it is the presence (1) or absence (0) of bars in each of the twelve module positions that make BC412 binary.  
This symbology is also bi-directional and self-clocking, with a start character and a stop character.
- Serial Cmd:** *<K481,status,check character output,fixed symbol length status,fixed symbol length>*
- Default:** **Disabled**
- Options:** **0 = Disabled**                      1 = Enabled

### Check Character Output (BC412)

- Usage:** **Check Character Output**, added to the symbol, provides additional security.
- Definition:** When enabled, the check character character is read and compared along with the symbol data. When disabled, symbol data is sent without the check character.
- Serial Cmd:** *<K481,status,check character output,fixed symbol length status,fixed symbol length>*
- Default:** **Disabled**
- Options:** **0 = Disabled**                      1 = Enabled

### Fixed Symbol Length Status (BC412)

- Definition:** When enabled, the reader will check the symbol length against the symbol length field. If disabled, any length will be considered valid.
- Serial Cmd:** *<K481,status,check character output,fixed symbol length status,fixed symbol length>*
- Default:** **Disabled**
- Options:** **0 = Disabled**                      1 = Enabled

### Fixed Symbol Length (BC412)

- Definition:** When enabled, the check character character is read and compared along with the symbol data. When disabled, symbol data is sent without the check character.
- Serial Cmd:** *<K481,status,check character output,fixed symbol length status,fixed symbol length>*
- Default:** **10**
- Options:** 1 to 64

## Interleaved 2 of 5

**Usage:** I-2/5 has been popular because it is the most dense symbology for printing numeric characters less than 10 characters in length; however, Microscan does not recommend this symbology for any new applications because of inherent problems such as truncation.

**Definition:** A dense, continuous, self-checking, numeric symbology. Characters are paired together so that each character has five elements, two wide and three narrow, representing numbers 0 through 9, with the bars representing the first character and the interleaved spaces representing the second character. (A check character is highly recommended).

**Important:** You must set Symbol Length in order to decode I-2/5 symbols, unless **Range Mode** is enabled.

**Serial Cmd:** <**K472,status,check character status,check character output status,symbol length #1,symbol length #2,guard bar status,range mode status**>

**Default:** **Enabled**

**Options:** 0 = Disabled                      1 = **Enabled**

### Check Character Status (Interleaved 2 of 5)

**Usage:** This option is not typically used, but it can be enabled for additional security in applications where the host requires redundant check character verification.

**Definition:** An error correcting routine in which the check character character is added.

**Serial Cmd:** <**K472,status,check character status,check character output status,symbol length #1,symbol length #2,guard bar status,range mode status**>

**Default:** **Disabled**

**Options:** 0 = **Disabled**                      1 = Enabled

### Check Character Output Status (Interleaved 2 of 5)

**Definition:** When enabled, a check character character is sent along with the symbol data for added data security.

**Serial Cmd:** <**K472,status,check character status,check character output status,symbol length #1,symbol length #2,guard bar status,range mode status**>

**Default:** **Disabled**

**Options:** 0 = **Disabled**                      1 = Enabled



## Symbol Length #1 (Interleaved 2 of 5)

- Usage:** Useful in applications where 1 2/5 symbols of a specific length are required.
- Definition:** The **Symbol Length # 1** field is one of two fields against which the decoded symbol is compared before accepting it as valid or rejecting it.
- Serial Cmd:** <**K472**,status,check character status,check character output,**symbol length #1**,symbol length #2,guard bar status,range mode status>
- Default:** 16
- Options:** 0 to 64, even only
- Important:** If **Range Mode** is disabled, the length of the symbol must match either **Symbol Length # 1** or **Symbol Length # 2** to be considered a valid symbol.
- If **Range Mode** is enabled, **Symbol Length # 1** and **Symbol Length # 2** form a range into which the length of the symbol must fall to be considered valid.

## Symbol Length #2 (Interleaved 2 of 5)

- Usage:** Useful in applications where 1 2/5 symbols of a specific length are required.
- Definition:** The **Symbol Length # 2** field is one of two fields against which the decoded symbol is compared before accepting it as valid or rejecting it.
- Serial Cmd:** <**K472**,status,check character status,check character output,symbol length #1,**symbol length #2**,guard bar status,range mode status>
- Default:** 6
- Options:** 0 to 64, even only
- Important:** If **Range Mode** is disabled, the length of the symbol must match either **Symbol Length # 2** or **Symbol Length # 1** to be considered a valid symbol.
- If **Range Mode** is enabled, **Symbol Length # 2** and **Symbol Length # 1** form a range into which the length of the symbol must fall to be considered valid.

## **Guard Bar Status (Interleaved 2 of 5)**

**Note:** Whenever **Guard Bar** is enabled, the presence of guard bars (also called “bearer bars”) is required for decoding to take place.

**Usage:** Useful when I-2/5 multisymbols are enabled to prevent false data output. This typically occurs with highly tilted or skewed symbols.

**Definition:** A guard bar is a heavy bar, at least twice the width of the wide bar, surrounding the printed I-2/5 symbol and helping to prevent false reads.

**Serial Cmd:** <**K472**,status,check character status,check character output,symbol length #1,symbol length #2,**guard bar status**,range mode status>

**Default:** **Disabled**

**Options:** **0 = Disabled**                      **1 = Enabled**

## **Range Mode Status (Interleaved 2 of 5)**

**Usage:** Useful in applications where I 2/5 symbols of a specific length are required.

**Definition:** When **Range Mode** is disabled, the reader checks the value of the symbol length against the values set in **Symbol Length # 1** and **Symbol Length # 2**. If the symbol length does not match either of the preset values, then it is rejected as invalid.

When **Range Mode** is enabled, **Symbol Length # 1** and **Symbol Length # 2** are combined to form a range of valid symbol lengths. Any symbol length that does not fall into this range is rejected as an invalid symbol. Either of the preset symbol length values in the **Symbol Length # 1** and **Symbol Length # 2** fields can form the start or end of the range.

**Serial Cmd:** <**K472**,status,check character status,check character output,symbol length #1,symbol length #2,guard bar status,**range mode status**>

**Default:** **Enabled**

**Options:** **0 = Disabled**                      **1 = Enabled**

## Code 93

- Usage:** Sometimes used in clinical applications.
- Definition:** Code 93 is a variable-length, continuous symbology employing four element widths. Each Code 93 character has nine modules that may be either black or white. Each character contains three bars and three spaces.
- Serial Cmd:** **<K475,status,fixed symbol length status,symbol length>**
- Default:** **Disabled**
- Options:** **0 = Disabled**                      1 = Enabled

### ***Fixed Symbol Length Status (Code 93)***

- Definition:** When disabled, the reader will accept any Code 93 symbol provided is doesn't exceed the system's maximum capabilities.  
When enabled, the reader will reject any Code 93 symbol that doesn't match the fixed symbol length.
- Serial Cmd:** **<K475,status,fixed symbol length status,symbol length>**
- Default:** **Disabled**
- Options:** **0 = Disabled**                      1 = Enabled

### ***Symbol Length (Code 93)***

- Definition:** This is the symbol length value against which all Code 93 symbols will be compared.
- Serial Cmd:** **<K475,status,fixed symbol length status,symbol length>**
- Default:** **10**
- Options:** 1 to 64

## Codabar

- Usage:** Used in photo-finishing and library applications. Previously used in medical applications, but not typically used in newer medical applications.
- Definition:** Codabar is a 16-bit character set (0 through 9, and the characters \$, :, /, ., +, and –) with start/stop codes and at least two distinctly different bar widths.
- Serial Cmd:** <K471,status,start/stop match,start/stop output,large intercharacter gap,fixed symbol length status,symbol length,check character type,check character output>
- Default:** **Disabled**
- Options:** **0 = Disabled**                      1 = Enabled

### Start/Stop Match (Codabar)

- Definition:** When disabled, the reader will decode Codabar symbols whether or not the start and stop characters are the same.  
When enabled, the reader will not decode Codabar symbols unless the start and stop characters are the same.
- Serial Cmd:** <K471,status,**start/stop match**,start/stop output,large intercharacter gap,fixed symbol length status,symbol length,check character type,check character output>
- Default:** **Enabled**
- Options:** 0 = Disabled                      1 = Enabled

### Start/Stop Output (Codabar)

- Definition:** When disabled, the start and stop characters will *not* be present in the data output of the decoded symbol.  
When enabled, the start and stop characters *will* be present in the data output of the decoded symbol.  
**Note:** Because the start and stop characters are included as part of the data, the characters must be included as part of the length in a fixed length mode of operation.
- Serial Cmd:** <K471,status,start/stop match,**start/stop output**,large intercharacter gap,fixed symbol length status,symbol length,check character type,check character output>
- Default:** **Enabled**
- Options:** 0 = Disabled                      1 = Enabled

## Large Intercharacter Gap (Codabar)

- Definition:** When disabled, the spaces between characters, or the “intercharacter gap”, are ignored during the decode process.  
**Note:** If the intercharacter space is large enough to be considered a margin, the symbol will not decode, regardless of this parameter’s setting.
- Serial Cmd:** <**K471**,status,start/stop match,start/stop output,**large intercharacter gap**,fixed symbol length status,symbol length,check character type,check character output>
- Default:** **Disabled**
- Options:** **0 = Disabled**                      1 = Enabled

## Fixed Symbol Length Status (Codabar)

- Definition:** When disabled, the reader will accept any Codabar symbol provided it doesn’t exceed the system’s maximum capabilities.  
 When enabled, the reader will reject any Codabar symbol that doesn’t match the fixed length.
- Serial Cmd:** <**K471**,status,start/stop match,start/stop output,large intercharacter gap,**fixed symbol length status**,symbol length,check character type,check character output>
- Default:** **Disabled**
- Options:** **0 = Disabled**                      1 = Enabled

## Symbol Length (Codabar)

- Definition:** This is the value against which all Codabar symbol lengths will be compared.
- Serial Cmd:** <**K471**,status,start/stop match,start/stop output,large intercharacter gap,fixed symbol length status,**symbol length**,check character type,check character output>
- Default:** **10**
- Options:** 1 to 64

# Check Character Type (Codabar)

**Definition:** When disabled, the reader will not perform any character checking calculations on decoded Codabar symbols.

When set to **Mod 16**, the reader will perform a modulus 16 check character calculation on the symbol. If the symbol does not pass this calculation, it will not be decoded.

When set to **NW7**, The reader will perform an NW7 modulus 11 check character calculation on the symbol. If the symbol does not pass this calculation, it will not be decoded.

When set to **Both**, the reader will perform both the Mod 16 and NW7 modulus 11 check character calculations on the symbol. If the symbol does not pass either calculation, it will not be decoded.

**Serial Cmd:** <**K471**,status,start/stop match,start/stop output,large intercharacter gap,fixed symbol length status,symbol length,**check character type**,check character output>

**Default:** Disabled

**Options:**    **0 = Disabled**                      1 = Mod 16                      2 = NW7 (Mod 11)  
                 3 = Mod 16 and NW7

# Check Character Output (Codabar)

**Definition:** When this field is disabled and a check character calculation is enabled, the reader will strip the verified check character from the symbol data output. This condition must be accounted for if a fixed length is also being used.

When enabled, the reader will output the check character as part of the symbol data. This condition must be accounted for if a fixed length is also being used.

**Serial Cmd:** <**K471**,status,start/stop match,start/stop output,large intercharacter gap,fixed symbol length status,symbol length,check character type,**check character output**>

**Default:** Disabled

**Options:**    **0 = Disabled**                      1 = Enabled

# UPC/EAN

- Usage:** Used primarily in point-of-sale applications in the retail industry. It is commonly used with Microscan readers in applications in combination with **Matchcode** when there is a need to verify that the right product is being placed in the right packaging.
- Definition:** **UPC** (Universal Product Code) is a fixed length, numeric, continuous symbology. UPC can have two- or five-digit supplemental bar code data following the normal code. The UPC Version A (UPC, A) symbol is used to encode a 12 digit number. The first digit is the number system character, the next five are the manufacturer number, the next five are the product number, and the last digit is the checksum character.  
When enabled, the reader will read UPC Version A and UPC Version E only.
- Serial Cmd:** <**K473**,**UPC status**,**EAN status**,**supplementals status**,**separator status**,**separator character**,**supplemental type**,**format UPC-E as UPC-A**>
- Default:** **Enabled**
- Options:** 0 = Disabled                      **1 = Enabled**

## EAN Status

- Usage:** **EAN** is the European version of the UPC symbology and is used in European market applications.  
**Note:** **UPC** must be enabled for **EAN** to take effect.
- Definition:** EAN is a subset of UPC. When enabled, the reader will read UPC Version A, UPC Version E, EAN 13, and EAN 8. It also appends a leading zero to UPC Version A symbol information and transmits 13 digits. If transmitting 13 digits when reading UPC Version A symbols is not desired, disable EAN.  
**Note:** The extra character identifies the country of origin.
- Serial Cmd:** <**K473**,**UPC status**,**EAN status**,**supplementals status**,**separator status**,**separator character**,**supplemental type**,**format UPC-E as UPC-A**>
- Default:** **Enabled**
- Options:** 0 = Disabled                      **1 = Enabled**

# Supplementals Status (UPC/EAN)

- Usage:* Reads **Supplementals** typically used in publications and documentation.
- Definition:* A supplemental is a 2 to 5 digit symbol appended to the main symbol. When set to **Enabled** or **Required**, the reader reads supplemental code data that has been appended to the standard UPC or EAN codes.
- Serial Cmd:* <**K473**,UPC status,EAN status,**supplementals status**,separator status,separator character,supplemental type,format UPC-E as UPC-A>
- Default:* **Disabled**
- Options:* **0 = Disabled**                      1 = Enabled                      2 = Required

## Disabled

UPC **Supplementals** will not be decoded.

## Enabled

When enabled, the reader will try to decode a main and a supplemental. However, if a supplemental is not decoded, the main will be sent by itself at the end of the read cycle.

## Required

When set to **Required**, both the main and the supplemental symbols must be read or a single No Read condition results.

For example, if **Supplementals** is set to **Required**, **Separator** is enabled, and an asterisk is defined as the UPC separator character. Then the data is displayed as:

MAIN \* SUPPLEMENTAL.

**Note:** Under no circumstances will the supplemental symbol data be sent without a main symbol.

**Note:** If additional symbols—other than the main or supplemental—will be read in the same read cycle, **Number of Symbols** should be set accordingly.

# Separator Status (UPC/EAN)

- Usage:* Allows users to distinguish between the main and **Supplemental** symbols.
- Definition:* A character can be inserted between the standard UPC or EAN symbology and the supplemental symbology when **Supplementals** is set to **Enabled** or **Required**.
- Serial Cmd:* <**K473**,UPC status,EAN status,supplementals status,**separator status**,separator character,supplemental type,format UPC-E as UPC-A>
- Default:* **Disabled**
- Options:* **0 = Disabled**                      1 = Enabled



## Separator Character (UPC/EAN)

**Note:** If **Separator Character** has been changed to any other character and you wish to redefine the separator as a comma, you will need to use **ESP**.

**Usage:** As required by the application.

**Definition:** Allows the user to change the separator character from a comma to a new character.

**Serial Cmd:** <**K473**,UPC status,EAN status,supplementals status,separator status,  
**separator character**,supplemental type,format UPC-E as UPC-A>

**Default:** , (comma)

**Options:** Any ASCII character.

**Note:** Whenever **Separator Character** is defined as a comma ( , ) sending a <**K473,s?**> command will return the current settings, including the separator character comma which appears after the separator status comma.

**Important:** The ASCII characters <, >, and , can only be entered as hex pairs:

For <: <**K473h**,,,,,**3C**>

For >: <**K473h**,,,,,**3E**>

For , : <**K473h**,,,,,**2C**>

# Supplemental Type (UPC/EAN)

- Usage:* As required by symbology used in application.
- Definition:* Allows the user to select 2 character or 5 character supplements, or both.
- Serial Cmd:* <**K473**,UPC status,EAN status,supplementals status,separator status,separator character,**supplemental type**,format UPC-E as UPC-A>
- Default:* **Both**
- Options:* **0 = Both**                      1 = 2 characters only                      2 = 5 characters only

## Both

Either 2 character or 5 character supplementals will be considered valid.

## 2 Characters Only

Only two character supplementals will be considered valid.

## 5 Characters Only

Only five character supplementals will be considered valid.

# Format UPC-E as UPC-A (UPC/EAN)

- Definition:* When disabled, the reader will output the version E symbols in their encoded 6-character format.  
When enabled, the reader will format the symbol as either a 12-character UPC-A symbol or an EAN-13 symbol, depending on the state of the EAN status parameter. This formatting reverses the zero suppression that is used to generate the symbol in the UPC specification.
- Serial Cmd:* <**K473**,UPC status,EAN status,supplementals status,separator status,separator character,supplemental type,**format UPC-E as UPC-A**>
- Default:* **Disabled**
- Options:* **0 = Disabled**                      1 = Enabled

# Pharmacode

- Usage:** Used mostly with packaging for the pharmaceuticals industry.
- Definition:** Encodes up to five different numbers, each with its own color, which may be entered in decimal or “binary” format with a 1 represented by a thick bar and a 0 represented by a thin bar. Bar width is independent of height. In decimal format, each part can be up to 999,999. In binary format, each input can have up to 19 ones and zeros.  
**Important:** When Pharmacode is enabled, other linear symbologies will not decode properly. Disable Pharmacode before reading other linear symbologies.
- Serial Cmd:** *<K477,status,fixed symbol length status,fixed symbol length,minimum number of bars,bar width status,direction,fixed threshold value>*
- Default:** **Disabled**
- Options:** **0 = Disabled**                      1 = Enabled

## Fixed Symbol Length Status (Pharmacode)

- Definition:** When enabled, the reader will check the symbol length against the symbol length field. If disabled, any length will be considered valid.
- Serial Cmd:** *<K477,status,fixed symbol length status,fixed symbol length,minimum number of bars,bar width status,direction,fixed threshold value>*
- Default:** **Disabled**
- Options:** **0 = Disabled**                      1 = Enabled

## Fixed Symbol Length (Pharmacode)

- Definition:** Specifies the exact number of bars that must be present for the reader to recognize and decode the Pharmacode symbol.
- Serial Cmd:** *<K477,status,fixed symbol length status,fixed symbol length,minimum number of bars,bar width status,direction,fixed threshold value>*
- Default:** **5**
- Options:** 1 to 16

## Minimum Number of Bars (Pharmacode)

- Definition:** Sets the minimum number of bars that a Pharmacode symbol must have to be considered valid.
- Serial Cmd:** *<K477,status,fixed symbol length status,fixed symbol length,minimum number of bars,bar width status,direction,fixed threshold value>*
- Default:** **2**
- Options:** 1 to 16

## **Bar Width Status (Pharmacode)**

**Definition:** If set to **Mixed**, the reader will autodiscriminate between narrow bars and wide bars. If set to **All Narrow**, all bars will be considered as narrow bars. If set to **All Wide**, all bars will be considered as wide bars. If set to **Fixed Threshold**, it will use the fixed threshold value to determine whether the bars are narrow or wide. The **Bar Width Status** setting will be ignored when the reader is able to tell the difference between the narrow and the wide bars.

**Serial Cmd:** <**K477**,status,fixed symbol length status,fixed symbol length,minimum number of bars,**bar width status**,direction,fixed threshold value>

**Default:** **Mixed**

**Options:** **0 = Mixed**  
1 = All Narrow  
2 = All Wide  
3 = Fixed Threshold

## **Direction (Pharmacode)**

**Definition:** Specifies the direction in which a symbol can be read.

**Serial Cmd:** <**K477**,status,fixed symbol length status,fixed symbol length,minimum number of bars,bar width status,**direction**,fixed threshold value>

**Default:** **Forward**

**Options:** **0 = Forward**                      1 = Reverse

## **Fixed Threshold Value (Pharmacode)**

**Definition:** Used when **Bar Width Status** is set to **Fixed Threshold**. Defines the minimum difference in pixels that will distinguish a narrow bar from a wide bar.

**Serial Cmd:** <**K477**,status,fixed symbol length status,fixed symbol length,minimum number of bars,bar width status,direction,**fixed threshold value**>

**Default:** **10**

**Options:** 1 to 65535

# Postal Symbologies

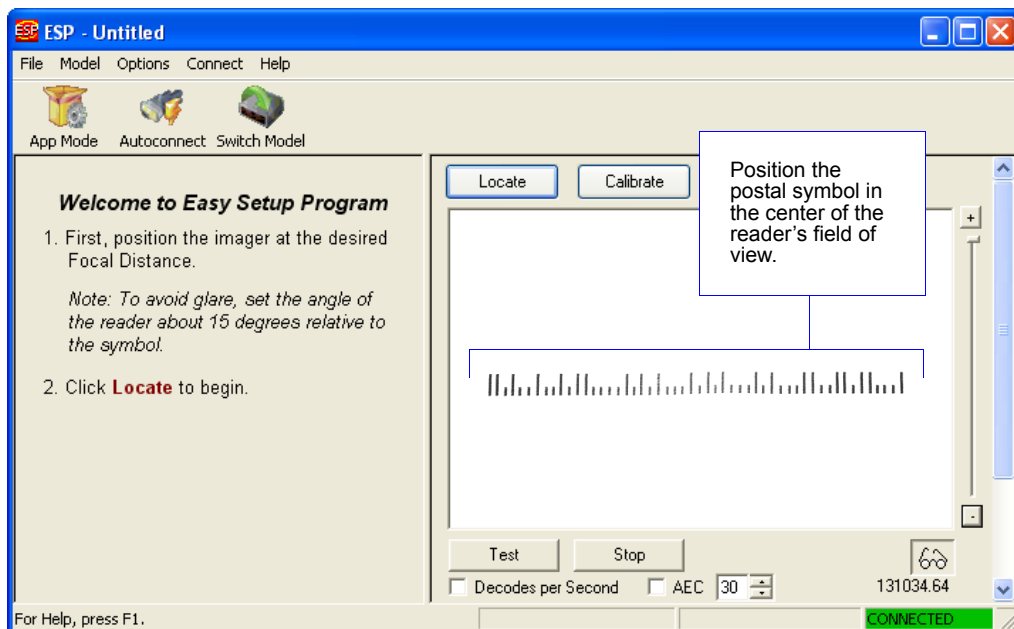
**Important: Postal Symbologies** must have a minimum pixels-per-element value of **4** to be decoded reliably by the MicroHAWK ID.

The reader must be configured to specific read range, field of view, and camera parameters before decoding Postal Symbologies.

**For optimal decode results with a Standard Density (SD) reader:**

- Set **Focal Distance** to **6"** and set gain and exposure settings for a high contrast image.
- Position the reader **3.5"** to **6"** from the symbol.
- Position the symbol as close to the center of the reader's field of view as possible.

**Note:** Measurements based on a 16 mil (60 dpi) POSTNET symbol.



## **Postal Symbology Type**

**Usage:** The following 1D **Postal Symbologies** are used in mail sortation, auditing, certified mail, registered mail, metered mail, and point-of-sale (POS) applications.

**Definition:** Determines the postal symbology that will be decoded by the reader.

**Serial Cmd:** <**K460,postal symbology type,POSTNET status,PLANET status,USPS4CB status,POSTNET allow B and B' fields,Australia Post allow 0 FCC**>

**Default:** **Disabled**

**Options:** **0 = Disabled**

1 = U.S. Post (POSTNET, PLANET, USPS4CB)

2 = Australia Post

3 = Japan Post

4 = Royal Mail

5 = KIX

6 = UPU

## U.S. Post (POSTNET, PLANET, USPS4CB)

When **U.S. Post** is enabled (<**K460,1**>), the reader will only decode **POSTNET**, **PLANET**, and **USPS4CB** symbols.

**Important:** **POSTNET Status**, **PLANET Status**, and **USPS4CB Status** are enabled by default. However, if any of the three U.S. Post symbologies is set to **disabled** individually, symbols of that type will not be decoded by the reader even when U.S. Post is enabled.

For example, if **U.S. Post** is enabled but **POSTNET Status** is disabled (<**K460,1,0**>), **POSTNET** symbols will not be decoded by the reader.

See **POSTNET Status**, **PLANET Status**, and **GS1 DataBar** for more detail about U.S. Post symbologies.

## Australia Post

When **Australia Post** is enabled (<**K460,2**>), the reader will only decode Australia Post symbols.

## Japan Post

When **Japan Post** is enabled (<**K460,3**>), the reader will only decode Japan Post symbols.

## Royal Mail

When **Royal Mail** is enabled (<**K460,4**>), the reader will only decode Royal Mail symbols.

## KIX

When **KIX** is enabled (<**K460,5**>), the reader will only decode KIX symbols.

## UPU

When **UPU** is enabled, the reader will decode UPU symbols.

For example, if **Postal Symbology Type** is set to **UPU** and **POSTNET Status** is enabled and (<**K460,6,1**>), the reader will attempt to decode both UPU and POSTNET symbols.

## **POSTNET Status**

**Usage:** **POSTNET** is used by the United States Postal Service to direct mail. The ZIP Code or ZIP+4 Code is encoded in the symbol. Data is encoded in half-height and full-height bars, making POSTNET a “2-state” symbology. The delivery point (usually the last two digits of the address or post office box number) is also typically encoded in POSTNET symbols.

**Definition:** If **U.S. Post** and **POSTNET Status** are both enabled, the reader will decode POSTNET symbols.

**Serial Cmd:** <**K460**,postal symbology type,**POSTNET status**,**PLANET status**,  
USPS4CB status,POSTNET allow B and B' fields,Australia Post allow 0  
FCC>

**Default:** **Enabled**

**Options:** 0 = Disabled

**1 = Enabled**

## **PLANET Status**

**Usage:** **PLANET (Postal Alphanumeric Encoding Technique)** is a symbology used by the United States Postal Service to track and identify items during delivery. Each PLANET symbol is either 12 or 14 digits long, and encodes data in half-height and full-height bars, making PLANET a “2-state” symbology. The symbol always starts and ends with a full-height bar, or “guard rail”, and each individual digit is represented by a set of five bars in which two of the bars are always short.

**Definition:** If **U.S. Post** and **PLANET Status** are both enabled, the reader will decode PLANET symbols.

**Serial Cmd:** <**K460**,postal symbology type,POSTNET status,**PLANET status**,  
USPS4CB status,POSTNET allow B and B' fields,Australia Post allow 0  
FCC>

**Default:** **Enabled**

**Options:** 0 = Disabled

**1 = Enabled**



## USPS4CB Status

**Usage:** **USPS4CB**, also called **Intelligent Mail**, is used by the United States Postal Service to sort and track individual items as well as flats of mail. USPS4CB combines the capabilities of POSTNET and PLANET, and can encode 31 digits (65 bars). USPS4CB symbols are slightly longer than POSTNET symbols, and offer additional flexibility in choosing symbol height and width. Data is encoded in four types of bars ("states"), each of which is identified by a name and a value. This type of postal symbol is known as "4-state". Each bar has a "tracker", or middle section, to which an "ascender" (top section) or "descender" (bottom section) may be added. The 4-state format allows the symbol to contain more information, and makes it easier to decode. 4-state symbols can also be printed easily in a variety of media, including dot matrix, inkjet, and laser.

**Definition:** If **U.S. Post** and **USPS4CB Status** are both enabled, the reader will decode USPS4CB symbols.

**Serial Cmd:** <**K460**,postal symbology type,POSTNET status,PLANET status,**USPS4CB status**,POSTNET allow B and B' fields,Australia Post allow 0 FCC>

**Default:** **Enabled**

**Options:** 0 = Disabled  
1 = **Enabled**

## POSTNET Allow B and B' Fields

**Definition:** If **U.S. Post** and **POSTNET** are enabled, and **POSTNET Allow B and B' Fields** is enabled, the reader will allow B and B' fields in POSTNET symbols.

**Serial Cmd:** <**K460**,postal symbology type,POSTNET status,PLANET status,USPS4CB status,**POSTNET allow B and B' fields**,Australia Post allow 0 FCC>

**Default:** **Disabled**

**Options:** 0 = **Disabled**  
1 = Enabled

## Australia Post Allow 0 FCC

**Definition:** If **Australia Post** is enabled, and **Australia Post Allow 0 FCC** is enabled, the reader will allow 0 FCC in Australia Post symbols.

**Serial Cmd:** <**K460**,postal symbology type,POSTNET status,PLANET status,USPS4CB status,POSTNET allow B and B' fields,**Australia Post allow 0 FCC**>

**Default:** **Disabled**

**Options:** 0 = **Disabled**  
1 = Enabled

# GS1 DataBar

## ***DataBar Expanded***

**Usage:** Used to encode primary and supplementary data in retail point-of-sale and other applications.

**Definition:** DataBar Expanded is a variable length symbology that can encode supplementary information in addition to the 14-digit EAN item identification number and is capable of encoding up to 74 numeric or 41 alphabetic characters.

**Serial Cmd:** <**K484**,*status*,*fixed symbol length status*,*fixed symbol length*>

**Default:** **Disabled**

**Options:** **0 = Disabled**      1 = Enabled

Where appropriate, use 1 (non-stacked) for better performance over 2 (stacked and non-stacked).

## **Fixed Symbol Length Status (DataBar Expanded)**

**Definition:** When enabled, the reader will check the symbol length against the symbol length field, minus the embedded check character. If disabled, any length would be considered valid.

**Serial Cmd:** <**K484**,*status*,***fixed symbol length status***,*fixed symbol length*>

**Default:** **Disabled**

**Options:** **0 = Disabled**      1 = Enabled

## **Fixed Symbol Length (DataBar Expanded)**

**Usage:** **Fixed Symbol Length** helps prevent truncations and increases data integrity by ensuring that only one symbol length will be accepted.

**Definition:** Specifies the exact number of characters that the reader will recognize (this does not include start, stop, and check character characters). The reader ignores any symbol not having the specified length.

**Serial Cmd:** <**K484**,*status*,*fixed symbol length status*,***fixed symbol length***>

**Default:** **14**

**Options:** 1 to 74

## ***DataBar Limited***

**Usage:** DataBar Limited is designed to be read by laser and CCD readers. It is not recommended for omnidirectional slot scanners.

**Definition:** Encodes a smaller 14-digit symbol (74 modules wide) that is not omnidirectional.

**Serial Cmd:** **<K483,status>**

**Default:** **Disabled**

**Options:** **0 = Disabled**      1 = Enabled

## ***DataBar-14***

**Usage:** Used in the grocery, retail, and prescription drug industries where 14-digit EAN item identification may be needed.

**Definition:** DataBar-14 is a fixed symbol length symbology that encodes 14 digits, including a 1-digit indicator. DataBar-14 is 96 modules wide. It can be stacked in two rows, it can read omnidirectionally if printed in full height, or horizontally if height-truncated for small marking.

**Serial Cmd:** **<K482,status>**

**Default:** **Disabled**

**Options:** **0 = Disabled**      1 = Enabled

## PDF417

**Usage:** Used in applications where a large amount of information (over 32 characters) needs to be encoded within a symbol, typically where the symbol is transported from one facility to another. For example, an automobile assembly line might use a single symbol with multiple fields of information that will be read at several stations along the way, without reference to a database.

**Definition:** A two-dimensional, multi-row (3 to 90), continuous, variable length symbology that has high data capacity for storing up to 2,700 numeric characters, 1,800 printable ASCII characters, or 1,100 binary characters per symbol. Each symbol character consists of 4 bars and 4 spaces in a 17-module structure.

**Serial Cmd:** **<K476,status,[unused],fixed symbol length status,fixed symbol length>**

**Default:** **Enabled**

**Options:** 0 = Disabled                      **1 = Enabled**

**Note:** Sending **<a1>** will cause PDF417 data to be prefaced with information consisting of error correction level (ECC Level **n**), number of rows (**n** Rows), number of columns (**n** Columns), number of informative code words (**n** Info Code Words) and the number of data characters (**n** Data Characters). This feature can be disabled by re-sending **<a1>**.

### **Fixed Symbol Length Status (PDF417)**

**Serial Cmd:** **<K476,status,[unused],fixed symbol length status,fixed symbol length>**

**Default:** **Disabled**

**Options:** **0 = Disabled**                      1 = Enabled

### **Fixed Symbol Length (PDF417)**

**Usage:** Used to increase data integrity by ensuring that only one symbol length will be accepted.

**Definition:** When enabled, the PDF symbol must contain the same number of characters as the symbol length setting before it can be considered a good read. The reader will ignore any symbol not having the specified length.

**Serial Cmd:** **<K476,status,[unused],fixed symbol length status,fixed symbol length>**

**Default:** **10**

**Options:** 1 to 2710

**Note:** **Fixed Symbol Length Status** must be enabled for **Fixed Symbol Length** to take effect.

# MicroPDF417

- Usage:** Used for labelling small items that need large data capacity.
- Definition:** A variant of PDF417, a very efficient and compact stacked symbology that can encode up to 250 alphanumeric characters or 366 numeric characters per symbol.
- Serial Cmd:** **<K485,status,[unused],fixed symbol length status,fixed symbol length>**
- Default:** **Disabled**
- Options:** **0 = Disabled**      1 = Enabled

## ***Fixed Symbol Length Status (MicroPDF417)***

- Serial Cmd:** **<K485,status,[unused],fixed symbol length status,fixed symbol length>**
- Default:** **Disabled**
- Options:** **0 = Disabled**      1 = Enabled

## ***Fixed Symbol Length (MicroPDF417)***

- Usage:** Used to increase data integrity by ensuring that only one symbol length will be accepted.
- Definition:** When enabled, the MicroPDF417 symbol must contain the same number of characters as the symbol length setting before it can be considered a good read. The reader will ignore any symbol not having the specified length.
- Serial Cmd:** **<K485,status,[unused],fixed symbol length status,fixed symbol length>**
- Default:** **10**
- Options:** 1 to 366

**Note:** **Fixed Symbol Length Status** must be enabled for **Fixed Symbol Length** to take effect.

## Composite

When set to **Enabled** or **Required**, will decode the 2D composite component of a linear symbol. The linear symbol can be DataBar-14, DataBar Expanded, DataBar Limited, EAN-128, UPC-A, EAN-13, EAN-8, and UPC-E.

*Usage:* Allows reading by both linear and 2D readers.

*Definition:* Combines 2D and linear width-modulated symbology on the same symbol where different messages can be read by each reader type.

*Serial Cmd:* <**K453**,*mode*,*separator status*,*separator*>

*Default:* **Disabled**

*Options:* **0 = Disabled**                      1 = Enabled                      2 = Required

### Enabled

If **Composite** is set to **Enabled**, the reader will decode both the 2D composite and linear components. However, if the 2D composite component is not decoded, the linear data will be sent by itself at the end of the read cycle.

### Required

If set to **Required**, the reader must decode both components, or a No Read will occur.

### Separator Status (Composite)

*Usage:* Allows the user to distinguish between the main and **Supplemental** symbols.

*Definition:* Separates the linear and the composite component.

*Serial Cmd:* <**K453**,*mode*,**separator status**,*separator*>

*Default:* **Disabled**

*Options:* **0 = Disabled**                      1 = Enabled

### Separator Character (Composite)

**Note:** The Separator Character will be the same as the character defined in the **Multisymbol Separator** field of the <**K222**> command.

*Usage:* As required by the application.

*Definition:* Allows the user to change the separator character from a comma to a new character.

*Serial Cmd:* <**K453**,*mode*,*separator status*,**separator**>

*Default:* , (comma)

*Options:* Any ASCII character.

**Important:** The ASCII characters <, >, and , can only be entered as hex pairs:

For <: <**K453h**,,,**3C**>

For >: <**K453h**,,,**3E**>

For , : <**K453h**,,,**2C**>

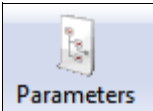
# 7 I/O Parameters

## Contents

I/O Parameters by ESP .....	7-2
I/O Parameters Serial Commands .....	7-3
Symbol Data Output .....	7-4
No Read Message .....	7-7
Read Duration Output .....	7-8
Output Indicators .....	7-9
Beeper .....	7-13
LED Configuration .....	7-14
Serial Verification .....	7-15
Configurable Output 1 .....	7-17
Trend Analysis Output 1 .....	7-20
ISO/IEC 16022 Symbol Quality Output 1 .....	7-23
Diagnostic Output 1 .....	7-26
Configurable Output 2 .....	7-27
Trend Analysis Output 2 .....	7-27
ISO/IEC 16022 Symbol Quality Output 2 .....	7-27
Diagnostic Output 2 .....	7-27
Configurable Output 3 .....	7-28
Trend Analysis Output 3 .....	7-28
ISO/IEC 16022 Symbol Quality Output 3 .....	7-28
Diagnostic Output 3 .....	7-28
Power-On/Reset Counts .....	7-29
Time Since Reset .....	7-30
Service Message .....	7-31
Frame Information .....	7-32
Image Output .....	7-33
Database Identifier Output .....	7-36
Quality Output .....	7-37

This section includes instructions on setting up conditions for changing input/output electrical transitions for control of the reader's internal and external devices. A discrete I/O (in/out) signal is an electrical transition from one voltage level to another so that digital switching can occur.

# I/O Parameters by ESP



Click the **Parameters** button and then the **I/O** tab.

Parameters	ESP Values
<div><div><div></div></div><div>I/O Parameters</div></div>	
<div><div><div></div></div><div>Symbol Data Output</div></div>	Good Read
<div><div><div></div></div><div>No Read Message</div></div>	Enabled
<div><div><div></div></div><div>Read Duration Output</div></div>	Disabled
<div><div><div></div></div><div>Output Object Info</div></div>	Disabled
<div><div><div></div></div><div>Output Indicators</div></div>	
<div><div><div></div></div><div>Serial Verification</div></div>	
<div><div><div></div></div><div>EZ Button</div></div>	Enabled
<div><div><div></div></div><div>Calibration Options</div></div>	
<div><div><div></div></div><div>Database Identifier Output</div></div>	Disabled*
<div><div><div></div></div><div>Output 1 Parameters</div></div>	Disabled*
<div><div><div></div></div><div>Output 2 Parameters</div></div>	Enabled
<div><div><div></div></div><div>Output 3 Parameters</div></div>	
<div><div><div></div></div><div>Quality Output</div></div>	
<div><div><div></div></div><div>Quality Output Separat...</div></div>	
<div><div><div></div></div><div>Decodes/Trigger Status</div></div>	Disabled

To open nested options, **single-click** the **+**.

To change a setting, **double-click** the setting and use your cursor to scroll through the options.



# I/O Parameters Serial Commands

Power On/Reset Counts	<K406,power-on,resets,power-on saves,customer default saves>
Time Since Reset (Read-Only)	<K407,hours,minutes>
Service Message	<K409,status,service message,threshold,resolution>
User-Defined Name	<K412,user-defined name>
Serial Verification	<K701,serial command echo status,serial command beep status,control/hex output>
Beeper	<K702,status>
Quality Output Separator	<K704,quality output separator,decodes per trigger status>
Symbol Data Output	<K705,status,when to output,symbology identifier status>
Read Duration Output	<K706,status,separator>
No Read Message	<K714,status,message>
Object Output Information	<K734,output frame number status,output coordinates status>
LED Configuration	<K737,ISO/IEC 16022 grade>
Green Flash and Target LEDs	<K750,green flash LED status,target pattern LED status,green flash duration>
Database Identifier Output	<K759,status,separator character>
Trend Analysis Output 1	<K780,trend analysis mode,number of triggers,number to output on,decodes per trigger>
Trend Analysis Output 2	<K781,trend analysis mode,number of triggers,number to output on,decodes per trigger>
Trend Analysis Output 3	<K782,trend analysis mode,number of triggers,number to output on,decodes per trigger>
Diagnostic Output 1	<K790,service unit message status>
Diagnostic Output 2	<K791,service unit message status>
Diagnostic Output 3	<K792,service unit message status>
ISO/IEC 16022 Symbol Quality Output 1	<K800,output on symbol contrast,symbol contrast threshold,output on print growth,print growth threshold,output on axial non-uniformity,axial non-uniformity threshold,output on Unused ECC,Unused ECC threshold>
ISO/IEC 16022 Symbol Quality Output 2	<K801,output on symbol contrast,symbol contrast threshold,output on print growth,print growth threshold,output on axial non-uniformity,axial non-uniformity threshold,output on Unused ECC,Unused ECC threshold>
ISO/IEC 16022 Symbol Quality Output 3	<K802,output on symbol contrast,symbol contrast threshold,output on print growth,print growth threshold,output on axial non-uniformity,axial non-uniformity threshold,output on Unused ECC,Unused ECC threshold>
Output 1 Parameters	<K810,output on,output state,pulse width,output mode>
Output 2 Parameters	<K811,output on,output state,pulse width,output mode>
Output 3 Parameters	<K812,output on,output state,pulse width,output mode>

# Symbol Data Output

## Symbol Data Output Status

**Note:** **Symbol Data Output** relates to data and should not be confused with **Outputs 1, 2, and 3** listed in the **Output Parameters** which describe output states and functions.

*Usage:* Useful when the host needs symbol data only under certain conditions.

*Definition:* Defines the conditions under which decoded symbol data is transmitted to the host.

*Serial Cmd:* <**K705,symbol data output status,when to output**>

*Default:* **Good Read**

*Options:* 0 = Disabled  
1 = Match  
2 = Mismatch  
**3 = Good Read**

**Note:** **Symbol Data Output Status**, if set to **Match** or **Mismatch**, will not take effect unless **Matchcode Type** is enabled and a master symbol is loaded into memory.

## Disabled

*Usage:* It is useful when an application only needs to use the discrete outputs and can allow the reader to do the decision-making. When **Disabled**, the host does not need the symbol data and the communication lines are used only for setup and status checks.

*Definition:* When set to **Disabled**, the reader will not transmit any data that is generated during a read cycle (symbols, No Reads, etc.)

## Match

*Usage:* **Match** is used in an application that requires specific symbol information and needs to sort, route, or verify based on matching the specific symbol data.

*Definition:* When set to **Match**, the reader transmits symbol data whenever a symbol matches a master symbol. However, if **Matchcode Type** is **Disabled**, it transmits on any good read.

**Note:** A No Read can still be transmitted if **Enabled**.

## Mismatch

*Usage:* Mismatch is typically used as a flag within the host system to prevent an item from being routed in the wrong container.

*Definition:* With Mismatch enabled, the reader transmits symbol data whenever the symbol data information does NOT match the master symbol.

**Note:** A No Read can still be transmitted if **Enabled**.

## **Good Read**

*Usage:* **Good Read** is used when an application requires all symbol data to be transmitted. It's typically used in tracking applications in which each object is uniquely identified.

*Definition:* With **Good Read** enabled, the reader transmits symbol data on any good read regardless of **Matchcode Type** setting.

**Note:** A No Read can still be transmitted if **Enabled**.

## When to Output Symbol Data

**Definition:** This command allows the user to choose when symbol data can be sent to the host.

**Serial Cmd:** <**K705**,symbol data output status,**when to output**>

**Default:** **As Soon As Possible**

**Options:** **0 = ASAP**                      1 = End of Read Cycle

### As Soon As Possible

**Usage:** **As Soon As Possible** is useful in applications in which symbol data needs to be moved quickly to the host, typically when the host is making decisions based on symbol data.

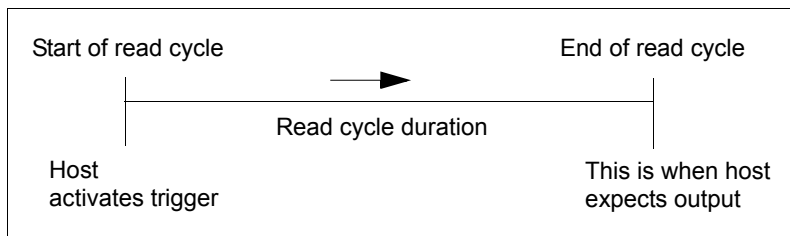
**Definition:** Enabling **As Soon As Possible** causes symbol data to be sent to the host immediately after a symbol has been successfully decoded.

**Note:** More than one decode might in fact be required to qualify as a good read, depending on how **Decodes Before Output** is set.

### End of Read Cycle

**Usage:** **End of Read Cycle** is useful in timing-based systems in which the host is not ready to accept data at the time that it is decoded.

**Definition:** Enabling **End of Read Cycle** means that symbol data does not get sent to the host until the read cycle ends with a timeout or new trigger.



# No Read Message

**Usage:** Used in applications where the host needs serial verification that a symbol has not been read and especially useful in new print verification.

**Definition:** When enabled, and if no symbol has been decoded before timeout or the end of the read cycle, the No Read message will be transmitted to the host.

## No Read Message Mode

**Serial Cmd:** <K714,No Read message status,No Read message>

**Default:** Enabled

**Options:** 0 = Disabled                      1 = Enabled

### Disabled

Only symbol data is output after a read cycle.

### Enabled

When the reader is in a triggered mode, a No Read message will be appended for each failed read attempt.

## No Read Message

**Definition:** Any combination of ASCII characters can be defined as the No Read message.

**Serial Cmd:** <K714,No Read message status,**No Read message**>

**Default:** NOREAD

**Options:** A string of up to 64 characters.

**Note:** No Read Message will only be transmitted if **Symbol Data Output** is set to **Match**, **Mismatch**, or **Good Read**.

**No Read Message** can be set to any ASCII character.

**Important:** The ASCII characters <, >, and , can only be entered as hex pairs:

For <: <K714h,,3C>

For >: <K714h,,3E>

For , : <K714h,,2C>

## Read Duration Output

**Usage:** Useful in evaluating actual read cycle timing results, especially when initially setting up an application to determine maximum line speed (obtainable based on spacing between symbols.)

**Definition:** When enabled the duration of the read cycle (in milliseconds) is appended to the symbol data.  
The read duration is the time from the beginning of the read cycle until data is output.

### Read Duration Output Mode

**Serial Cmd:** <K706,status,separator>

**Default:** Disabled

**Options:** 0 = Disabled      1 = Enabled

**Important:** To measure the entire read cycle when in **External Level** trigger mode, set **When to Output Symbol Data to End of Read Cycle**.

This output can measure over 49 days' worth of duration; if exceeded, the "OVERFLOW" message will be output in place of the duration.

### Read Duration Output Separator

**Definition:** User defined character that separates the symbol information from the **Read Duration Output**.

**Serial Cmd:** <K706,status,separator>

**Default:** [space character]

**Options:** Any ASCII character.

**Important:** The ASCII characters <, >, and , can only be entered as hex pairs:

For <: <K706h,,3C>

For >: <K706h,,3E>

For , : <K706h,,2C>

# Output Indicators

The MicroHAWK ID has a beeper and three LED arrays, as follows:

1. A target pattern of blue LEDs for centering the field of view, which is projected from the front of the reader .
2. An array of green LEDs projected from the front of the reader that can be programmed to flash in response to user-defined conditions.
3. A row of five status LEDs on the side of the reader.

## Green Flash Mode

**Usage:** Used as a visual verification that a good read has occurred.

**Definition:** An array of green LEDs in the front of the reader can be programmed to flash in response to user-defined conditions.

**Serial Cmd:** <**K750,green flash mode,target pattern status,green flash duration**>

**Default:** **Good Read**

**Options:**

- 0 = Disabled
- 1 = Good Read**
- 2 = Static Presentation
- 3 = Match
- 4 = Mismatch
- 5 = Strobe

## Disabled

Green flash LEDs are disabled.

## Good Read

Green flash LEDs will flash when a good read condition is met or when **Matchcode** is enabled and a match occurs.

## Static Presentation Mode

**Static Presentation Mode** is used in conjunction with **Continuous Read Mode**: <K200,0>.

When operating in **Static Presentation Mode**, the red LEDs will illuminate while the reader is searching for a symbol in **Continuous Read Mode**. When a symbol is placed in the field of view and a good read occurs, the green LEDs will illuminate and stay on for the duration of time set in **Green Flash Duration**. Only one read will occur during that time unless more than one symbol is enabled in **Number of Symbols**.

**Note:** If **Static Presentation Mode** is selected but the reader is not in **Continuous Read**, the **Green Flash** will not occur.

### To use Static Presentation:

1. Enable **Continuous Read**.
2. Select the number of symbols.
3. Enable **Static Presentation** in **Green Flash Mode**.
4. Select the read time in **Green Flash Duration**.

## Match

The green LEDs will flash when a match condition is met. If multisymbol is enabled, then green flash LEDs will illuminate only if all symbols qualify as a match. If matchcode is disabled, then this mode will activate the LEDs on a good read.

## Mismatch

Same as **Match**, except that LEDs will illuminate on a mismatch.

## Strobe

Green flash LEDs will act as an illumination strobe for image capture. If it is required that the green flash LEDs be the *only* illumination for image capture, then the internal illumination LEDs can be disabled.



## Target Pattern

- Usage:** Assists users in positioning and locating symbols in the center of the reader's field of view.
- Definition:** The user can control when the targeting system is ON or OFF, and can save this condition for power-on.
- Serial Cmd:** <K750,green flash mode,**target pattern status**,green flash duration>
- Default:** **Always ON**
- Options:**
- 0 = Always OFF
  - 1 = ON only when not in the read cycle
  - 2 = ON only when in the read cycle
  - 3 = Always ON**

### Always OFF

The target pattern will remain OFF at all times unless overridden by an operational command.

### ON Only When Not in the Read Cycle

The target pattern is always ON except during the read cycle. If the operational command overrides this setting, the target pattern will remain on at all times.

### ON Only When in the Read Cycle

The target pattern will remain OFF except during the read cycle. If the operational command overrides this setting, the target pattern will remain on at all times.

### Always ON

The target pattern is always ON.

## **Green Flash Duration**

*Usage:* Provides visual verification that a good read has occurred.

*Definition:* When a good read occurs, the green LEDs will illuminate and stay on for the time set in the **Green Flash Duration** value.

*Serial Cmd:* <**K750**,green flash mode,target pattern status,**green flash duration**>

*Default:* **100** (1 second)

*Options:* 0 to 65535 (in 10 ms increments)

# Beeper

*Usage:* An audible verification that either a good read or a No Read has occurred.

*Definition:* A beep is emitted after each good read or No Read.

*Serial Cmd:* <K702,beeper status>

*Default:* **Good Read**

*Options:* 0 = Disabled      **1 = Good Read**      2 = No Read

**Note:** The beeper will also sound at any of the following times:

- When the reader is defaulted.
- When a send/save command is sent from **ESP**.
- At the conclusion of a calibration procedure.
- When a <Z>, <Zp>, <Zrd>, or <K701,,1> command is sent.

# LED Configuration

*Usage:* Useful as a visible indicator of read rates and symbol quality.

*Definition:* Determines the mode in which the status LEDs operate.

*Serial Cmd:* <**K737,LED mode,ISO/IEC 16022 grade**>

*Default:* **Standard**

*Options:* **0 = Standard**      1 = ISO/IEC 16022 grade

In ISO/IEC 16022 grade mode the LEDs represent the grade of the first Data Matrix symbol decoded in the read cycle.

The parameter to be graded is set in the **ISO/IEC 16022 Grade** options.

For example, to program the LEDs to indicate the ISO/IEC 16022 print growth grade, set **LED Mode** to **ISO/IEC 16022 Grade** and set **ISO/IEC 16022 Grade** to **Print Growth**. If all the LEDs from 20% to 100% are illuminated, the read result is a grade A; if only the 20% LED is illuminated, the result is a grade F.

## Standard

In **Standard Mode**, the STATUS LED indicates read cycle status and the GOOD READ LED illuminates upon a good read at the end of a read cycle. In a read rate test, these LEDs represent the percentage of good reads per images captured.

## ISO/IEC 16022 Grade

*Usage:* Provides visual grading of specific ISO/IEC 16022 parameters.

*Definition:* Determines which ISO/IEC 16022 parameter the reader will grade via the LEDs.

*Serial Cmd:* <**K737,LED mode,ISO/IEC 16022 grade**>

*Default:* **Final Grade**

*Options:* **0 = Final Grade**  
1 = Symbol Contrast  
2 = Print Growth  
3 = Axial Non-Uniformity  
4 = Unused ECC

# Serial Verification

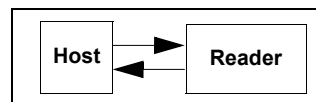
Allows the user to verify configuration command status.

## Serial Command Echo Status

**Usage:** This command is useful in removing any doubt about the reader's interpretation of any configuration command.

For example, if the current preamble is "SOM" and **<K701,1,START>** is entered, the reader will echo back **<K701,SOM>** since the attempted entry "START" exceeds the four character limit for that command. Therefore, it is rejected and the existing "SOM" message is echoed back and remains the preamble message.

**Definition:** When enabled, a configuration command received from the host is echoed back to the host with the resultant settings.



**Function:** If a command with multiple fields is processed, some of the fields may have been processed properly while others were not. The changes will appear in the string echoed back so that the user will know which fields did or did not change.

**Serial Cmd:** **<K701,serial command echo status,serial command beep status,control/hex output>**

**Default:** **Disabled**

**Options:** **0 = Disabled**      1 = Enabled

## Serial Command Beep Status

**Usage:** Used to audibly verify the acceptance and validity of a command.

**Definition:** Causes the reader to beep once whenever a K command is entered to indicate that the command was accepted and processed.

**Function:** If an invalid command is entered, the reader beeps 5 times to indicate an invalid entry. However, this does not necessarily mean that all data fields have been entered incorrectly. Only one bad field needs to be found in order to activate the 5 beep response.

**Serial Cmd:** **<K701,serial command echo status,serial command beep status,control/hex output>**

**Default:** **Disabled**

**Options:** **0 = Disabled**      1 = Enabled

## Control/Hex Output

**Usage:** Useful for viewing settings with binary characters when using serial commands on a terminal.

**Definition:** Determines the response to a **Serial Command Echo** or status request command.

When set to **Control**, two characters are transmitted to represent a non-displayable character. For example, a carriage return will be shown as: **^M**.

**Serial Cmd:** **<K701,serial command echo status,serial command beep status,control/hex output>**

**Default:** **Control**

**Options:** **0 = Control**                      1 = Hex

# Configurable Output 1

<input type="checkbox"/> Output 1 Parameters	
Output On	Trend Analysis
Output State	Normally Closed
Pulse Width	5
Output Mode	Pulse
<input checked="" type="checkbox"/> Trend Analysis	
<input checked="" type="checkbox"/> Symbol Quality (ISO/IEC 16022)	
<input type="checkbox"/> Diagnostics	

**Usage:** This option provides discrete signalling to host software to control external devices such as PLCs and relays. It is useful for routing, sorting, and to prevent mis-packaging and mis-routing.

**Definition:** Sets the discrete output functions for specific user-selected conditions. Allows the user to set the conditions under which an output (or outputs) will be activated.

**Serial Cmd:** `<K810,output on,output state,pulse width,output mode>`

**Default:** **Mismatch or No Read**

**Options:**

- 0 = Mismatch or No Read**
- 1 = Match (or Good Read)
- 2 = Mismatch
- 3 = No Read
- 4 = Trend Analysis
- 5 = Symbol Quality
- 6 = Diagnostic Warning
- 7 = In Read Cycle

**Note:** If **Output On** is set to **Match** or **Mismatch**, a transition (switching) will not occur unless **Matchcode Type** is enabled and a master symbol is loaded into memory.

## Mismatch or No Read

Activates discrete output when the data does not match that of the master symbol or the symbol has not been decoded before the end of the read cycle.

## Match (or Good Read)

Activates a discrete output when the symbol data matches the master symbol.

**Note:** If you want to output for a good read and **Matchcode** is not enabled, you can enable any output for **Match**.

## Mismatch

Activates a discrete output whenever the symbol data does not match that of the master symbol.

## No Read

Activates a discrete output whenever the symbol data is not decoded before the end of the read cycle.

## **Trend Analysis**

*Usage:* Typically used when successful decodes are occurring but a discrete output is needed to flag a trend in quality issues.

*Definition:* Activates discrete output when a trend analysis condition is met, depending on the trend analysis option enabled.

## **Symbol Quality**

*Usage:* Typically used when a discrete indication is needed to flag a symbol quality condition.

*Definition:* Activates discrete output when a symbol quality condition is met, depending on the symbol quality option enabled.

## **Diagnostic Warning**

*Usage:* Typically used when a discrete indication of a diagnostic condition is needed.

*Definition:* Activates discrete output when a diagnostic warning condition is met, depending on the diagnostic option enabled.

## **In Read Cycle**

*Definition:* Activates a discrete output when the reader is in a read cycle.



## Output State

**Definition:** Sets the active electrical state of the discrete output.

**Serial Cmd:** <**K810,output on,output state,pulse width,output mode**>

**Default:** **Negative**

**Options:** **0 = Normally Open**      1 = Normally Closed

## Pulse Width

**Definition:** Sets the time in 10 ms increments that the discrete output remains active.

**Serial Cmd:** <**K810,output on,output state,pulse width,output mode**>

**Default:** **5** (Corresponds to 50 ms)

**Options:** 0 to 255 (0 to 2.55 seconds). Divide the number entered on the command line by 100 for time in seconds.

## Output Mode

**Definition:** Sets the condition in which the discrete output is de-activated.

**Serial Cmd:** <**K810,output on,output state,pulse width,output mode**>

**Default:** **Pulse**

**Options:** **0 = Pulse**  
2 = Latch Mode 2  
3 = Latch Mode 3

## Pulse

This is the default mode of operation in which the programmable output is activated when the **Output On** condition has been met and held active for the duration of the selected pulse width.

### Latch Mode 2 (Unlatch Opposite Condition)

The programmable output is activated when the **Output On** condition has been met and held active until the opposite condition selected under **Output On** has been met.

For example, if **No Read** is enabled under **Output On**, the programmable output will go active on a No Read and remain active until the opposite condition, a good read, occurs.

### Latch Mode 3 (Unlatch Re-enter Read Cycle)

The programmable output is active when the Output On condition has been met and is held active until a new read cycle begins.

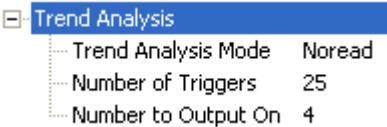
**Note:** All of the **Output On** modes are inhibited when any **Output on Warning** is active for **Output 1**.

# Trend Analysis Output 1

**Note:** **Output On** under **Output 1 Parameters** must be set to **Trend Analysis** for this output to function.



Under **Output 1 Parameters**, expand the **Trend Analysis** tree:



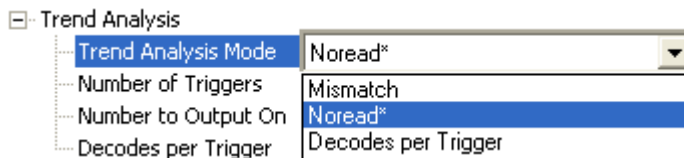
**Usage:** Useful in cases where the user doesn't want to shut down for one condition but wants to monitor quality and read conditions.

**Definition:** Applies **Trend Analysis** settings to **Output 1**.  
With **Trend Analysis**, the user can track the occurrences and frequency of mismatches, No Reads, and the number of reads per trigger, and output the results to any of three outputs.

**Example:** *Trend Analysis Mode = No Read*  
*Trigger Evaluation Period= 25 triggers (read cycles)*  
*Number to Output On = 4*  
In this example, the reader will activate an output when 4 No Reads occur within a period of 25 triggers (read cycles).

## Trend Analysis Mode

**Definition:** Sets the trend condition (**Mismatch**, **No Read**, or **Decodes per Trigger**) that will activate the output.



**Serial Cmd:** <K780,trend analysis mode,trigger evaluation period,number to output on,decodes per trigger threshold>

**Default:** No Read

**Options:**  
 0 = Mismatch  
 1 = No Read  
 2 = Decodes per Trigger

### Mismatch

Output will be activated when the number of mismatches equals the value entered for **Number to Output On** within the trigger window selected in **Number of Triggers**.

### No Read

Output will be activated when the number of No Reads equals the value entered for **Number to Output On** within the trigger window selected in **Number of Triggers**.

### Decodes per Trigger

Output will be activated when the number of decodes equals the value entered for **Number to Output On** within the trigger window selected in **Number of Triggers**.

## **Trigger Evaluation Period**

**Definition:** The number of triggers to examine for the trend analysis condition.

**Serial Cmd:** <**K780**,trend analysis mode,**trigger evaluation period**,number to output on,decodes per trigger threshold>

**Default:** 0

**Options:** 0 to 255

## **Number to Output On**

**Usage:** *Example:* If **Number to Output On** is set to 3 and **Trend Analysis Mode** is set to **No Read**, then the output will not be activated until 3 No Reads have occurred.

**Definition:** Sets the number of **Trend Analysis Mode** events (mismatches, No Reads, or reads/trigger as configured by Trend Analysis Mode) to occur within the trigger evaluation period before activating the associated output.

**Serial Cmd:** <**K780**,trend analysis mode,trigger evaluation period,**number to output on**,decodes per trigger threshold>

**Default:** 0

**Options:** 0 to 255

## **Decodes per Trigger Threshold**

**Definition:** The appropriate output will be activated if, at the end of the read cycle, the symbol decode count is below this setting.

**Note:** To activate this feature the reader must be in Decodes per Trigger mode. To put the reader in this mode during the read cycle and the Trend Analysis operation, the Decodes per Trigger status in the quality settings must be enabled. Enabling this setting will also append the decode count to the symbol data.

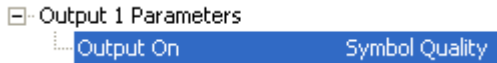
**Serial Cmd:** <**K780**,trend analysis mode,trigger evaluation period,number to output on,decodes per trigger threshold>

**Default:** 0

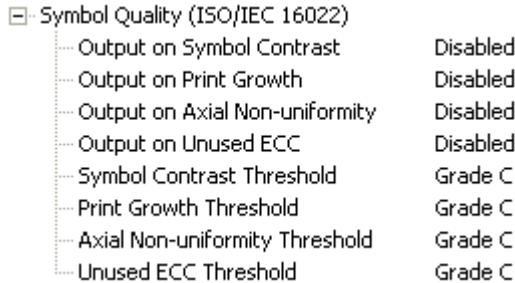
**Options:** 0 to 65535

# ISO/IEC 16022 Symbol Quality Output 1

**Note:** **Output On** under **Output 1 Parameters** must be set to **Symbol Quality** for this output to function.



Under **Output 1 Parameters**, expand the **Symbol Quality (ISO/IEC 16022)** tree.



## Output on Symbol Contrast

**Usage:** Lets the user know if symbol quality is less than acceptable.

**Definition:** If enabled, toggles Output 1 to an active state when **Symbol Contrast Threshold** is met.

**Serial Cmd:** <K800,output on symbol contrast,symbol contrast threshold,output on print growth,print growth threshold,output on axial non-uniformity,axial non-uniformity threshold,output on UEC,UEC threshold>

**Default:** **Disabled**

**Options:** **0 = Disabled**                      1 = Enabled

## Output on Print Growth

**Usage:** Lets the user know if symbol quality is less than acceptable.

**Definition:** If enabled, toggles **Output 1** to an active state when **Print Growth Threshold** is met.

**Serial Cmd:** <K800,output on symbol contrast,symbol contrast threshold,output on print growth,print growth threshold,output on axial non-uniformity,axial non-uniformity threshold,output on UEC,UEC threshold>

**Default:** **Disabled**

**Options:** **0 = Disabled**                      1 = Enabled

## Output on Axial Non-Uniformity

**Usage:** Lets the user know if symbol quality is less than acceptable.

**Definition:** If enabled, toggles **Output 1** to an active state when **Axial Non-Uniformity Threshold** is met.

**Serial Cmd:** <**K800**,output on symbol contrast,symbol contrast threshold,output on print growth,print growth threshold,**output on axial non-uniformity**,axial non-uniformity threshold,output on UEC,UEC threshold>

**Default:** **Disabled**

**Options:** **0 = Disabled** 1 = Enabled

## Output on Unused Error Correction

**Usage:** Lets the user know if symbol quality is less than acceptable.

**Definition:** If enabled, toggles **Output 1** to an active state when **UEC Threshold** is met.

**Serial Cmd:** <**K800**,output on symbol contrast,symbol contrast threshold,output on print growth,print growth threshold,output on axial non-uniformity,axial non-uniformity threshold,**output on UEC**,UEC threshold>

**Default:** **Disabled**

**Options:** **0 = Disabled** 1 = Enabled

## Symbol Contrast Threshold

**Usage:** Lets the user set the acceptable level of symbol quality.

**Definition:** Conforms to ISO/IEC 16022 symbol quality grading (A,B,C,D).

**Serial Cmd:** <**K800**,output on symbol contrast,**symbol contrast threshold**,output on print growth,print growth threshold,output on axial non-uniformity,axial non-uniformity threshold,output on UEC,UEC threshold>

**Default:** **Grade C**

**Options:** 0 = Grade A

1 = Grade B

**2 = Grade C**

3 = Grade D

## **Print Growth Threshold**

**Usage:** Lets the user set the acceptable level of symbol quality.

**Definition:** Conforms to ISO/IEC 16022 symbol quality grading (A,B,C,D).

**Serial Cmd:** *<K800,output on symbol contrast,symbol contrast threshold,output on print growth,print growth threshold,output on axial non-uniformity,axial non-uniformity threshold,output on UEC,UEC threshold>*

**Default:** **Grade C**

**Options:** 0 = Grade A  
1 = Grade B  
**2 = Grade C**  
3 = Grade D

## **Axial Non-Uniformity Threshold**

**Usage:** Lets the user set the acceptable level of symbol quality.

**Definition:** Conforms to ISO/IEC 16022 symbol quality grading (A,B,C,D).

**Serial Cmd:** *<K800,output on symbol contrast,symbol contrast threshold,output on print growth,print growth threshold,output on axial non-uniformity,axial non-uniformity threshold,output on UEC,UEC threshold>*

**Default:** **Grade C**

**Options:** 0 = Grade A  
1 = Grade B  
**2 = Grade C**  
3 = Grade D

## **Unused Error Correction Threshold**

**Usage:** Lets the user set the acceptable level of symbol quality.

**Definition:** Conforms to ISO/IEC 16022 symbol quality grading (A,B,C,D).

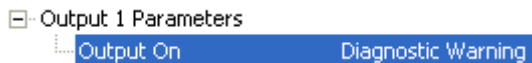
**Serial Cmd:** *<K800,output on symbol contrast,symbol contrast threshold,output on print growth,print growth threshold,output on axial non-uniformity,axial non-uniformity threshold,output on UEC,UEC threshold>*

**Default:** **Grade C**

**Options:** 0 = Grade A  
1 = Grade B  
**2 = Grade C**  
3 = Grade D

## Diagnostics Output 1

**Note:** **Output On** under **Output 1 Parameters** must be set to **Diagnostic Warning** for this output to function.



Under **Output 1 Parameters**, expand the **Diagnostics** tree.



**Serial Cmd:** <K790,unused 1,service unit>

**Default:** 0

**Options:** 0 = Disabled

### Service Unit

When **Diagnostic Warning** is enabled, the **Output On** configuration has no effect. The output will remain active as long as one of the diagnostic warning conditions is met. The output will become inactive once it detects that there are no diagnostic warning conditions.

**Definition:** Allows the user to set up the output to toggle to active when the service timer has expired. This condition will only be held for one service timer click.

**Note:** This feature cannot be used if the reader is in a **Continuous Read** mode.

**Serial Cmd:** <K790,unused 1,service unit>

**Default:** 0

**Options:** 0 = Disabled 1 = Enabled



## Configurable Output 2

**Note:** **Output 2** has the same parameters and default settings as **Output 1**.

[-] Output 2 Parameters	
[-] Output On	Symbol Quality
[-] Output State	Normally Closed
[-] Pulse Width	5
[-] Output Mode	Pulse
[+] Trend Analysis	
[+] Symbol Quality (ISO/IEC 16022)	
[+] Diagnostics	

*Serial Cmd: <K811,output on,output state,pulse width,output mode>*

## Trend Analysis Output 2

**Note:** **Output On** under **Output 2 Parameters** must be set to **Trend Analysis** for this output to function.

[-] Output 2 Parameters	
[-] Output On	Trend Analysis

*Serial Cmd: <K781,trend analysis mode,trigger evaluation period,number to output on>*

## ISO/IEC 16022 Symbol Quality Output 2

**Note:** **Output On** under **Output 2 Parameters** must be set to **Symbol Quality** for this output to function.

[-] Output 2 Parameters	
[-] Output On	Symbol Quality

*Serial Cmd: <K801,output on symbol contrast,symbol contrast threshold,output on print growth,print growth threshold,output on axial non-uniformity,axial non- uniformity threshold,output on UEC,UEC threshold>*

## Diagnostics Output 2

**Note:** **Output On** under **Output 2 Parameters** must be set to **Diagnostic Warning** for this output to function.

[-] Output 2 Parameters	
[-] Output On	Diagnostic Warning

*Serial Cmd: <K791,unused 1,service unit>*

# Configurable Output 3

**Note:** **Output 3** has the same parameters and default settings as **Output 1** and **Output 2**.

[-] Output 3 Parameters	
[-] Output On	Mismatch or Noread
[-] Output State	Normally Closed
[-] Pulse Width	5
[-] Output Mode	Pulse
[+] Trend Analysis	
[+] Symbol Quality (ISO/IEC 16022)	
[+] Diagnostics	

*Serial Cmd: <K812,output on,output state,pulse width,output mode>*

## Trend Analysis Output 3

**Note:** **Output On** under **Output 3 Parameters** must be set to **Trend Analysis** for this output to function.

[-] Output 3 Parameters	
[-] Output On	Trend Analysis

*Serial Cmd: <K782,trend analysis mode,trigger evaluation period,number to output on>*

## ISO/IEC 16022 Symbol Quality Output 3

**Note:** **Output On** under **Output 3 Parameters** must be set to **Symbol Quality** for this output to function.

[-] Output 3 Parameters	
[-] Output On	Symbol Quality

*Serial Cmd: <K802,output on symbol contrast,symbol contrast threshold,output on print growth,print growth threshold,output on axial non-uniformity,axial non-uniformity threshold,output on UEC,UEC threshold>*

## Diagnostics Output 3

**Note:** **Output On** under **Output 3 Parameters** must be set to **Diagnostic Warning** for this output to function.

[-] Output 3 Parameters	
[-] Output On	Diagnostic Warning

*Serial Cmd: <K792,unused 1,service unit>*

# Power On/Reset Counts

**Note:** **Power On/Reset Counts** is a read-only command.

## **Power-On**

*Serial Cmd:* <**K406**,**power-on**,**resets**,**power-on saves**,**power-on flash saves**>

**Power-On** uses a 16-bit counter that increments each time the reader is powered-on.

## **Resets**

*Serial Cmd:* <**K406**,**power-on**,**resets**,**power-on saves**,**power-on flash saves**>

**Resets** uses a 16-bit counter that increments each time the reader is reset. This value is reset at power-on.

## **Power-On Saves**

*Serial Cmd:* <**K406**,**power-on**,**resets**,**power-on saves**,**power-on flash saves**>

**Power-On Saves** uses a 16-bit counter that increments each time an reader setting is saved for power-on with a <**Z**> command.

## **Power-On Flash Saves**

*Serial Cmd:* <**K406**,**power-on**,**resets**,**power-on saves**,**power-on flash saves**>

**Power-On Flash Saves** uses a 16-bit counter that increments each time an reader setting is saved to the customer parameter section of flash memory with a <**Zc**> command.

## Time Since Reset

**Note:** **Time Since Reset** is a read-only command.

### ***Hours***

*Serial Cmd:* <**K407**,*hours*,*minutes*>

*Default:*       **16-bit counter** (0 to 65535)

**Hours** uses a 16-bit counter that increments every 60 minutes.

### ***Minutes***

*Serial Cmd:* <**K407**,*hours*,*minutes*>

*Default:*       **16-bit counter** (0 to 60)

**Minutes** uses a 16-bit counter that increments every 60 seconds.

**Note:** Time counts are reset at power-on, but not with an <**A**> or <**Z**> command.

## Service Message

When **Service Message** is enabled, a message of up to 10 ASCII characters is sent whenever the system detects that the service timer's limit has been reached. The service timer is reset at power-on, meaning that the service timer's limit is the amount of time since last reset. Service timer increments can be set in seconds or minutes.

*Serial Cmd:* <**K409**,*status*,*service message*,*threshold*,*resolution*>

*Default:* **0**

*Options:* **0 = Disabled**                      1 = Enabled

## Service Message

*Serial Cmd:* <**K409**,*status*,*service message*,*threshold*,*resolution*>

*Default:* **SERVICE**

*Options:* An ASCII string between 1 and 10 characters.

**Important:** The ASCII characters <, >, and , can only be entered as hex pairs:

For <: <**K409h**,**,3C**>

For >: <**K409h**,**,3E**>

For , : <**K409h**,**,2C**>

## Threshold

*Serial Cmd:* <**K409**,*status*,*service message*,*threshold*,*resolution*>

*Default:* **300** (5 minutes)

*Options:* 1 to 65535

## Resolution

*Serial Cmd:* <**K409**,*status*,*service message*,*threshold*,*resolution*>

*Default:* **Seconds**

*Options:* **0 = Seconds**                      1 = Minutes

## Frame Information

### *Output Frame Number*

*Serial Cmd:* <**K734**,*output frame number*,*output coordinates*>

*Default:*       **Disabled**

*Options:*       **0 = Disabled**                      1 = Enabled

#### **Disabled**

When **Output Frame Number** is set to **Disabled**, the frame number is not output as part of the symbol data.

#### **Enabled**

When **Output Frame Number** is set to **Enabled**, the number of the frame in which the symbol was first decoded is attached to the symbol output information. The output format is “F*nnn*”, where “*nnn*” is a 3-digit decimal value from 0 to 255. The frame number rolls over to 0 once it passes 255.

If this feature is enabled, the frame information will follow the symbol data, and it will precede any symbol quality data (if symbol quality settings are enabled—see **Symbol Quality Separator**). The symbol quality separator is located between the end of the already-formatted data and the “F” that precedes the frame number.

### *Output Coordinates*

*Serial Cmd:* <**K734**,*output frame number*,**output coordinates**>

*Default:*       **Disabled**

*Options:*       **0 = Disabled**                      1 = Enabled

#### **Disabled**

When **Output Coordinates** is set to **Disabled**, the coordinates are not output as part of the symbol data.

#### **Enabled**

When **Output Coordinates** is set to **Enabled**, the four sets of object coordinates for the first successfully decoded symbol are attached to the symbol data output in the following format:

“(*nnn*,*mmm*)”, where “*nnn*” is a 3-digit X-coordinate and “*mmm*” is a 3-digit Y-coordinate. The validation separator is located between the end of the already formatted data and the first coordinate point. The point with the lowest X + Y sum will be output first, followed by the next three points in a clockwise direction.

*Example:*

(032,040)(287,056)(287,279)(048,271)

# Image Output

- Usage:** This feature allows the user to select what read cycle condition triggers the image file output.
- Definition:** Used to output an image file when a specified read cycle condition has been met, such as No Read or Good Read. When the read cycle is complete, the normal read cycle output will occur, including any configured postamble or preamble data. The read cycle will then check if image output is enabled, and if the specified read cycle condition has been met. If both these conditions are true, the image file will be output to the user-specified communication port in the user-specified image file format.
- Serial Cmd:** **<K739,image output mode,communication port,file format,JPEG quality>**
- Default:** **Disabled**
- Options:** **0 = Disabled**  
 1 = Good Read  
 2 = No Read  
 3 = No Read or Good Read  
 4 = Mismatch

## Good Read

If a Good Read condition is met during the read cycle, the image file corresponding with the first Good Read will be output. The image file will immediately follow the read cycle output and postambles. For a Good Read condition to occur, all symbols in the read cycle must qualify.

## No Read

If a No Read condition occurs during the read cycle, the image file corresponding with the first No Read will output. The image file will immediately follow the read cycle output and postambles. There are conditions in which the read cycle may have a No Read condition but the image will *not* be output. For example, if the read cycle is looking for two No Read symbols in the same image capture but one is a Good Read, the image will not be output, because it is not considered a No Read image.

## No Read or Good Read

If a No Read condition or a Good Read condition occurs, the first image of either kind will be output.

## Mismatch

If a Good Read occurs during the read cycle but the image is a Mismatch condition, the first Mismatch image will be output.

## Communication Port

**Definition:** This setting selects the communication port through which the image file will be transferred to the host.

**Serial Cmd:** <K739,image output mode,**communication port**,file format,JPEG quality>

**Default:** Host

**Options:** 0 = Host 1 = TCP Port 1 2 = TCP Port 2

## File Format

**Definition:** This setting selects the format of the image file to be output.

Note: If JPEG is the chosen image file format, the resolution of the image will be determined by the setting in the **JPEG Quality** field.

**Serial Cmd:** <K739,image output mode,communication port,**file format**,JPEG quality>

**Default:** JPEG

**Options:** 0 = Bitmap 1 = JPEG 2 = Binary

## JPEG Quality

**Definition:** This setting determines the quality of JPEG image compression. The user may select a value from 1 (lowest) to 100 (highest).

**Serial Cmd:** <K739,image output mode,communication port,file format,**JPEG quality**>

**Default:** 90

**Options:** 1 to 100 (percent)



## ***Image Output Example***

The following example assumes that the reader is set to output on No Read, and that it is in a rapid capture mode of 3 captures.

***Number of Symbols:*** 1  
***Frame # 1:*** No Read  
***Frame # 2:*** No Read  
***Frame # 3:*** No Read  
***Read Cycle Result:*** No Read  
***Output Frame:*** **Frame # 1**

***Number of Symbols:*** 1  
***Frame # 1:*** No Read  
***Frame # 2:*** No Read  
***Frame # 3:*** Good Read  
***Read Cycle Result:*** Good Read  
***Output Frame:*** **None**

***Number of Symbols:*** 2  
***Frame # 1:*** No Read  
***Frame # 2:*** No Read  
***Frame # 3:*** Good Read, Symbol # 1  
***Read Cycle Result:*** No Read  
***Output Frame:*** **Frame # 1**

## Database Identifier Output

*Usage:* Useful in keeping track of which database entries read which symbols.

*Serial Cmd:* <K759,status,separator>

*Default:* 0

*Options:* 0 = Disabled 1 = Enabled

### Output Status

#### Disabled

When this command is disabled, no database identifier information will be output.

#### Enabled

When this command is enabled, the reader will append a two-digit number and the characters “DB” to the data output following the separator for each symbol decoded using **Configuration Database**. For example, if the separator is an underscore character and the second database entry reads a symbol encoded with “data capture” during the read cycle, the symbol data output will be “data capture\_DB02”. If the database is not active, no identifiers will be attached to output.

### Separator

*Usage:* The separator character separates the symbol data from the database identifier.

*Serial Cmd:* <K759,status,separator>

*Default:* <space>

*Options:* Any ASCII character.

**Important:** The ASCII characters <, >, and , can only be entered as hex pairs:

For <: <K759h,,3C>

For >: <K759h,,3E>

For , : <K759h,,2C>

# Quality Output

## Separator

**Definition:** The separator character separates quality output data from symbol data.

**Serial Cmd:** <K704,separator,decodes per trigger status>

**Default:** ,

**Options:** Any ASCII character.

**Important:** The ASCII characters <, >, and , can only be entered as hex pairs:

For <: <K704h,3C>

For >: <K704h,3E>

For , : <K704h,2C>

## Decodes per Trigger Status

**Definition:** When this feature is enabled, the reader enters a state where it processes frames as configured until the end of a read cycle, with or without a successful symbol decode. When the read cycle ends, the reader outputs any decoded symbol data along with the decodes per trigger count.

**Serial Cmd:** <K704,separator,decodes per trigger status>

**Default:** 0

**Options:** 0 = Disabled                      1 = Enabled



# 8 *Symbol Quality*

**Contents**

Symbol Quality by ESP..... 8-2

Symbol Quality Serial Commands ..... 8-3

Overview of Symbol Quality..... 8-4

Symbol Quality Separator/Data Matrix Output Mode..... 8-8

ISO/IEC 16022 Symbol Quality Output..... 8-10

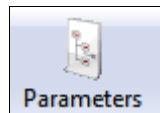
ISO/IEC 16022 Symbol Quality Output by ESP..... 8-12

Microscan Symbol Quality Output ..... 8-13

Microscan Symbol Quality Output by ESP ..... 8-16

This section describes parameters that, when enabled, will output detailed symbol quality evaluations.

# Symbol Quality by ESP



Click the **Parameters** button and then the **Symbol Quality** tab.

Parameters	ESP Values
[-] Symbol Quality	
[-] Global	
Total Read Time	Disabled
Symbol Quality Separator	SP
Output Mode	Grade
[-] Data Matrix	
[-] ISO/IEC 16022 Parameters	
Symbol Contrast	Disabled*
Print Growth	Disabled*
Axial Non-uniformity	Enabled
Unused ECC	Disabled
[+] Grading	

To open nested options, **single-click** the +.

To change a setting, **double-click** the setting and use your cursor to scroll through the options.

# Symbol Quality Serial Commands

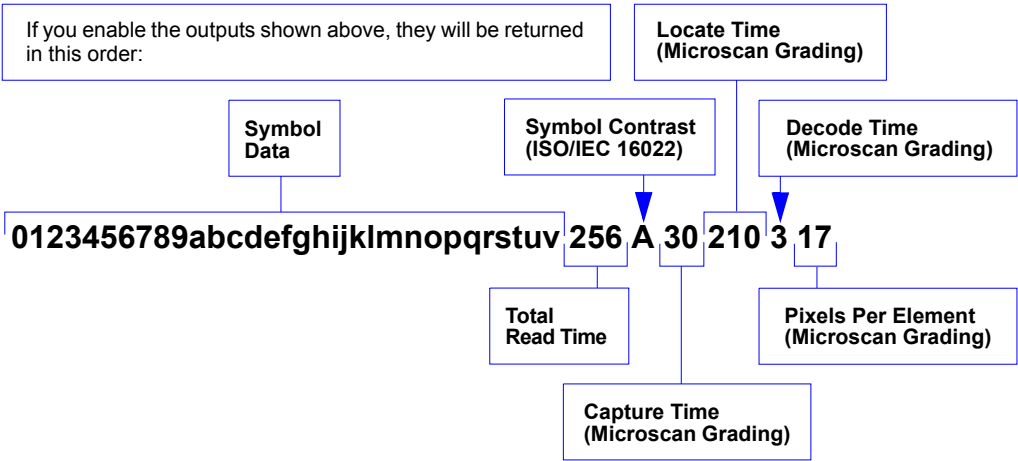
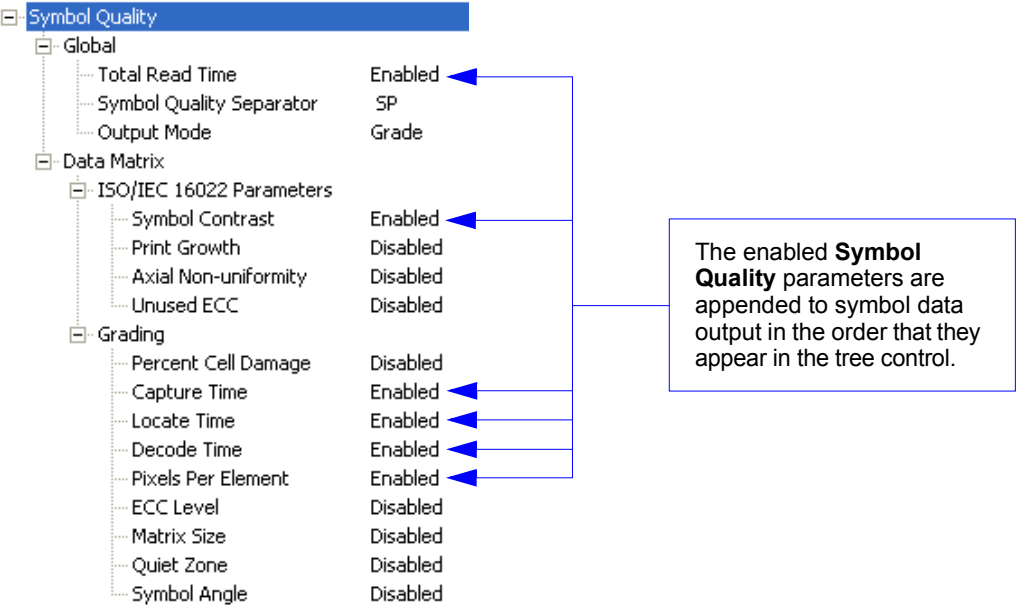
Symbol Quality Separator / Output Mode	< <b>K708</b> ,symbol quality separator,Data Matrix output mode>
ISO/IEC 16022 Symbol Quality Output	< <b>K709</b> ,symbol contrast,print growth,axial non-uniformity,unused ECC>
Microscan Symbol Quality Output	< <b>K710</b> ,percent cell damage,total read time,capture time,locate time,decode time,pixels per element,ECC level,matrix size,quiet zone,symbol angle>

# Overview of Symbol Quality

Information about symbol quality and timing can be appended to symbol data by enabling specific evaluation parameters. The order in which these values are appended corresponds directly to the order in which they appear in **ESP's Symbol Quality** tree control.

Symbol Quality parameters are separated into ISO/IEC 16022 Symbol Quality Output and Microscan Symbol Quality Output.

**Total Read Time**, **Symbol Quality Separator**, and **Output Mode** are common to both ISO/IEC 16022 and Microscan Grading Parameters.



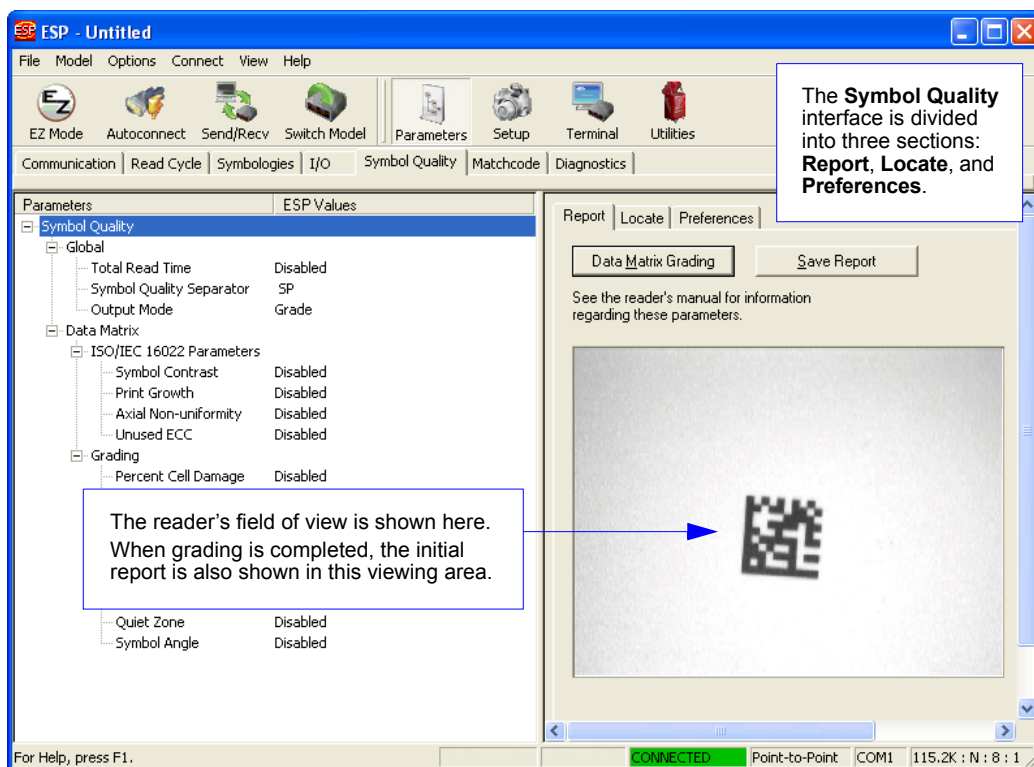


## Report Tab

The **Report** tab allows the user to initiate **Data Matrix Grading** and to generate and save grading reports.

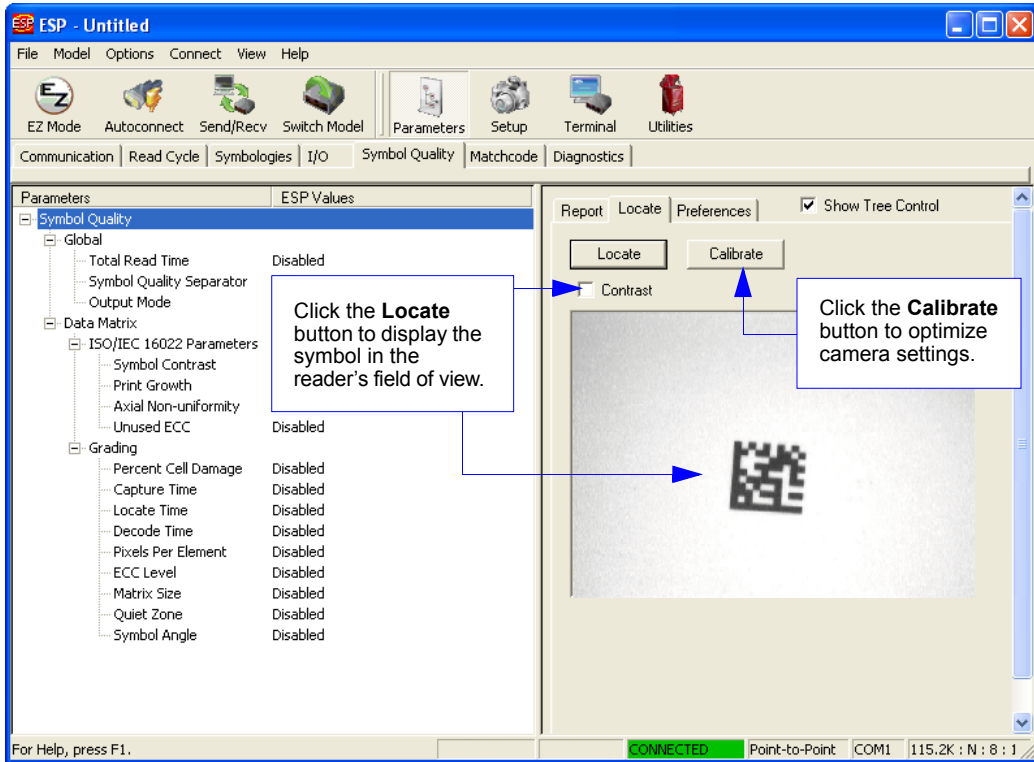
To save the report as a PDF file, an HTML file, a CSV file, or an RTF file, click the **Save Report** button.

Reports appear in the viewing area underneath the **Data Matrix Grading** and **Save Report** buttons.



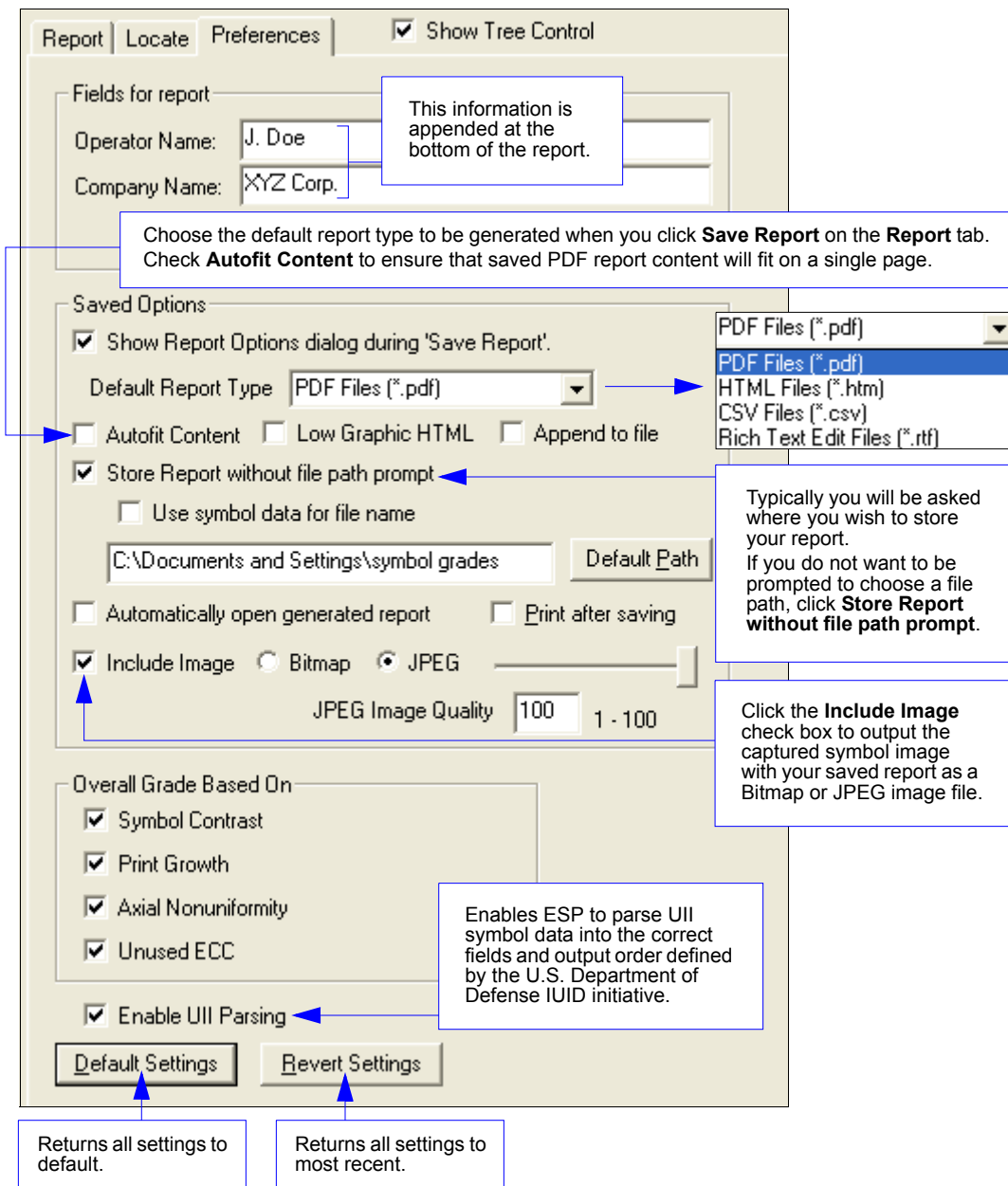
## Locate Tab

The **Locate** tab allows the user to determine the position of the symbol within the reader's field of view, and to optimize the reader's camera settings for symbol quality evaluation.



## Preferences Tab

The **Preferences** tab allows the user to set defaults that are automatically activated when grading is completed. For this reason, the user is advised to set preferences before performing grading.



# Symbol Quality Separator/Data Matrix Output Mode

**Symbol Quality Separator** applies to both Symbol Quality groups: ISO/IEC 16022 Symbol Quality Output and Microscan Symbol Quality Output.

## Symbol Quality Separator

- Definition:** Inserts a separator between each enabled field of the symbol quality output.
- Serial Cmd:** <K708,symbol quality separator,output mode>
- Default:** <SP> (space character)
- Options:** Any ASCII character.

**Important:** The ASCII characters <, >, and , can only be entered as hex pairs:

- For <: <K708h,3C>
- For >: <K708h,3E>
- For , : <K708h,2C>

## Symbol Quality Separator by ESP

ParametersESP Values

[-] Symbol Quality

[-] Global

Total Read TimeDisabled

Symbol Quality SeparatorSP

Output ModeGrade

When you double-click on the default separator SP, the **Symbol Quality Separator Calculator** appears. Use the calculator keypad for one-click separator configuration.

[-] Symbol Quality

[-] Global

Total Read TimeDisabled

Symbol Quality Separator

Output Mode

[-] Data Matrix

[-] ISO/IEC 16022 Parameters

Symbol Contrast

Print Growth

Axial Non-uniformity

Unused ECC

[-] Grading

Percent Cell Damage

Capture Time

SP

NUL	SOH	STX	ETX	EOT	ENQ
ACK	BEL	BS	HT	LF	VT
FF	CR	SO	SI	DLE	DC1
DC2	DC3	DC4	NAK	SYN	ETB
CAN	EM	SUB	ESC	FS	GS
RS	US	SP			

Click 'Delete' to remove characters.

## **Data Matrix Output Mode**

**Note: Output Mode** applies to ISO/IEC 16022 symbol quality parameters.

**Definition:** Output Mode specifies how the four output parameters, if enabled, are formatted.

**Serial Cmd:** <K708,symbol quality separator,**output mode**>

**Default:** **Grade**

**Options:** **0 = Grade**                      1 = Value

### **Grade**

If in **Grade Mode**, a grade (A,B,C,D) is appended to the symbol data.

### **Value**

If in **Value Mode**, the calculated value for the given parameter is appended to the symbol data.

# ISO/IEC 16022 Symbol Quality Output

Symbol Quality Parameters for Data Matrix symbols specified by ISO/IEC 16022.

## Symbol Contrast

- Usage:** Lets the user know if contrast settings are less than acceptable.
- Definition:** All the pixels that fall within the area of the test symbol, including its required zone, will be sorted by their reflectance values to select the darkest 10% and the lightest 10% of the pixels. The arithmetic mean of the darkest and the lightest pixels is calculated and the difference of the two means is the Symbol Contrast.
- (ANSI) Symbol Contrast grading is defined this way:
- A (4.0) if SC > 70%
  - B (3.0) if SC > 55%
  - C (2.0) if SC > 40%
  - D (1.0) if SC > 20%
  - F (0.0) if SC < 20%
- If enabled, the symbol contrast is appended to the symbol data according to the **ISO/IEC 16022 Symbol Quality Output Mode** setting.
- Serial Cmd:** <**K709**,symbol contrast,print growth,axial non-uniformity,UEC>
- Default:** Disabled
- Options:** 0 = Disabled 1 = Enabled

## Print Growth

- Definition:** The extent to which dark or light markings appropriately fill or exceed their module boundaries. These values are determined by counting pixels in the clock pattern of the binary digitized image, then comparing it to a nominal value and minimum and maximum values. The print growth grade is defined in this way:
- A (4.0) if  $-.050 < PG < 0.50$
  - B (3.0) if  $-.070 < PG < 0.70$
  - C (2.0) if  $-.085 < PG < 0.85$
  - D (1.0) if  $-.10 < PG < 1.00$
  - F (0.0) if  $PG < -1.00$  or  $PG > 1.00$
- If enabled, the print growth is appended to the symbol data according to the **ISO/IEC 16022 Symbol Quality Output Mode** setting.
- Serial Cmd:** <**K709**,symbol contrast,print growth,axial non-uniformity,UEC>
- Default:** Disabled
- Options:** 0 = Disabled 1 = Enabled

## Axial Non-Uniformity (Data Matrix Only)

**Definition:** Axial non-uniformity is a measure of how much the sampling point spacing differs from one axis to another, namely  $AN = \text{abs}(XAVG - YAVG) / ((XAVG + YAVG)/2)$  where  $\text{abs}()$  yields the absolute value. If a symbology has more than two major axes, then AN is computed for those two average spacings which differ the most. (ANSI) axial non-uniformity grading is defined this way:

A (4.0) if  $AN < .06$

B (2.0) if  $AN < .08$

C (2.0) if  $AN < .10$

D (1.0) if  $AN < .12$

F (0.0) if  $AN > .12$

If enabled, the axial non-uniformity is appended to the symbol data according the **ISO/IEC 16022 Symbol Quality Output Mode** setting.

**Serial Cmd:** **<K709,symbol contrast,print growth,axial non-uniformity,UEC>**

**Default:** **Disabled**

**Options:** **0 = Disabled** 1 = Enabled

## Unused Error Correction (Data Matrix Only)

**Definition:** The correction capacity of Reed-Solomon decoding is expressed in the equation:  $e + 2d < d - p$ , where **e** is the number of erasures, **d** is the number of error correction code words, and **p** is the number of code words reserved for error detection.

A (4.0) if  $UEC > .62$

B (3.0) if  $UEC > .50$

C (2.0) if  $UEC > .37$

D (1.0) if  $UEC > .25$

F (0.0) if  $UEC < .25$

If enabled, the UEC is appended to the symbol data according to the **ISO/IEC 16022 Symbol Quality Output Mode** setting.

**Serial Cmd:** **<K709,symbol contrast,print growth,axial non-uniformity,UEC>**

**Default:** **Disabled**

**Options:** **0 = Disabled** 1 = Enabled

# ISO/IEC 16022 Symbol Quality Output by ESP

ESP's **Symbol Quality** interface allows you to evaluate Data Matrix symbols for compliance with ISO/IEC 16022 requirements.

First, determine which ISO/IEC 16022 Parameters you need to evaluate, using the **Symbol Quality** tree control.

Data Matrix

ISO/IEC 16022 Parameters

Symbol Contrast

Print Growth

Axial Non-uniformity

Unused ECC

Enabled

Enabled

Enabled

Enabled

Once you have set these Data Matrix evaluation parameters, go to the **Preferences Tab** and set your Data Matrix Grading Report output preferences.

After your output preferences are set, click the **Data Matrix Grading** button.

There will be a wait of a few seconds, and then the evaluation results will appear in the **Symbol Quality** view, in a format similar to the one shown below.

Data Matrix Grading Report										GRADE
ISO/IEC 16022:	Symbol Contrast	=	80 %	Dx	-	54 %	Dnom	-	50 %	A
	Print Growth	=	0.26	Dy	-	54 %	Dmax	-	65 %	A
	Axial Nonuniformity	=	0.00	Xavg	-	13.2	Dmin	-	35 %	A
	Unused ECC	=	100 %	Yavg	-	13.1	Eact	-	0	A
GRADING:	% Cell Damage	=	0 %	Emax	-	10				
	Total Read Time	=	99 ms							
	Capture Time	=	65 ms							
	Locate Time	=	27 ms							
	Decode Time	=	7 ms							
	Pixels/Element	=	13.1							
	ECC Level	=	200							
	Matrix Size	=	14x14							
	Quiet Zone	=	PASS							

If you want to view the results in a report format, click the **Save Report** button. The report will be output in the format you chose in the **Preferences** dialog.

8-12

MicroHAWK ID-20, ID-30, ID-40 User Manual



# Microscan Symbol Quality Output

## Percent Cell Damage (Data Matrix Only)

**Definition:** When this feature is enabled, the cell damage percentage is appended to data output.

**Serial Cmd:** <K710,percent cell damage,total read time,capture time,locate time,decode time,pixels per element,ECC level,matrix size,quiet zone,symbol angle>

**Default:** Disabled

**Options:** 0 = Disabled 1 = Enabled

## Total Read Time

**Definition:** The time that transpires between the image capture and the output of the decoded data, including locate time.

When enabled, the total read time is appended to the symbol data.

**Serial Cmd:** <K710,percent cell damage,**total read time**<sup>a</sup>,capture time,locate time,decode time,pixels per element,ECC level,matrix size,quiet zone,symbol angle>

**Default:** Disabled

**Options:** 0 = Disabled 1 = Enabled

a. **Total Read Time** applies globally to all three Symbol Quality standards.

## Capture Time

**Definition:** Capture time (in milliseconds) is a fixed “overhead” that includes the time of capture and transfer of the image.

When enabled, the capture time is appended to the symbol data.

**Serial Cmd:** <K710,percent cell damage,total read time,**capture time**,locate time,decode time,pixels per element,ECC level,matrix size,quiet zone,symbol angle>

**Default:** Disabled

**Options:** 0 = Disabled 1 = Enabled

## Locate Time

**Definition:** The time in milliseconds from the start of image processing until the symbol has been located and is ready to be decoded.

When enabled, the locate time is appended to the symbol data.

**Serial Cmd:** <K710,percent cell damage,total read time,capture time,**locate time**,decode time,pixels per element,ECC level,matrix size,quiet zone,symbol angle>

**Default:** Disabled

**Options:** 0 = Disabled 1 = Enabled

## Decode Time

**Definition:** The time in milliseconds required to decode a symbol.  
When enabled, the decode time is appended to the symbol data.

**Serial Cmd:** <K710,percent cell damage,total read time,capture time,locate time,**decode time**,pixels per element,ECC level,matrix size,quiet zone,symbol angle>

**Default:** Disabled

**Options:** 0 = Disabled 1 = Enabled

## Pixels Per Element (Data Matrix Only)

**Definition:** The number of pixels for each element, either dark or light for both **x** and **y** directions.  
When enabled, the pixels per element value is appended to the symbol data.

**Serial Cmd:** <K710,percent cell damage,total read time,capture time,locate time,decode time,**pixels per element**,ECC level,matrix size,quiet zone,symbol angle>

**Default:** Disabled

**Options:** 0 = Disabled 1 = Enabled

## Error Correction Level (Data Matrix Only)

**Definition:** Outputs the Data Matrix ECC level.  
When enabled, the ECC level is appended to the symbol data.

**Serial Cmd:** <K710,percent cell damage,total read time,capture time,locate time,decode time,pixels per element,**ECC level**,matrix size,quiet zone,symbol angle>

**Default:** Disabled

**Options:** 0 = Disabled 1 = Enabled

## Matrix Size (Data Matrix Only)

**Definition:** Defines the symbol matrix size, in number of pixels in both the **x** and **y** axis.  
When enabled, the matrix size value is appended to the symbol data.

**Serial Cmd:** <K710,percent cell damage,total read time,capture time,locate time,decode time,pixels per element,ECC level,**matrix size**,quiet zone,symbol angle>

**Default:** Disabled

**Options:** 0 = Disabled 1 = Enabled

## Quiet Zone (Data Matrix Only)

**Definition:** When this feature is enabled, the size of the quiet zone is evaluated and a PASS or FAIL message is appended to the symbol data.

**Serial Cmd:** <**K710**,percent cell damage,total read time,capture time,locate time,decode time,pixels per element,ECC level,matrix size,**quiet zone**,symbol angle>

**Default:** Disabled

**Options:** 0 = Disabled 1 = Enabled

## Symbol Angle (Data Matrix Only)

**Definition:** When this feature is enabled, the symbol orientation is appended to data output as a degree value representing the angle of the Data Matrix symbol's L-shaped finder pattern relative to the reader.

**Serial Cmd:** <**K710**,percent cell damage,total read time,capture time,locate time,decode time,pixels per element,ECC level,matrix size,**quiet zone**,**symbol angle**>

**Default:** Disabled

**Options:** 0 = Disabled 1 = Enabled

# Microscan Symbol Quality Output by ESP

ESP's **Symbol Quality** interface allows you to evaluate Data Matrix symbols for compliance with a rigorous set of standards, such as Locate Time, Capture Time, and Decode Time.

First, determine which parameters you want to evaluate, using the **Symbol Quality** tree control.

☐ Grading

Percent Cell Damage

Disabled

Capture Time

Disabled

Locate Time

Disabled

Decode Time

Disabled

Pixels Per Element

Disabled

ECC Level

Disabled

Matrix Size

Disabled

Quiet Zone

Disabled

Symbol Angle

Disabled

Once you have set these Data Matrix evaluation parameters, go to the **Preferences Tab** and set your Data Matrix Grading Report output preferences.

Report | Locate | Preferences | **Save Test Control**

Fields to report

Operator Name:

Company Name:

Saved Options

☒ Show Report Contents during Testing Time Report

Default Report Type:

☐ Auto Cancel ☐ Low-Speed HTML ☐ Append to file

☒ Show Report when the test is over

☐ Use saved data for the next

☐ Annotations and settings window open

☐ Annotations do open (printed report)

☒ Include Images ☐ Storage ☐ JPEG

☐ Print After Saving

Overall Grade Based On

☒ Symbol Contrast

☐ Print Growth

☐ Axial Nonuniformity

☐ Unused ECC

☐ Grade All Passing

After your output preferences are set, click the **Data Matrix Grading** button.

Data Matrix Grading

There will be a wait of a few seconds, and then the evaluation results will appear in the **Symbol Quality** view, in a format similar to the one shown below.

Data Matrix Grading Report										GRADE
ISO/IEC 16022:	Symbol Contrast	=	80 %							A
	Print Growth	=	0.26	Dx	- 51 %	Dnom	- 50 %			A
				Dy	- 54 %	Dmax	- 65 %			
						Dmin	- 35 %			
	Axial Nonuniformity	=	0.00	Xavg	- 13.2					A
	Unused ECC	=	100 %	Yavg	- 13.1					A
				Eact	- 0					
				Emax	- 10					
GRADING:	% Cell Damage	=	0 %							
	Total Read Time	=	99 ms							
	Capture Time	=	65 ms							
	Locate Time	=	27 ms							
	Decode Time	=	7 ms							
	Pixels/Element	=	13.1							
	ECC Level	=	200							
	Matrix Size	=	14x14							
	Quiet Zone	=	PASS							

If you want to view the results in a report format, click the **Save Report** button. The report will be output in the format you chose in the **Preferences** dialog.

Save Report...



# 9 Matchcode

**Contents**

Matchcode by ESP ..... 9-2

Matchcode Serial Commands ..... 9-3

Overview of Matchcode ..... 9-4

Matchcode Type ..... 9-5

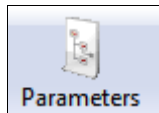
Match Replace..... 9-10

Mismatch Replace..... 9-11

New Master Pin ..... 9-12

This section explains the MicroHAWK ID’s matchcode output functions and the master symbol database setup.

# Matchcode by ESP



Click the **Parameters** button and then the **Matchcode** tab.

Parameters	ESP Values
<div><div>+</div>Matchcode</div>	
Matchcode Type	Disabled
Sequential Matching	Increment
Match Start Position	0
Match Length	1
Wild Card	*
Sequence on No Read	Enabled
Sequence on Mismatch	Disabled
New Master Pin	Disabled*
<div><div>+</div>Match Replace</div>	Disabled*
Replacement String	Enabled
<div><div>+</div>Mismatch Replace</div>	Disabled
Replacement String	MISMATCH

To open nested options, **single-click** the +.

To change a setting, **double-click** the setting and use your cursor to scroll through the options.

## Matchcode Serial Commands

Matchcode	<i>&lt;K223,matchcode type,sequential matching,match start position,match length,wildcard character, sequence on No Read,sequence on mismatch&gt;</i>
Number of Master Symbols	<i>&lt;K224,number of master symbols&gt;</i>
New Master Pin	<i>&lt;K225,status&gt;</i>
Sequence Step	<i>&lt;K228,sequence step value&gt;</i>
Master Symbol Data	<i>&lt;K231,index,master symbol data&gt;</i>
Match Replace	<i>&lt;K735,status,match replacement string&gt;</i>
Mismatch Replace	<i>&lt;K736,status,mismatch replacement string&gt;</i>

## Overview of Matchcode

**Definition:** **Matchcode** allows the user to store master symbol data in the reader's memory, compare that data against other symbol data, and define how symbol data and/or discrete signal output will be directed.

A master symbol database can be set up for up to 10 master symbols.

**Note:** **Matchcode** will function with multiple symbols; however, if **Matchcode Type** is set to **Sequential** or if **Triggering Mode** is set to **Continuous Read 1 Output**, the reader will behave as if **Number of Symbols** were set to **1**, regardless of the user-defined configuration.

**Usage:** Matchcode is used in applications to sort, route, or verify data based on matching the specific symbol in a variety of ways as defined in this section. For example, a manufacturer might sort a product based on dates that are embedded in the symbol.

### Steps for Entering and Using Master Symbols

1. Set **Triggering Mode** to **External** or **Serial**.
2. Choose the method of symbol comparison that fits your application.
3. Define the output you want to achieve with your matchcode setup:
  - a. Symbol data output
  - b. Discrete output
4. Select the number of master symbols you want to create.
5. Decide which of 4 ways you want to enter your master symbol(s):
  - a. Use **ESP** to type master symbol data directly.
  - b. Send a serial command with symbol data in the form of **<M231, master symbol#, data>**.
  - c. Send a **<G> (Read Next Symbol as Master Symbol)** command.
  - d. Enable the **New Master Pin** command and activate the discrete input to store the next symbol read as the master symbol.



# Matchcode Type

**Definition:** Allows the user to choose the way that master symbols will be compared with subsequently read symbols.

**Note:** First set **Triggering Mode** to **External** or **Serial**.

**ESP:**

Parameters	
[-] Matchcode	
Matchcode Type	Disabled*
Sequential Matching	Disabled*
Match Start Position	Enabled
Match Length	Wild Card
Wild Card	Sequential
Sequence on Noread	Enabled
Sequence on Mismatch	Disabled
New Master Pin	Disabled
[-] Match Replace	Disabled
Replacement String	MATCH
[-] Mismatch Replace	Disabled
Replacement String	MISMATCH

**Serial Cmd:** <K223,**matchcode type**,sequential matching,match start position,match length,wild card character,sequence on No Read,sequence on mismatch>

**Default:** **Disabled**

**Options:** **0 = Disabled**  
 1 = Enabled  
 2 = Wild Card  
 3 = Sequential

**Disabled:** Has no effect on operations.

**Enabled:** Instructs the reader to compare symbols or portions of symbols with the master symbol.

**Wild Card:** Allows the user to enter user-defined wild card characters in the master symbol.

**Sequential:** Instructs the reader to sequence after each match (numeric only) and compare symbols or portions of symbols for sequential numbers.  
**Note:** If **Matchcode Type** is set to **Sequential**, the reader will behave as if **Number of Symbols** were set to **1**, regardless of the user-defined configuration.

## Sequential Matching

- Usage:** Useful in tracking product serial numbers that increment or decrement sequentially.
- Definition:** With Sequential enabled, Sequential Matching determines if a count is in ascending (incremental) or descending (decremental) order.
- Serial Cmd:** *<K223,matchcode type,sequential matching,match start position,match length,wild card character,sequence on No Read,sequence on mismatch>*
- Default:** **Increment**
- Options:** **0 = Increment**                      1 = Decrement

## Match Start Position

- Usage:** **Match Start Position** is useful in defining specific portions of a symbol for comparison. For example, if a symbol contains a part number, manufacturing date, and lot code info, but you are only interested in the part number information, you can set the reader to sort only the part number and ignore the other characters.
- Definition:** **Match Start Position** determines the portions of symbols that will be matched by defining the first character in the symbol (from left to right) that will be compared with those of the master symbol, when **Matchcode Type** is set to **Enabled** or **Sequential**.
- Function:** For example, if **Match Start Position** is set to **3**, the first 2 characters read in the symbol will be ignored and only the 3rd and subsequent characters to the right will be compared, up to the number of characters specified by **Match Length**.
- Serial Cmd:** *<K223,matchcode type,sequential matching,match start position,match length,wild card character,sequence on No Read,sequence on mismatch>*
- Default:** **0**
- Options:** 0 to 3000

**Note:** **Match Start Position** must be set to **1** or greater to enable this feature. A **0** setting will disable this feature.

## Match Length

- Usage:** *Example:* If **Match Length** is set to **6** in a 10-character symbol, and **Match Start Position** is set to **2**, only the 2nd through 7th characters (from left to right) will be compared.
- Definition:** Defines the length of the character string that will be compared with that of the master symbol when **Match Start Position** is set to **1** or greater. When **Match Start Position** is set to **0**, no comparison will occur.
- Serial Cmd:** *<K223,matchcode type,sequential matching,match start position,match length,wild card character,sequence on No Read,sequence on mismatch>*
- Default:** **1**
- Options:** 1 to 3000

## Wild Card Character

- Usage:** *Example:* With **Wild Card Character** defined as the default asterisk, defining **CR\*34** as the master symbol will result in matches for CR134 and CR234, but not CR2345. Entering URGENT\*\* as your master symbol will result in matches for URGENT, URGENT1, and URGENT12 but not for URGENT123. This means any wild cards appended to the master symbol data will result in matches of symbols in variable lengths up to the master symbol lengths but not over. However, wild cards in the beginning or center of a symbol (e.g., UR\*\*NT) do not allow for variable symbol lengths.
- Definition:** Wild Card Character allows a user to define a wild card character as part of the master symbol.
- Serial Cmd:** *<K223,matchcode type,sequential matching,match start position,match length,wild card character,sequence on No Read,sequence on mismatch>*
- Default:** \* (asterisk)
- Options:** Any ASCII character.

**Important:** The ASCII characters <, >, and , can only be entered as hex pairs:

For <: <K223h,,,,,3C>

For >: <K223h,,,,,3E>

For , : <K223h,,,,,2C>

### Sequence on No Read

**Usage:** **Sequence on No Read** is useful when the reader needs to stay in sequence even if no decode occurs.

**Definition:** When **Sequence on No Read** is **Enabled** and **Matchcode** is set to **Sequential**, the reader sequences the master symbol on every match or No Read. When disabled, it does not sequence on a No Read.

**Serial Cmd:** **<K223,matchcode type,sequential matching,match start position,match length,wild card character,sequence on No Read,sequence on mismatch>**

**Default:** Enabled

*Options:*            0 = Disabled

**1 = Enabled**

As an example of **Sequence on No Read Enabled**, consider the following decodes:

<i><b>Master Symbol</b></i>	<i><b>Decoded Symbol</b></i>	<i><b>Master Symbol after Decode</b></i>
001	001	002
002	002	003
003	No Read	004 (sequenced on No Read)
004	004	005
005	No Read	006 (sequenced on No Read)
006	No Read	007 (sequenced on No Read)
007	007	008

As an example of **Sequence on No Read Disabled**, consider the following series of decodes:

<i>Master Symbol</i>	<i>Decoded Symbol</i>	<i>Master Symbol after Decode</i>
001	001	002
002	002	003
003	No Read	003 (not sequenced)
003	003	004
004	No Read	004 (not sequenced)
004	No Read	004 (not sequenced)
004	004	005

## Sequence on Mismatch

**Note:** Matchcode must be set to **Sequential** for this command to function.

**Usage:** Enable this parameter if every trigger event should have a decode *and* more than one consecutive mismatch may occur.

Disable this parameter if every trigger event should have a decode but no more than one consecutive mismatch may occur.

**Definition:** When set to Enabled, the master symbol sequences on every decode, match, or mismatch.

When set to Disabled, the master symbol will not sequence whenever consecutive mismatches occur.

**Serial Cmd:** <K223,matchcode type,sequential matching,match start position,match length,wild card character,sequence on No Read,**sequence on mismatch**>

**Default:** Disabled

**Options:** 0 = Disabled 1 = Enabled

The reader will sequence the master to one more or one less than the decoded symbol.

As an example of **Sequence on Mismatch Enabled**, consider the following decodes:

<b>Master Symbol</b>	<b>Decoded Symbol</b>	<b>Master Symbol after Decode</b>
001	001	002
002	002	003
003	abc	004 (sequenced on mismatch)
004	004	005
005	def	006 (sequenced on mismatch)
006	ghi	007 (sequenced on mismatch)
007	007	008

As an example of **Sequence on Mismatch Disabled**, consider the following decodes:

<b>Master Symbol</b>	<b>Decoded Symbol</b>	<b>Master Symbol after Decode</b>
001	001	002
002	002	003
003	abc	004 (sequenced because of previous match)
004	004	005
005	def	006 (sequenced because of previous match)
006	ghi	006 (not sequenced)
006	006	007

Options:      0 = Disabled                      1 = Enabled

For , : **<K735h,,2C>**

**Usage:** Provides a convenient shortcut for applications that need to output a predefined text string whenever a symbol does not match a master symbol.

**Definition:** Outputs a user-defined data string whenever a mismatch occurs and **Matchcode** is enabled.

**Serial Cmd:** **<K736,status,replacement string>**

**Default:** Disabled

Options:      **0 = Disabled**                      **1 = Enabled**

## Replacement String

**Definition:** User-defined data string that, when enabled, replaces symbol data whenever a mismatch occurs.

**Serial Cmd:** **<K736,status,replacement string>**

*Default:* **MISMATCH**

**Options:** An ASCII string up to 64 characters.

**Important:** The ASCII characters `<`, `>`, and `,` can only be entered as hex pairs:

For <: <K736h,,3C>

For >: <K736h,,3E>

For , : <K736h,,2C>

## New Master Pin

**Definition:** If **Matchcode** and **New Master Pin** are enabled and the new master pin is momentarily connected to ground (must be held low for a minimum of 10 ms) master symbol information will be loaded into the database based on the next read cycle that achieves a Good Read, starting with **Index 1**. The **Master Symbol Database** will be loaded with all symbols decoded in the read cycle as long as it does not exceed the **Number of Master Symbols** parameter.

**Note:** For the MicroHAWK ID-30, Input 1 (pin 12) is Default and Input 2 (pin 8) is New Master. For The MicroHAWK ID-40, Input 1 (pin 3) is Default and Input 2 (pin 4) is New Master.

**Serial Cmd:** **<K225,status>**

**Default:** **Disabled**

**Options:** **0 = Disabled**                      1 = Enabled

## New Master Load Status

**Definition:** The new master status responds with the number of the next master position to be loaded, where 0 represents “idle” or “no master to be loaded.”

**Example:** If the user has the **Number of Master Symbols** set to 1, and then either sends a **<G>** or toggles an active **New Master Pin**, the state will be 1, and prior to reading and effectively loading position 1, the response to **<NEWM>** would be **<NEWM/1>**. Once a symbol has been read and loaded, the status will be cleared: **<NEWM/0>**.

**Serial Cmd:** **<NEWM>**



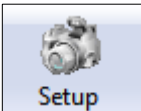
# ■ **10 Camera and IP Setup**

## **Contents**

Camera and IP Setup by ESP .....	10-2
Camera and IP Setup Serial Commands.....	10-3
Video.....	10-4
Evaluation .....	10-5
Calibration.....	10-8
Window of Interest.....	10-9
Configuration Database.....	10-13
Dynamic Setup .....	10-14
Pixel Binning .....	10-15
Camera Settings .....	10-16
Color Filter .....	10-17
Illumination Brightness.....	10-18
Morphological Pre-Processing.....	10-19
Morphological Operation and Operator Size .....	10-20

This section provides adjustment parameters for the physical controls of the camera, image acquisition, database settings, and image diagnostics.

# Camera and IP Setup by ESP



Click the **Setup** button to bring up the **Camera Setup** menu.

Parameters	ESP Values
Camera Setup	
Camera Settings	
Gain	25
Exposure	10000
Pixel Binning	Disabled
Illumination Brightness	High
Window of Interest	
Top	0
Left	0
Height	1944
Width	2592
Image Processing Settings	
Morphological Preprocessing	Disabled
Operation	Erode*
Operator Size	Erode*
Focus Distance	Dilate
Distance Units	Open
Lens Model	Close
White Balance	45 degree
Red Gain	25
Green Gain	0
Blue Gain	1

To change a setting, **double-click** the setting and use your cursor to scroll through the options.

To open nested options, **single-click** the + and it will become a – as shown above.

## Camera and IP Setup Serial Commands

Window of Interest	<K516,row pointer,column pointer,row depth,column width>
Illumination Brightness	<K536,brightness,light source>
Camera Settings	<K541,exposure,gain>
Pixel Binning	<K542,status>
Color Filter	<K543,color filter>
Morphological Pre-Processing	<K550,status>
Morphological Operation	<K551,morphological operation,operator size>

# Video

The **Video** view is similar to the **EZ Mode** interface, in that the user has the ability to perform the same **Locate**, **Calibrate**, and **Test** routines. There is also a focal distance adjustment tool to the right of the video view.

**Video** also features **Capture and Decode** functionality, which is similar to the **Configuration** interfaces (**Communication**, **Read Cycle**, **Symbologies**, **I/O**, **Matchcode**, and **Diagnostics**).

The screenshot shows the Video interface with the following components and callouts:

- Locate** button: Activates the reader's target pattern and initiates live video display of the symbol in the field of view.
- Calibrate** button: Initiates Calibration routine.
- Capture and Decode** button: Allows the user to take an image capture of the symbol in the field of view at any time, and to decode the symbol data. Similar to the **Capture and Decode** function in the **Configuration** views.
- Test** button: Initiates the **Test** routine. Allows the user to test the reader's read rate and decodes per second, and also displays decoded symbol data in the field at the bottom of the screen. Click **Stop** to end the **Test** routine.
- Stop** button: Click **Stop** to end the **Test** routine.
- Decodes per Second** checkbox: Checked.
- Contrast** checkbox: Unchecked.
- Focal Distance** slider: Clicking this icon enables the focal distance slide control to the right of the video view. The focal distance value is displayed just below the icon. The current value is 4.00 inches.
- Symbol Data Table**:

Attribute	Value
Number of Symbols	1
Symbol #1	
Symbol Data	Microscan

Symbol data and **Test** results are shown in this field.

When enabled, shows the symbol in higher contrast than the default contrast value.

## Evaluation

In **Evaluation**, you can view images currently in the reader, capture and decode a symbol, save it as a digital file, and perform histogram and line scan evaluations.

When you click on the **Evaluation** tab, you will see the images that are currently stored in the reader.

Click **Receive** to refresh this view.

Click **Capture/Decode** to display the current stored image. Only one capture and decode event will occur, regardless of read cycle settings.

Click **Read** to trigger a read cycle. If there is enough time in the read cycle, up to 32 good reads (or 6 full-scale images) can be captured and displayed, depending on the size of the images, and depending on the number enabled in **Number of Captures** under **Capture Mode** in the **Camera Setup** menu tree to the left of the tabs.

If you click **Save**, the current image will be saved to the location of your choice.

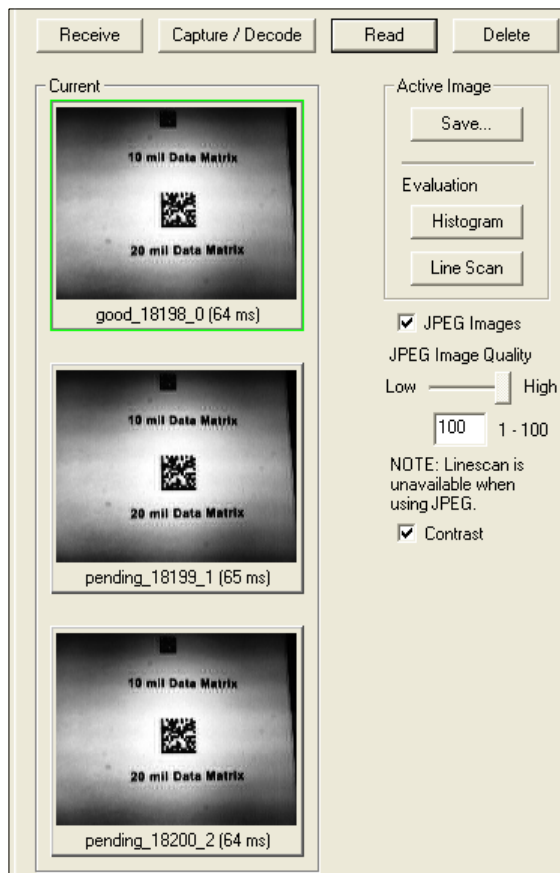
**Note:** An image can only be saved in the format in which it was uploaded to **ESP**. JPEG images will be saved as **.jpg**, and bitmaps will be saved as **.bmp**.

When you click the **JPEG Image** box, notice that the **Line Scan** button is grayed out.

The JPEG option allows faster captures and transfers, but since the JPEG standard compresses image data, it is not suitable for the more rigorous demands of line scan evaluation.

JPEG also allows you to adjust the image quality (resolution) by adjusting the sliding tab between 1 and 100, 1 being the lowest quality and 100 being the highest.

When possible, use the highest quality; when image transfer speed must be increased, use a lower image quality setting. Adjustments for this setting will depend on your specific hardware and software limitations.



## Histogram

**Usage:** Useful in determining quality and contrast of symbols.

**Definition:** A histogram is a graphic representation of the numeric count of the occurrence of each intensity (gray level) in an image. The horizontal axis represents the values of gray levels, and the vertical axis represents the number of pixels for each gray level.

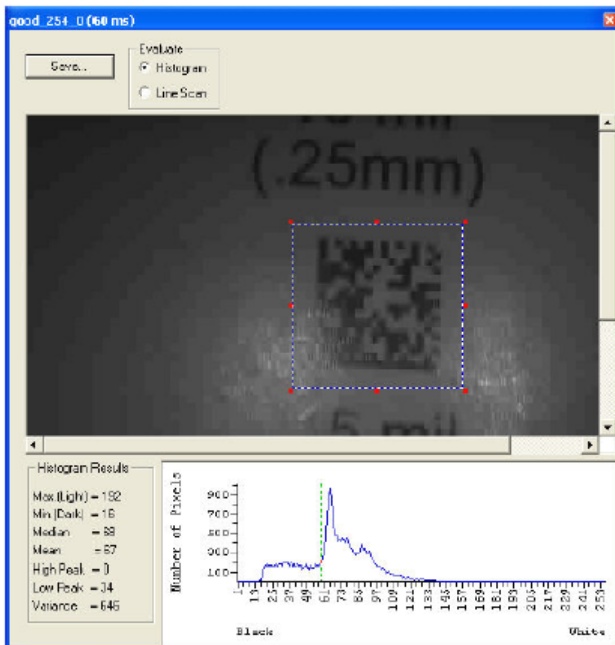
**Note:** Since histograms are performed in the reader, the results will be saved regardless of whether the image was uploaded as a JPEG (.jpg) or a bitmap (.bmp).

1. From the **Evaluation** window, click the **Histogram** button.

The current image is transferred into the histogram operation. This may take a moment, since all the relevant pixels are being evaluated intensively.

2. When the **Histogram** window opens, you may need to expand the window and/or adjust the scroll bars in order to bring the image into view.
3. To generate a histogram, click and drag your cursor diagonally across the symbol or a portion of the symbol.

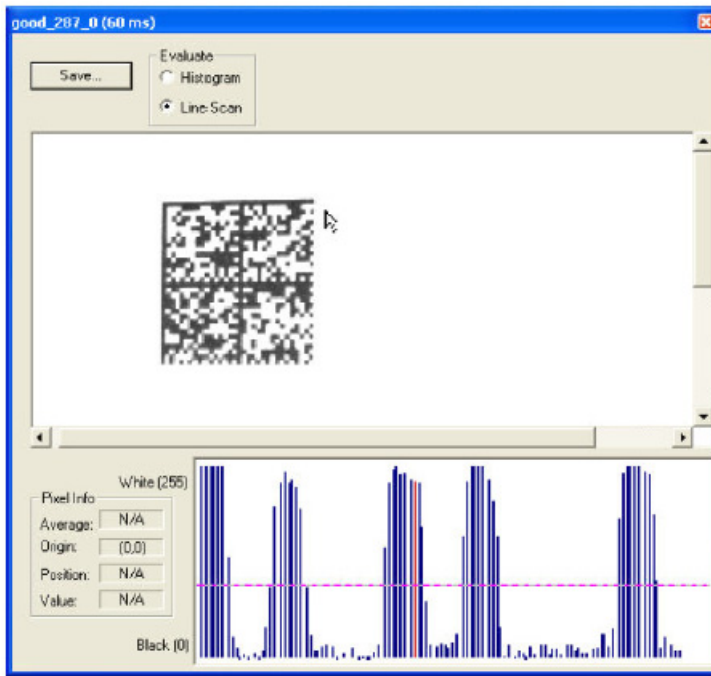
The image will be surrounded by an “area of interest box”, a dashed blue line with red anchor points that can be selected and moved by placing a cursor inside the box, and can be resized by clicking and dragging the anchor points.



## Line Scan

1. From the **Evaluation** tab, click the **Line Scan** button. A window like that shown below will appear.
2. Drag your cursor *horizontally* across the image.

This will create a dashed horizontal line. Pixel information and a visual representation of the dark and light pixels will be displayed below the image.



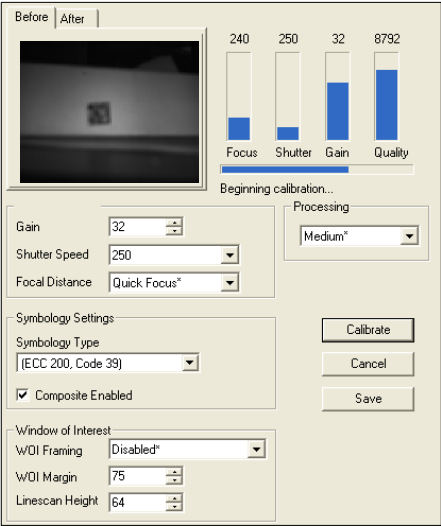
When you center your cursor over the dashed line, it becomes a double arrow cross. This allows you to move the line anywhere in the window. You can also use your keyboard arrows to move the line incrementally in any direction.

As with the histogram, the line scan compares light and dark pixels, but in a spatial distribution. On the **Y** axis of the graph below, **0** is black and **255** is white; the **X** axis represents the horizontal axis of the symbol as described by the line scan.

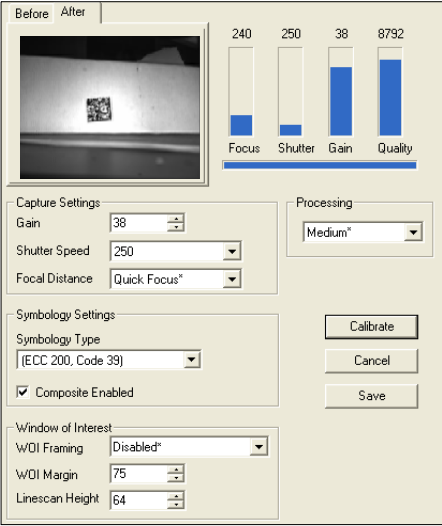
When you click anywhere on the graph, a vertical red line appears at that point and its position and value (in terms of black-to-white) are updated in the **Pixel Info** table to the left of the graph (237 in this example). In addition, a horizontal dashed red line is displayed that indicates the average value.

# Calibration

The Calibration interface in **ESP** is a powerful and intuitive way to optimize the reader's performance. It allows the user to control several complex, simultaneous calibration operations, and to follow the progress of those operations using cues such as progress bars, real-time representations of calibration values, and other dynamic user feedback. See the **Calibration** section for full documentation of this feature.



*ESP's Calibration interface shown during a calibration routine*



*ESP's Calibration interface shown after a calibration routine*



## Window of Interest

The active pixel area of the image sensor is called the **Window of Interest** (WOI). The WOI allows the user to select an area of the field of view in which the desired symbol is located.

The programmable window of interest increases decode speed, improves threshold, and makes it easy to select specific symbols from among several in the field of view. The user provides the upper-left pixel location and the size of the window to define the Window of Interest.

**Important: Window of Interest** will shrink the field of view and therefore could cause symbols to be missed in dynamic applications.

## Window of Interest by ESP

1. From the **Camera** menu, click the **WOI** tab to bring up **Window of Interest**.

If you haven't already captured an image, click the **Capture and Decode** button to decode the present image. If successful, the **Good Read** indicator on the **WOI** tab will be green and the symbol will appear in the pane below.

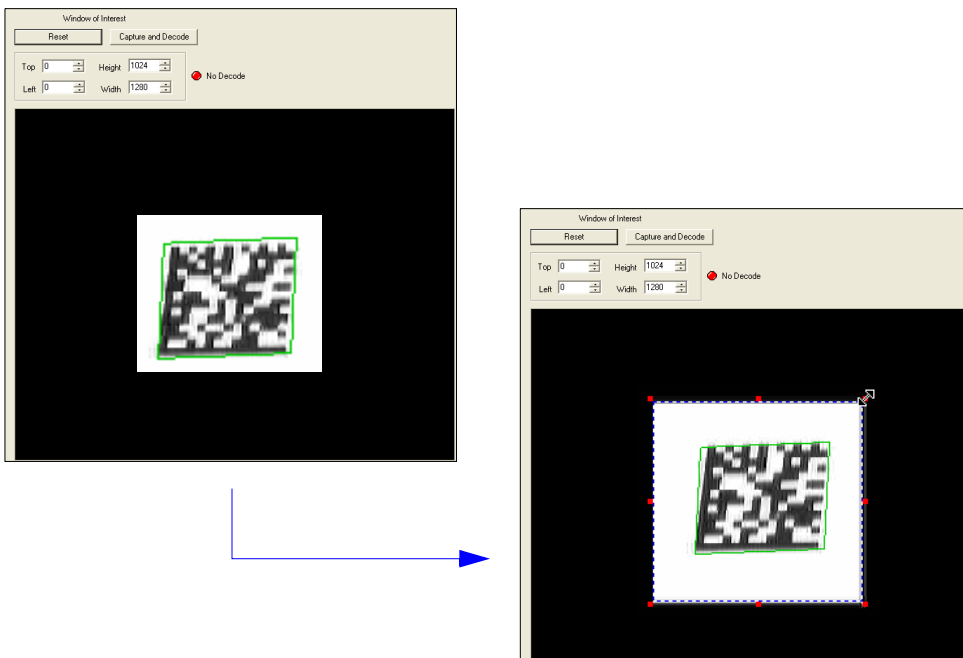
**Note:** You can resize the image by clicking and dragging the **ESP** window from the lower right corner. This is useful where very small symbols are being read.

2. Click and drag your cursor over the symbol that you want to isolate for reading.

Notice that the surrounding area goes black.

You can use the handles on the image area that you have just drawn to resize the region of interest. You can also click on the center of the window of interest and move it.

3. Test the new settings in **Read Rate Mode**.



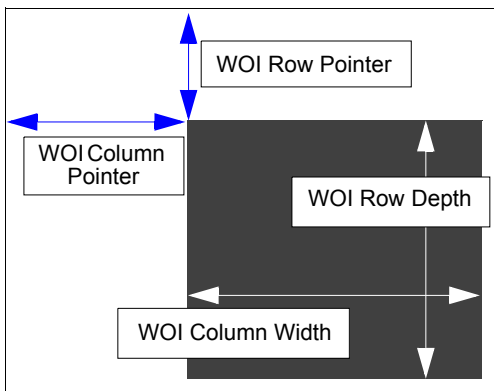
**Note:** To remove the window of interest, click the **Reset** button or click anywhere in the WOI pane.

Note that all pixels not in the WOI are defined as black.

Because the reader has far less processing to do in a smaller window, read rates typically increase dramatically. One possible downside is that the chance of missing a symbol increases with the smaller window. Always verify that your WOI will be large enough to allow for any random movement of symbols in your field of view.

## Window of Interest by Serial Command

The figure shows where to locate the start position of the row and column pointers and how to measure the column depth and row width dimensions.



### Top (Row Pointer)

**Definition:** Defines the row position of the upper-left starting point of the window.

**Serial Cmd:** `<K516,row pointer,column pointer,row depth,column width>`

**Default:** **QSXGA MicroHAWK ID: 0**

**SXGA MicroHAWK ID: 0**

**WVGA MicroHAWK ID: 0**

**Options:** **QSXGA MicroHAWK ID:** 0 to (1944 – row depth)

**SXGA MicroHAWK ID:** 0 to (1024 – row depth)

**WVGA MicroHAWK ID:** 0 to (480 – row depth)

### Left (Column Pointer)

**Definition:** Defines the column position of the upper-left starting point of the window.

**Serial Cmd:** `<K516,row pointer,column pointer,row depth,column width>`

**Default:** **QSXGA MicroHAWK ID: 0**

**SXGA MicroHAWK ID: 0**

**WVGA MicroHAWK ID: 0**

**Options:** **QSXGA MicroHAWK ID:** 0 to (2592 – column width)

**SXGA MicroHAWK ID:** 0 to (1280 – column width)

**WVGA MicroHAWK ID:** 0 to (752 – column width)

## Height (Row Depth)

- Definition:* Defines the size, in rows, of the window. Maximum value is defined as the maximum row size of the image sensor minus the **Top** value.
- Serial Cmd:* <**K516**,row pointer,column pointer,**row depth**,column width>
- Default:* **QSXGA MicroHAWK ID: 1944**  
**SXGA MicroHAWK ID: 1024**  
**WVGA MicroHAWK ID: 480**
- Options:* **QSXGA MicroHAWK ID: 3 to (1944 – row pointer)**  
**SXGA MicroHAWK ID: 3 to (1024 – row pointer)**  
**WVGA MicroHAWK ID: 3 to (480 – row pointer)**

## Width (Column Width)

- Definition:* Defines the column position of the upper-left starting point of the window.
- Serial Cmd:* <**K516**,row pointer,column pointer,row depth,**column width**>
- Default:* **QSXGA MicroHAWK ID: 2592**  
**SXGA MicroHAWK ID: 1280**  
**WVGA MicroHAWK ID: 752**
- Options:* **QSXGA MicroHAWK ID: 8 to (2592 – column pointer)**  
**SXGA MicroHAWK ID: 8 to (1280 – column pointer)**  
**WVGA MicroHAWK ID: 8 to (752 – column pointer)**

# Configuration Database

The Configuration Database allows the user to manage multiple configuration profiles. It is a useful tool in applications that require several different reader configurations to be applied sequentially, and it allows the user to perform far more complex operations than would be possible with only one set of configuration parameters.

See **Configuration Database** for full documentation of this feature.

Active Indexes 5 ☒ Auto Sync Database settings with Reader Advanced Options...

### Capture Settings

<< WOI
ROI >>

Capture Index
Exposure
Gain
Pixel Binning
Top
Left
Height
Width

Current: 10000 25 Disable 0 0 1944 2592 ...

### Processing Settings

Morphology Operation
Morphology Size
Symbol Type

Erode Small ECC 200

	Capture Index	Exposure	Gain	Pixel Binning	Top	Left	Height	Width	Morphology Operation	Morphology Size	Symbol Type	
<input checked="" type="radio"/> 1	10000	20	Disable	0	0	1944	2592	...	Disable	Small	Disabled	Test
<input type="radio"/> 2	10000	20	Disable	0	0	1944	2592	...	Disable	Small	Disabled	Test
<input type="radio"/> 3	10000	20	Disable	0	0	1944	2592	...	Disable	Small	Disabled	Test
<input type="radio"/> 4	10000	20	Disable	0	0	1944	2592	...	Disable	Small	Disabled	Test
<input type="radio"/> 5	10000	20	Disable	0	0	1944	2592	...	Disable	Small	Disabled	Test
<input type="radio"/> 6	10000	20	Disable	0	0	1944	2592	...	Disable	Small	Disabled	Test
<input type="radio"/> 7	10000	20	Disable	0	0	1944	2592	...	Disable	Small	Disabled	Test
<input type="radio"/> 8	10000	20	Disable	0	0	1944	2592	...	Disable	Small	Disabled	Test
<input type="radio"/> 9	10000	20	Disable	0	0	1944	2592	...	Disable	Small	Disabled	Test
<input type="radio"/> 10	10000	20	Disable	0	0	1944	2592	...	Disable	Small	Disabled	Test

☒ Capture for Every Index
 ☐ Show Database Index in Output
 ☐ Sort Index Positions on Good Reads

Calibrate...
Receive Settings
Send Settings
Load Current To Index
Load Index To Current

*Configuration Database view, set for 5 Active Indexes, with Capture Settings highlighted*

## Dynamic Setup

**Dynamic Setup** is used to calculate image capture timing during a read cycle. Without proper timing, the reader will not be able to decode all symbols in a read cycle. An external trigger is used to activate image captures so the user can make timing adjustments as capture events occur.

The screenshot shows the 'Dynamic Setup' window. At the top left is a grayscale image of a barcode. To its right is a vertical 'Read Rate' bar with a blue fill, ranging from 0% at the bottom to 100% at the top. Below the barcode image are three spin boxes labeled 'Capture #', 'Number of Captures', and 'Number of Symbols', each with the value '1'. Below these are 'Delay (Sec)' and 'Delay Adjustment' controls. The 'Delay (Sec)' field shows '0.078432'. The 'Delay Adjustment' section has three radio buttons: 'Coarse', 'Medium' (which is selected), and 'Fine'. At the bottom is a horizontal slider with 'Decrease' and 'Increase' labels and arrow buttons. Callout boxes provide additional context: one points to the spin boxes stating they control 'Capture #', 'Number of Captures', and 'Number of Symbols'; another points to the 'Read Rate' bar stating it is represented graphically in real-time as adjustments are made; and a third points to the 'Delay Adjustment' section stating the slider allows precise control of time delays, with 'Coarse' and 'Fine' options determining adjustment increments.

The user can control Capture #, Number of Captures, and Number of Symbols using these spin boxes.

Read Rate is represented graphically in real-time as the user makes adjustments to the time delay between image captures.

Delay (Sec)  
0.078432

Delay Adjustment  
☐ Coarse  
☒ Medium  
☐ Fine

Decrease Increase

The Delay Adjustment slider allows the user extremely precise control of the time delays between image captures in a read cycle. Delay values are shown in seconds in the field above the slider. The Coarse and Fine Delay Adjustment options determine whether larger or smaller adjustment increments will be used.

# Pixel Binning

**Definition:** In addition to windowing the image sensor, smaller resolutions can be obtained by down sampling the entire captured image by using pixel binning. Pixel binning can increase the signal to noise ratio and produce a more pleasing output image with reduced artifacts. It will also improve low-light performance. It is important to note that enabling pixel binning does not affect the sensor frame rate as the pixels still need to be processed in order to be averaged and binned.

**Usage:** <K542,pixel binning>

**Default:** Disabled

**Options:** 0 = Disabled 1 = Enabled

## Disabled

Pixel-binning is disabled.

## Enabled

Two column pixels and two row pixels are averaged to create a single pixel value providing a 2:1 reduction in the vertical pixels and a 2:1 reduction in the horizontal pixels for a combined 4:1 reduction. An image with a dimension of 640 x 480 will be scaled to 320 x 240.

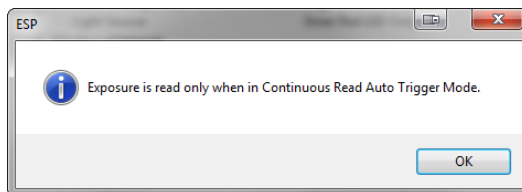
## Camera Settings

**Camera Settings** are typically obtained during the calibration process, and do not necessarily need to be modified directly by the user.

### QSXGA, SXGA, and WVGA

#### Exposure

- Usage:** Faster exposures reduce blurring in faster applications.  
Slower exposures are useful in slower applications and lower contrast applications.
- Definition:** This value sets the exposure or integration time for the image sensor pixels. The exposure setting in relation to the speed of the object is critical. If a slow exposure is selected to capture fast-moving objects, blurring or smearing of the image will occur. As exposure is increased, more light and gain become necessary, because the pixels' exposure time has decreased.
- Serial Cmd:** **<K541,exposure,gain>**
- Default:** **All versions: 2,500  $\mu$ s**
- Options:** Any number between 50 – 100,000
- Note:** The Exposure parameter is read-only when the reader is in **Continuous Read** mode.



#### Gain

- Usage:** Can be used to adjust the brightness of the image.
- Definition:** Sets the gain value for the image sensor. This setting can be configured through calibration, which is the recommended method. A higher gain value will increase the brightness of the image, but the visual noise reduction performance of the system is reduced. Before configuring gain settings, the desired exposure should already be set and the gain should be configured to optimize the exposure setting.
- Serial Cmd:** **<K541,exposure,gain>**
- Default:** **QSXGA: 33%**  
**SXGA: 0%**  
**WVGA: 33%**
- Options:** Any number between 0 – 100



## Color Filter

**Definition:** If you are using a 5 megapixel color MicroHAWK, this filter captures an image in color. To process the image for symbol decoding, the image must be filtered or the R, G, and B cells need to be equalized. This can be done by applying one of the filters available or by performing a white balance with the current camera configuration before trying to decode a symbol. This option specifies the filter method that is applied to the RGB color image to produce a monochrome image.

**Usage:** <K543,color filter>

**Default:** Disabled

**Options:** 0 = Disabled (raw, no filter)

1 = Enabled

2 = Horizontal 1D Symbols

### Disable

No filter is applied to image. If you are attempting to read symbols with no filter applied, a white balance should be performed first to balance the red, blue, and green cells of the image sensor.

### General Purpose

This is a general purpose filter that is applied to the RGB image to provide a grayscale image that can be used for most applications.

This filter should be used when reading 2D symbols, 1D symbols that are not positioned horizontally in the field of view, or a combination of 1D and 2D symbols.

### Horizontal 1D symbols

This filter is applied to the RGB image to provide a grayscale image that is specifically designed to provide the best suitable image for a horizontally-positioned 1D symbol.

This filter should be used when reading 1D symbols that are positioned horizontally across the field of view.

## **Illumination Brightness**

**Definition:** This feature allows the user to adjust the brightness of the illumination LEDs. Since the reader has control over the brightness of the illumination, it can provide consistent brightness output between readers through a factory calibration operation. Each of the brightness settings is calibrated to provide the same level of intensity for each reader.

**Serial Cmd:** **<K536,brightness>**

**Default:** **High**

**Options:** 0 = Off  
1 = Low  
2 = Medium  
**3 = High**  
4 = Constant

### ***Constant***

When set to **Constant**, the illumination is the same as the Medium setting. However, the LEDs will always be on during a read cycle and will only be off between read cycles. This cuts down on perceptible LED flashing.

## Morphological Pre-Processing

**Morphological Pre-Processing** allows the user to select the method for processing images, and to choose the operator size for that method.

**Important:** This command must be set to **Enabled** for **Morphological Operation** to function.

*Serial Cmd:*    <K550,*morphological pre-processing*>

*Default:*        **Disabled**

*Options:*        **0 = Disabled**                      1 = Enabled

# Morphological Operation and Operator Size

## *Morphological Operation*

**Definition:** **Morphological Operation** allows the user to select the method for processing captured images.

**Serial Cmd:** <K551,0,morphological operation,operator size>

**Default:** **Erode**

**Options:** **0 = Erode**

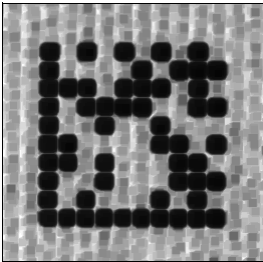
1 = Dilate

2 = Open

3 = Close

### **Erode**

**Erode** increases the dark cell size of a symbol. Useful for increasing the dark cell size of a dark-on-light Data Matrix symbol.



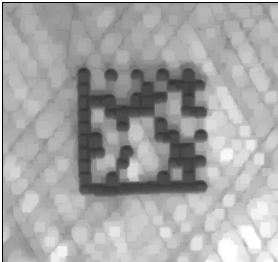
### **Dilate**

**Dilate** increases the light cell size of a symbol. Useful for increasing the light cell size of a light-on-dark Data Matrix symbol.



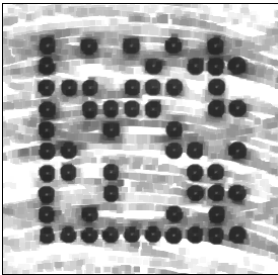
## Open

**Open** removes minor light defects of dark cells by performing a **Dilate** function followed by an **Erode** function.



## Close

**Close** removed minor dark defects of light cells by performing an **Erode** function followed by a **Dilate** function.



## **Operator Size**

*Definition:*     **Operator Size** determines the size of the area or “pixel neighborhood” (measured in pixels) in which the morphological operation is being performed.

*Serial Cmd:*    <**K551**,0,morphological operation,**operator size**>

*Default:*       **Small**

*Options:*       **3 = Small (3 pixels by 3 pixels)**  
                    5 = Medium (5 pixels by 5 pixels)  
                    7 = Large (7 pixels by 7 pixels)



# 11 Configuration Database

**Contents**

Configuration Database Serial Commands ..... 11-2

Number of Active Indexes ..... 11-3

Configuration Database Status..... 11-4

Database Mode ..... 11-9

Save Current Settings to Configuration Database..... 11-14

Load Current Settings from Configuration Database..... 11-15

Request Selected Index Settings ..... 11-16

Request All Configuration Database Settings ..... 11-16

This section concerns the various capture settings and processing settings that can be used to fine-tune the MicroHAWK ID's performance in your application.

# Configuration Database Serial Commands

Number of Active Indexes	<K252,number of active indexes,sort database>
Configuration Database Status	<K255,index,exposure,gain,pixel binning,row pointer,column pointer,row depth,column width,symbology,morphology operator, morphology size>
Save Current Settings to Database	<K255+,index>
Load Current Settings from Database	<K255-,index>
Request Selected Index Settings	<K255?,index>
Request All Database Settings	<K255?>
Database Mode	<K256,switch mode,frame count/time,image process looping,image dimensions>



# Number of Active Indexes

**Usage:** Useful for applications that require several different complex reader configurations to be applied sequentially. Multiple database indexes allow you to concatenate configuration profiles, and to perform more complex operations than would be possible with only one set of configuration parameters.

**Definition:** This feature allows you to set the number of database records (groups of settings) that will be used automatically during the read cycle. If **Number of Active Indexes** is set to 0, only the current reader settings will be used, not database entry settings.

**Serial Cmd:** <K252,number of active indexes,database sort>

**Default:** 0 (disabled)

**Options:** 0 to 10

## Database Sort

**Definition:** **Database Sort** moves the database entry that produced a successful decode to the first position in the list of database entries.

**Serial Cmd:** <K252,number of active indexes,database sort>

**Default:** Disabled

**Options:** 0 = Disabled                      1 = Enabled

## Number of Active Indexes by ESP

Click the arrow on the **Active Indexes** dropdown menu and select how many database indexes will be used during the read cycle. (0 to 10).

The screenshot shows the MicroHAWK configuration window. At the top, 'Active Indexes' is set to 5. Below this, there are sections for 'Capture Settings' and 'Processing Settings'. The 'Capture Settings' section includes fields for 'Current', 'Exposure', 'Gain', 'Pixel Binning', 'Top', 'Left', 'Height', and 'Width'. The 'Processing Settings' section includes 'Erode', 'Small', 'ECC 200', 'Morphology Operation', 'Morphology Size', and 'Symbol Type'. A table below these settings lists 10 database entries, each with a 'Test' button. At the bottom, there are checkboxes for 'Capture for Every Index', 'Show Database Index in Output', and 'Sort Index Positions on Good Reads' (which is checked). Buttons for 'Calibrate...', 'Receive Settings', 'Send Settings', 'Load Current To Index', and 'Load Index To Current' are at the very bottom.

**Sort Index Positions on Good Reads** moves the database entry that produces the first successful decode to the top of the list.

# Configuration Database Status

## Index

**Usage:** Useful for applications that require several different complex reader configurations to be applied sequentially. Multiple database indexes allow you to concatenate configuration profiles, and to perform more complex operations than would be possible with only one set of configuration parameters.

**Definition:** Determines the specific database index that will be used.

**Serial Cmd:** <**K255,index,exposure,gain,pixel binning,row pointer,column pointer,row depth,column width,symbology,morphology operator,morphology size**>

**Options:** 1 to 10

## Exposure

**Serial Cmd:** <**K255,index,exposure,gain,pixel binning,row pointer,column pointer,row depth,column width,symbology,morphology operator,morphology size**>

**Default:** **10,000**

**Options:** 50 to 100,000

**Note:** The Exposure parameter is read-only when the reader is in **Continuous Read** mode.

## Gain

**Serial Cmd:** <**K255,index,exposure,gain,pixel binning,row pointer,column pointer,row depth,column width,symbology,morphology operator,morphology size**>

**Default:** **25**

**Options:** 0 to 100

## Pixel Binning

**Serial Cmd:** <**K255,index,exposure,gain,pixel binning,row pointer,column pointer,row depth,column width,symbology,morphology operator,morphology size**>

**Default:** **Disabled**

**Options:** **0 = Disabled**                      1 = Enabled

**Important:** Pixel Binning has no effect when the **Image Dimension** mode is configured as **Region of Interest** in the **Database Mode** command. This is because the **Window of Interest** camera settings are determined by the software based on the **Region of Interest** setup. There is no benefit to Pixel Binning to increase processing speed when ROI is configured, because the frame size would need to be increased to make Pixel Binning possible.

## Row Pointer

**Definition:** The image dimension settings can be applied as a Window of Interest or a Region of Interest, depending on the image dimension mode selected in the **Database Mode** command.

**Serial Cmd:** <**K255**,index,exposure,gain,pixel binning,**row pointer**,column pointer,row depth,column width,symbology,morphology operator,morphology size>

**Default:** **All models: 0**

**Options:** **QSXGA MicroHAWK ID:** 0 to (1944 – row depth)  
**SXGA MicroHAWK ID:** 0 to (1024 – row depth)  
**WVGA MicroHAWK ID:** 0 to (480 – row depth)

## Column Pointer

**Definition:** The image dimension settings can be applied as a Window of Interest or a Region of Interest, depending on the image dimension mode selected in the **Database Mode** command.

**Serial Cmd:** <**K255**,index,exposure,gain,pixel binning,row pointer,**column pointer**,row depth,column width,symbology,morphology operator,morphology size>

**Default:** **All models: 0**

**Options:** **QSXGA MicroHAWK ID:** 0 to (2592 – column width)  
**SXGA MicroHAWK ID:** 0 to (1280 – column width)  
**WVGA MicroHAWK ID:** 0 to (752 – column width)

## Row Depth

**Definition:** The image dimension settings can be applied as a Window of Interest or a Region of Interest, depending on the image dimension mode selected in the **Database Mode** command.

**Serial Cmd:** <**K255**,index,exposure,gain,pixel binning,row pointer, column pointer,**row depth**,column width, symbology,morphology operator,morphology size>

**Default:** **QSXGA MicroHAWK ID: 1944**  
**SXGA MicroHAWK ID: 1024**  
**WVGA MicroHAWK ID: 480**

**Options:** **QSXGA MicroHAWK ID:** 3 to (1944 – row pointer)  
**SXGA MicroHAWK ID:** 3 to (1024 – row pointer)  
**WVGA MicroHAWK ID:** 3 to (480 – row pointer)

## Column Width

**Definition:** The image dimension settings can be applied as a Window of Interest or a Region of Interest, depending on the image dimension mode selected in the **Database Mode** command.

**Serial Cmd:** <**K255**,index,exposure,gain,pixel binning,row pointer,column pointer,row depth,**column width**,symbology,morphology operator,morphology size>

**Default**      **QSXGA MicroHAWK ID: 2592**

**SXGA MicroHAWK ID: 1280**

**WVGA MicroHAWK ID: 752**

**Options:**      **QSXGA MicroHAWK ID:** 8 to (2592 – column pointer)

**SXGA MicroHAWK ID:** 8 to (1280 – column pointer)

**WVGA MicroHAWK ID:** 8 to (752 – column pointer)

## Symbology

**Definition:** This field allows the user to configure the database to enable specific symbologies for selected database indexes. Symbology-specific parameters must be configured with the appropriate symbology command.

For example, if fixed length Code 128 is required, it must first be set up with the Code 128 command: <**K474**>.

To select a particular symbology, add the number value associated with that symbology.

**Examples:**

If Data Matrix and Code 39 are required, the parameter would be: 2 + 16 = **18**.

If I 2/5, BC412, and DataBar Limited are required, the parameter would be: 128 + 2048 + 16384 = **18560**.

**Serial Cmd:** <**K255**,index,exposure,gain,pixel binning,row pointer,column pointer,row depth,column width,**symbology**,morphology operator,morphology size>

**Default:**      **Disabled**

### Disabled

When **Symbology** is disabled, the database uses the current **Symbology** setup to determine active symbologies.

### Any Symbology (Add 1)

All symbologies *except Pharmacode* are enabled while this database index is being used.

### Data Matrix (Add 2)

If enabled, Data Matrix will be active for this database index.

**Important:** The ECC level must be configured using the Data Matrix command <**K479**>. If no ECC level has been configured, the reader will not decode Data Matrix symbols.

### **QR Code (Add 4)**

If enabled, QR Code will be active for this database index.

### **Code 128 (Add 8)**

If enabled, Code 128 will be active for this database index.

### **Code 39 (Add 16)**

If enabled, Code 39 will be active for this database index.

### **Codabar (Add 32)**

If enabled, Codabar will be active for this database index.

### **Code 93 (Add 64)**

If enabled, Code 93 will be active for this database index.

### **Interleaved 2 of 5 (Add 128)**

If enabled, Interleaved 2 of 5 will be active for this database index.

### **UPC/EAN (Add 256)**

If enabled, UPC/EAN will be active for this database index.

### **PDF417 (Add 512)**

If enabled, PDF417 will be active for this database index.

### **MicroPDF417 (Add 1024)**

If enabled, MicroPDF417 will be active for this database index.

### **BC412 (Add 2048)**

If enabled, BC412 will be active for this database index.

### **Pharmacode (Add 4096)**

If enabled, Pharmacode will be active for this database index.

### **DataBar-14 (Add 8192)**

If enabled, DataBar-14 will be active for this database index.

**Important:** If the stacked and non-stacked operation is required, the DataBar-14 command must be configured as follows: **<K482,2>**. If the DataBar-14 status parameter in the **<K482>** command is set to either Disabled or Enabled, the reader will only read non-stacked DataBar-14 symbols.

### **DataBar Limited (Add 16384)**

If enabled, DataBar Limited will be active for this database index.

## DataBar Expanded (Add 32768)

If enabled, DataBar Expanded will be active for this database index.

**Important:** If the stacked and non-stacked operation is required, the DataBar Expanded command must be configured as follows: **<K484,2>**. If the DataBar Expanded status parameter in the **<K484>** command is set to either Disabled or Enabled, the reader will only read non-stacked DataBar Expanded symbols.

## Micro QR Code (Add 65536)

If enabled, Micro QR Code will be active for this database index.

## Aztec (Add 131072)

If enabled, Aztec will be active for this database index.

## Postal Symbolologies (Add 262144)

If enabled, Postal Symbolologies will be active for this database index.

## Morphology Operator

**Definition:** Specifies the morphology operation (**erode**, **dilate**, **open**, **close**) used to pre-process the WOI.

**Serial Cmd:** **<K255,index,exposure,gain,pixel binning,row pointer,column pointer,row depth,column width,symbology,morphology operator,morphology size>**

**Default:** **Erode**

## Morphology Size

**Definition:** Specifies the morphology operator size to apply: **Small (3 x 3)**, **Medium (5 x 5)**, and **Large (7 x 7)**.

**Serial Cmd:** **<K255,index,exposure,gain,pixel binning,row pointer,column pointer,row depth,column width,symbology,morphology operator,morphology size>**

**Default:** **Disabled**

# Database Mode

## Switch Mode

**Definition:** Selects the event that causes the reader to load the next database entry to current, active settings. When **Frame Count/Time** expires and **Image Process Looping** is enabled, the next database entry with modifications to camera settings will be used.

**Note:** The image capture event always occurs when the first database entry is used.

**Note:** The **Switch Mode** setting has no effect on **Rapid Capture Mode**, which always operates in **Number of Image Frames** mode with a frame count of 1.

**Serial Cmd:** <K256,switch mode,frame count/time,image process looping,image dimensions>

**Default:** 1

**Options:** 0 = Time                                      1 = Number of Image Frames

## Time

When **Switch Mode** is set to **Time**, the reader will load the next database entry to current, active settings after a predefined time interval. The timer will start upon use of a database entry. If the timer expires during an image capture event, the timer will not start again until that database entry has been incremented and the new database entry has been loaded to current, active settings.

## Number of Image Frames

When **Switch Mode** is set to **Number of Image Frames**, the database entry is incremented after the predetermined number of image capture events has occurred.

## Frame Count/Time

**Definition:** Indicates the **Number of Image Frames** that must be captured or the amount of **Time** that must transpire before the reader will load the next database index entry.

**Serial Cmd:** <K256,switch mode,frame count/time,image process looping,image dimensions>

**Default:** 1 (frames/ms)

**Options:** 1 to 65535

## ***Image Process Looping***

**Usage:** Useful in applications where it is necessary to process a single captured image multiple times using different IP and decode parameters.

**Serial Cmd:** <**K256**,switch mode,frame count/time,**image process looping**,image dimensions>

**Default:** **Disabled**

**Options:** **0 = Disabled**                      1 = Enabled

### **Disabled**

When **Image Process Looping** is set to **Disabled**, an image is captured for every database configuration, whether or not any camera settings have been modified.

### **Enabled**

When **Image Process Looping** is set to **Enabled**, the last captured image frame is re-processed with the new IP and decode parameters. If camera settings have not been changed from the last capture event, and when a database configuration is loaded to current, active settings, no new image is captured.

**Note:** An exception to this is the first database index: a new image is always captured when the first database index is used. When a camera setting has been modified from one database setting to the next, a new image is captured. For example, if all database entries contained the same camera setting values but had different IP and decode parameters, an image frame would only be captured when the first database configuration was used.



## Image Dimensions

**Definition:** Determines how the image dimension parameters will be implemented.

**Serial Cmd:** <**K256**,switch mode,frame count/time,image process looping,**image dimensions**>

**Default:** 0

**Options:** 0 = **Window of Interest**      1 = Region of Interest

### Window of Interest (WOI) by ESP

When **Image Dimensions** is set to **Window of Interest**, the database image dimension parameters are *camera settings*, and they determine the size of the image to be captured.

Active Indexes: 5    ☒ Auto Sync Database settings with Reader    Advanced Options...

**Capture Settings**    << WOI    ROI >>    **Processing Settings**

	Capture Index	Exposure	Gain	Pixel Binning	Top	Left	Height	Width	Morphology Operation	Morphology Size	Symbol Type
Current	10000	25	Disable	0	0	1944	2592	...	Erode	Small	ECC 200
1	10000	20	Disable	0	0	1944	2592	...	Disable	Small	Disabled (Test)
2	10000	20	Disable	0	0	1944	2592	...	Disable	Small	Disabled (Test)
3	10000	20	Disable	0	0	1944	2592	...	Disable	Small	Disabled (Test)
4	10000	20	Disable	0	0	1944	2592	...	Disable	Small	Disabled (Test)
5	10000	20	Disable	0	0	1944	2592	...	Disable	Small	Disabled (Test)
6	10000	20	Disable	0	0	1944	2592	...	Disable	Small	Disabled (Test)
7	10000	20	Disable	0	0	1944	2592	...	Disable	Small	Disabled (Test)
8	10000	20	Disable	0	0	1944	2592	...	Disable	Small	Disabled (Test)
9	10000	20	Disable	0	0	1944	2592	...	Disable	Small	Disabled (Test)
10	10000	20	Disable	0	0	1944	2592	...	Disable	Small	Disabled (Test)

☒ Capture for Every Index    ☐ Show Database Index in Output    ☐ Sort Index Positions on Good Reads

Calibrate...    Receive Settings    Send Settings    Load Current To Index    Load Index To Current

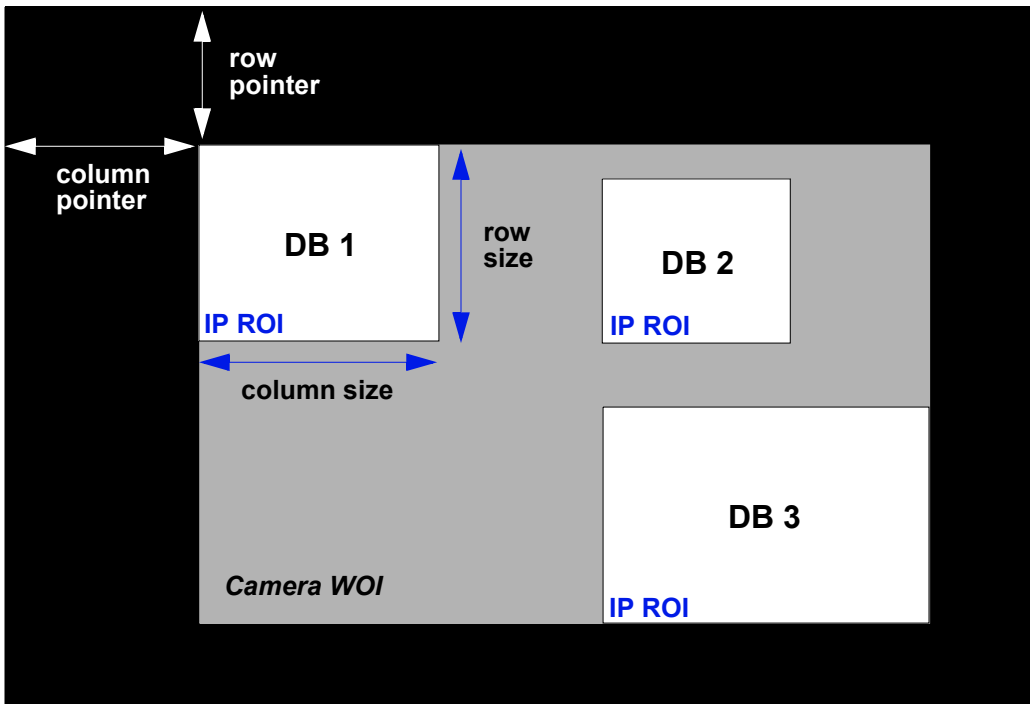
Camera settings (**Capture Index, Exposure, Gain, Pixel Binning**) located in the left half of the highlighted area.

Image capture dimensions (**Top, Left, Height, Width**) located in the right half of the highlighted area.

## Region of Interest (ROI)

When **Image Dimensions** is set to Region of Interest, the database image dimension parameters are *IP and decode settings*, and they determine the region or area of the captured image to be processed.

**ROI** coordinate data is based on the full image size. The **WOI** of the captured image will be configured to cover all the ROI settings in the current, active database entry. In the following example, there are three active database settings, each with a different ROI configuration. Their coordinates are based at point 0,0 of the full scale image. In this example, **DB1** and **DB3** determine the size of the captured image WOI, while DB2 has no impact. The image WOI is not configurable. It is automatically set up by the database according to the ROI settings.



*Full Image Size (SXGA Reader: 1280 x 1024; WVGA Reader: 752 x 480)*

**Note:** Since the **ROI** parameters are not a camera setting, a change in **ROI** parameters from one index to another does *not* indicate an image capture event.

This feature is intended to be used in conjunction with **Image Process Looping**, to allow different regions of a captured image to be processed using different IP and decode settings.

Also, if this feature is used in conjunction with **Output Filtering**, multiple decoded symbols in a captured image frame can be output according to their location in the field of view.

## Region of Interest (ROI) by ESP

When **Image Dimensions** is set to Region of Interest, the database image dimension parameters are *IP* and *decode settings*, and they determine the region or area of the captured image to be processed.

Active Indexes: 5 ☒ Auto Sync Database settings with Reader Advanced Options...

**Capture Settings**

Capture Index

Exposure

Gain

Pixel Binning

<< WOI

ROI >>

**Processing Settings**

Morphology Operation

Morphology Size

Symbol Type

Current: 10000 25 Disable 0 0 1944 2592 ... Erode Small ECC 200

<input checked="" type="radio"/> 1	10000	20	Disable	0	0	1944	2592	...	Disable	Small	Disabled	Test
<input type="radio"/> 2	10000	20	Disable	0	0	1944	2592	...	Disable	Small	Disabled	Test
<input type="radio"/> 3	10000	20	Disable	0	0	1944	2592	...	Disable	Small	Disabled	Test
<input type="radio"/> 4	10000	20	Disable	0	0	1944	2592	...	Disable	Small	Disabled	Test
<input type="radio"/> 5	10000	20	Disable	0	0	1944	2592	...	Disable	Small	Disabled	Test
<input type="radio"/> 6	10000	20	Disable	0	0	1944	2592	...	Disable	Small	Disabled	Test
<input type="radio"/> 7	10000	20	Disable	0	0	1944	2592	...	Disable	Small	Disabled	Test
<input type="radio"/> 8	10000	20	Disable	0	0	1944	2592	...	Disable	Small	Disabled	Test
<input type="radio"/> 9	10000	20	Disable	0	0	1944	2592	...	Disable	Small	Disabled	Test
<input type="radio"/> 10	10000	20	Disable	0	0	1944	2592	...	Disable	Small	Disabled	Test

☒ Capture for Every Index
 ☐ Show Database Index in Output
 ☐ Sort Index Positions on Good Reads

Calibrate...
 Receive Settings
 Send Settings
 Load Current To Index
 Load Index To Current

**Morphology Operation, Morphology Size, and Symbol Type are Processing Settings values.**

# Save Current Settings to Configuration Database

**Definition:** Allows current, active configuration settings to be saved to a selected database index.

**Serial Cmd:** <K255+,index>

**Example:**  
<K255+,5>

This command phrase saves the reader’s current, active configuration settings to database index 5.

## Save Current Settings to Configuration Database by ESP

Active Indexes 5 ☒ Auto Sync Database settings with Reader Advanced Options...

Capture Settings

Capture Index

Exposure

Gain

Pixel Binning

Top

Left

Height

Width

Processing Settings

Morphology Operation

Morphology Size

Symbol Type

Current 10000 25 Disable 0 0 1944 2592 ... Erode Small ECC 200

<input checked="" type="radio"/> 1	10000	20	Disable	0	0	1944	2592	...	Disable	Small	Disabled	Test
<input type="radio"/> 2	10000	20	Disable	0	0	1944	2592	...	Disable	Small	Disabled	Test
<input type="radio"/> 3	10000	20	Disable	0	0	1944	2592	...	Disable	Small	Disabled	Test
<input type="radio"/> 4	10000	20	Disable	0	0	1944	2592	...	Disable	Small	Disabled	Test
<input type="radio"/> 5	10000	20	Disable	0	0	1944	2592	...	Disable	Small	Disabled	Test
<input type="radio"/> 6	10000	20	Disable	0	0	1944	2592	...	Disable	Small	Disabled	Test
<input type="radio"/> 7	10000	20	Disable	0	0	1944	2592	...	Disable	Small	Disabled	Test
<input type="radio"/> 8	10000	20	Disable	0	0	1944	2592	...	Disable	Small	Disabled	Test
<input type="radio"/> 9	10000	20	Disable	0	0	1944	2592	...	Disable	Small	Disabled	Test
<input type="radio"/> 10	10000	20	Disable	0	0	1944	2592	...	Disable	Small	Disabled	Test

☒ Capture for Every Index ☐ Show Database Index in Output ☐ Sort Index Positions on Good Reads

Calibrate... Receive Settings Send Settings Load Current To Index Load Index To Current

Click the **Load Current to Index** button to save the reader’s current configuration parameters to the selected database index.

# Load Current Settings from Configuration Database

**Definition:** Allows the configuration settings contained in a selected database index to be loaded to current, active configuration settings.

**Serial Cmd:** <K255-,index>

**Example:**

<K255-,5>

This command phrase loads the configuration settings contained in database index 5 to current, active configuration settings.

## Load Current Settings from Configuration Database by ESP

The screenshot shows the ESP configuration window with 'Active Indexes' set to 5. The 'Capture Settings' and 'Processing Settings' tabs are visible. Below the settings, a table lists 10 database indices. The 'Load Index to Current' button is highlighted with a blue arrow pointing to it from a text box on the right.

Index	Current	10000	20	Disable	0	0	1944	2592	ECC	Small	Disabled	Test
1	10000	20	Disable	0	0	1944	2592	Disable	Small	Disabled	Test	
2	10000	20	Disable	0	0	1944	2592	Disable	Small	Disabled	Test	
3	10000	20	Disable	0	0	1944	2592	Disable	Small	Disabled	Test	
4	10000	20	Disable	0	0	1944	2592	Disable	Small	Disabled	Test	
5	10000	20	Disable	0	0	1944	2592	Disable	Small	Disabled	Test	
6	10000	20	Disable	0	0	1944	2592	Disable	Small	Disabled	Test	
7	10000	20	Disable	0	0	1944	2592	Disable	Small	Disabled	Test	
8	10000	20	Disable	0	0	1944	2592	Disable	Small	Disabled	Test	
9	10000	20	Disable	0	0	1944	2592	Disable	Small	Disabled	Test	
10	10000	20	Disable	0	0	1944	2592	Disable	Small	Disabled	Test	

Click the **Load Index to Current** button to load configuration settings from the selected index to current, active settings.

## Notes on Symbol Type

- The **current** DataBar Expanded status does not change if it is configured as Enabled (Stacked) and the **database** DataBar Expanded status is Enabled.
- The **current** DataBar-14 status does not change if it is configured as Enabled (Stacked) and the **database** DataBar Expanded status is Enabled.
- Data Matrix ECC level is determined by the **current** settings and not by **database** settings. Therefore, the database does not know which ECC level to enable, and has no effect on current Data Matrix ECC settings.

## **Request Selected Index Settings**

*Definition:* Returns configuration settings for the selected database index.

*Serial Cmd:* <K255?,index>

***Example:***

<K255?,5>

This command phrase returns the configuration settings for database index 5.

## **Request All Configuration Database Settings**

*Definition:* Returns configuration settings for all indexes in the Configuration Database.

*Serial Cmd:* <K255?>

***Example:***

<K255?>

This command phrase returns the configuration settings for all 10 database indexes.



# 12 *Terminal*

**Contents**

Terminal ..... 12-2

Find ..... 12-3

Send ..... 12-4

Macros..... 12-5

Terminal Menus..... 12-6

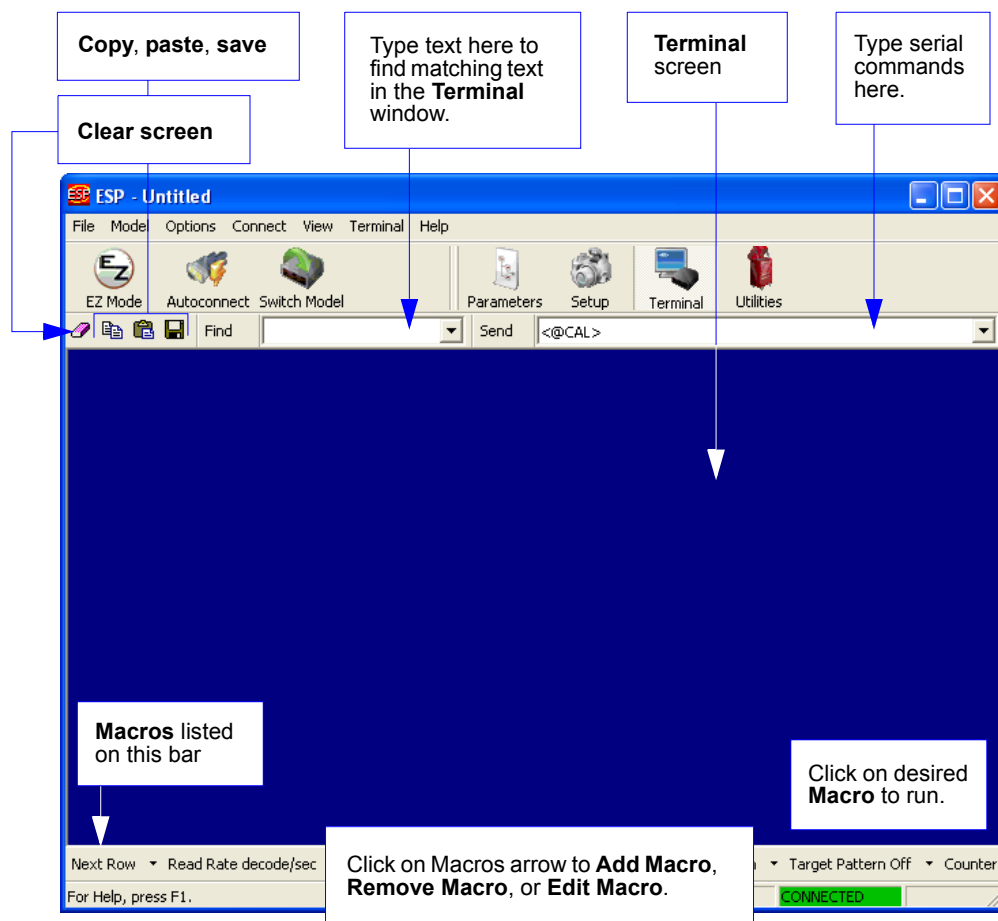
This section describes the **Terminal** window and macro functions in **ESP**.

# Terminal



Click this button to display the **Terminal** view.

This view will appear:



The **Terminal** screen allows you to send serial commands to the reader by using Macros, by copying and pasting, or by typing commands in the **Send** text field.

The **Terminal** screen also displays symbol data or information from the reader.

You can also right-click on the **Terminal** screen to bring up a menu of further options.



# Find

The **Find** function allows you to enter text strings to be searched for in the **Terminal** window. For example, a series of symbols have been scanned into the **Terminal** view and you want to determine if a particular symbol whose data begins with “ABC” has been read.

1. Type “ABC” into the **Find** box.



2. Press **Enter**.

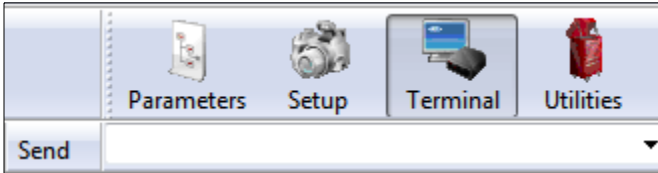
The first instance of “ABC” will be highlighted in the **Terminal** window.

3. Click the **Find** button to the left of the text field to locate additional instances of “ABC”.

## Send

The **Send** function allows you to enter serial commands and then send them to the reader.

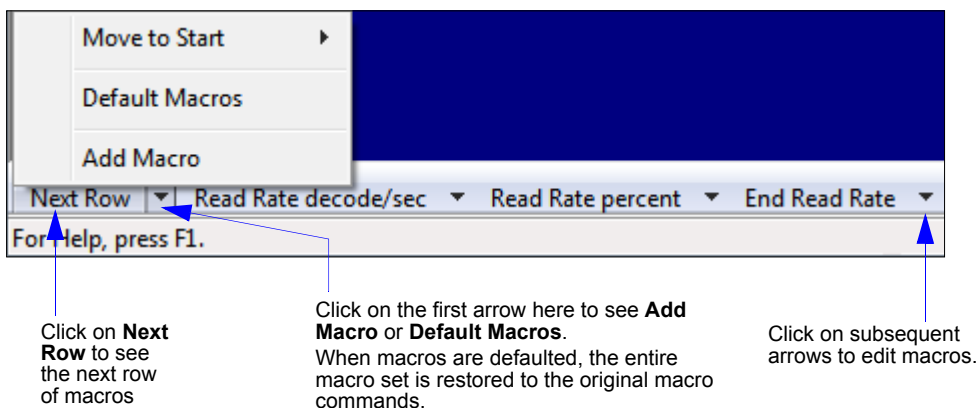
1. Type the command into the **Send** box.



2. Press **Enter**.
3. Click the **Send** button to the left of the text field to send .

# Macros

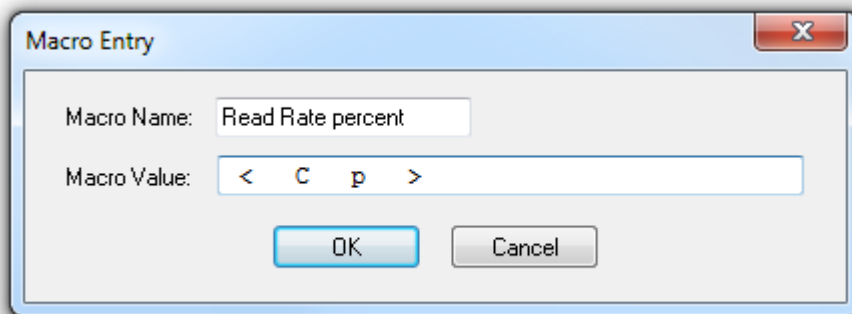
Macros can be stored in a macro selection bar, edited in a separate window, and executed by clicking on the macro name.



When you click on the macro name, the macro is executed in the **Terminal** window. If this is a command, it is sent to the reader at the same time that it is displayed.

## Editing a Macro

When you click the arrow next to any macro and select **Edit**, the following appears:



You can edit an existing macro or type in the **Macro Name** text field and define it in the **Macro Value** text field. Click **OK**.

# Terminal Menus

Right-click on the **Terminal** window to display the following menu:

Copy
Paste
Clear
Select All
Save...
Change Font...
Change Echo Font...
Enable Echo
Change Background Color ▶
Non-Printable Characters ▶
Default Settings
Keyboard Macros...

- **Copy** selected text to clipboard.
- **Paste** from Terminal or other text.
- **Clear** all text in Terminal window.
- **Select All** text in the Terminal window.
- **Save...** brings up a **Save As** dialog.
- **Change Font...** of text in Terminal; brings up a **Font** dialog.
- **Change Echo Font...** to change typed text; brings up a **Font** dialog.
- **Enable Echo** enables Echo text (typed by user).
- **Change Background Color** of Terminal window.
- **Non-Printable Characters** allows you to hide non-printable characters, or to show them in **Standard** or **Enhanced** format.
- **Default Settings** returns all the above settings to default.
- **Keyboard Macros** allows you to create new keyboard macro commands that can be sent from function keys (F2, F4, F5, etc.)

## Terminal Dropdown Menu

The dropdown **Terminal** menu has **Capture Text**, **Save Current Text**, **Send File**, **Find Next**, and **Find Previous** functions, as well as the same functions defined above.

Terminal	
Capture Text...	
Save Current Text...	
Send File...	
Change Font...	
Change Echo Font...	
Enable Echo	
Change Background Color	▶
Non-Printable Characters	▶
Default Settings	
Find Next	F3
Find Previous	Shift+F3
Keyboard Macros...	

- **Capture Text...** lets you append data in real time to a text file of your choice. While in operation, the text file cannot be opened. You can select Pause to interrupt the capture flow or Stop to end the flow and open the file.
- **Save Current Text...** saves all text in the Terminal window to a text file.
- **Send File** allows you to browse for specific files and send them to the reader.
- **Find Next** searches for a user-defined section of text in the Terminal.
- **Find Previous** operates in the same way as Find Next, but searches backward through Terminal text.

# 13 Utilities

## Contents

Serial Utility Commands .....	13-2
Read Rate .....	13-4
Counters .....	13-6
Device Control .....	13-8
Differences from Default .....	13-9
Master Database .....	13-10
Firmware .....	13-16
Default/Reset/Save .....	13-19
Reader Status Requests .....	13-21
Other Operational Serial Commands .....	13-23

Utility commands are generally commands that are performed during reader operations to check or determine read rates, or to perform miscellaneous operations on reader hardware. Serial utility commands are not prefaced with a “K” and a numeric code, nor do they require an initialization command (<A> and <Z>). They can be entered from within any terminal program or from **ESP’s Terminal** or **Utilities** windows.

# Serial Utility Commands

**Note:** For a list of all K commands, see **Serial Configuration Commands**.

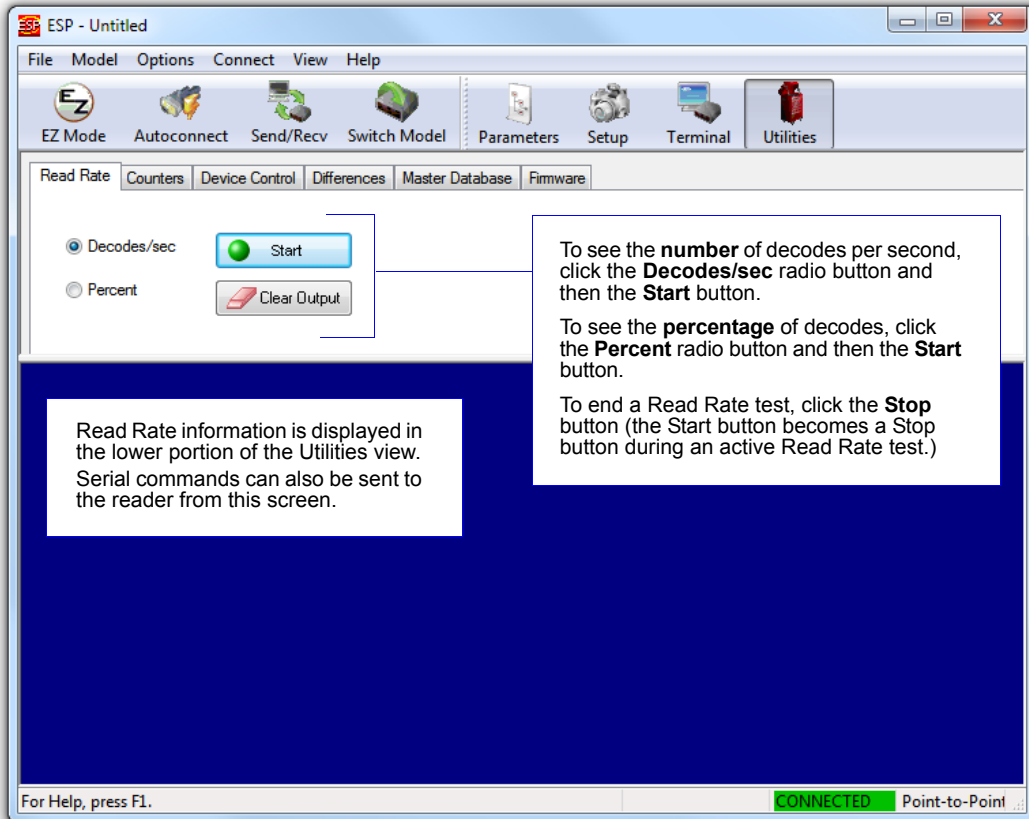
Type	Command	Name
Read Rate	<C>	Enter Decode Rate Test
	<Cp>	Enter Percent Rate Test
	<J>	Exit Decode Rate and Percent Rate Test
Counter Request and Clear	<N>	No Read Counter
	<O>	No Read Counter Reset
	<T>	Trigger Counter
	<U>	Trigger Counter Reset
	<V>	Good Read/Match Counter
	<W>	Good Read/Match Counter Reset
	<X>	Mismatch Counter
	<Y>	Mismatch Counter Reset
Master Database	<G>	Store Next Symbol Read to Database Index 1
	<Gn>	Store Next Symbol Read to Database Index <i>n</i>
	<NEWM>	New Master Load Status
Part Number / Checksum	<#>	Display All Firmware Part Numbers
	<#a>	Display Application Code Part Number
	<#b>	Display Boot Code Part Number
	<#p>	Display Profile Module Part Numbers
	<!>	Display All Available Firmware Checksums
	<!a>	Display Application Code Code Checksum
	<!b>	Display Boot Code Checksum
	<!p>	Display Profile Module Checksum
Device Control	<L1>	Programmable Output 1
	<L2>	Programmable Output 2
	<L3>	Programmable Output 3
	<I1>	Targeting System On
	<I0>	Targeting System Off
	<I>	Disable Reader
	<H>	Enable Reader
	<a1>	Include PDF Information in Preamble (PDF417)
Default/Reset/Save	<A>	Reset (does not save for power-on)
	<Ard>	Reset and Recall Microscan Defaults
	<Arp>	Reset and Recall Power-On Parameters
	<Arc>	Reset and Recall Customer Default Parameters
	<Z>	Save Current Settings for Power-On

<b>Default/Reset/Save</b>	<b>&lt;Zc&gt;</b>	Save Current Settings as Customer Default Parameters for Power-On
	<b>&lt;Zrd&gt;</b>	Recall Microscan Default Parameters and Save for Power-On
	<b>&lt;Zrc&gt;</b>	Recall Customer Default Parameters and Save for Power-On
<b>Reader Status Request Commands</b>	<b>&lt;?&gt;</b>	Reader Status Byte
	<b>&lt;K?&gt;</b>	All Status Request
	<b>&lt;K??&gt;</b>	All Descriptor Status Request
	<b>&lt;K?#&gt;</b>	All Range Status Request
	<b>&lt;Knnn?&gt;</b>	Single Status Request
	<b>&lt;Knnn??&gt;</b>	Single Descriptor Status Request
	<b>&lt;Knnn?#&gt;</b>	Single Range Status Request
<b>Train / Optimize</b>	<b>&lt;TRAIN&gt;</b>	Initiate Train Operation
	<b>&lt;UNTRAIN&gt;</b>	Initiate Untrain Operation
	<b>&lt;TRAIN?&gt;</b>	Train Status Request
	<b>&lt;OPT&gt;</b>	Initiate Optimize Operation
	<b>&lt;UNOPT&gt;</b>	Initiate Un-Optimize Operation
	<b>&lt;OPT?&gt;</b>	Display Optimize Status
<b>Other Operational Commands</b>	<b>&lt;uy,path,filename&gt;</b>	Y-Modem Upload Transfer Options
	<b>&lt;dy,path,filename&gt;</b>	Y-Modem Download Transfer Options
	<b>&lt;op,9&gt;</b>	Image Library Request
	<b>&lt;BCCFG&gt;</b>	Enter Bar Code Configuration
	<b>&lt;VAL&gt;</b>	Static Validation

# Read Rate



Click the **Utilities** button and then the **Read Rate** tab to display the Read Rate view.



Read Rate information is displayed in the lower portion of the Utilities view. Serial commands can also be sent to the reader from this screen.

To see the **number** of decodes per second, click the **Decodes/sec** radio button and then the **Start** button.

To see the **percentage** of decodes, click the **Percent** radio button and then the **Start** button.

To end a Read Rate test, click the **Stop** button (the Start button becomes a Stop button during an active Read Rate test.)



## ***Read Rate Serial Utility Commands***

### **Enter Decodes/Second Test**

Sending **<C>** instructs the reader to transmit the decodes per second and symbol data (if any). The decode rate can vary dramatically due to the angle and location of the symbol in relation to the field of view. This test is very useful in aligning and positioning the reader during setup.

### **Enter Percent Test**

Sending **<Cp>** instructs the reader to transmit the percentage of decodes and any decoded symbol data.

### **End Read Rate Test**

Sending **<J>** ends both the Percent test and the Decodes/Second test.

# Counters



Click the **Utilities** button and then the **Counters** tab to display the Counters view.

**Counter** commands can be a numeric value from **00000** to **65535**. After reaching the maximum numeric limit of 65535, an error message will be displayed and the counter will automatically roll over and start counting again at 00000. To obtain the cumulative total of counts after the rollover has occurred, add 65536 per each rollover (the reader does not keep track of the number of rollovers) to the current count.

**Note:** All counter values will be lost if power is cycled, or if the reader receives a **Reset** or **Save** command.

Click the **Request** button to display the appropriate count or **Clear** to set the counter to zero.

Read Rate

Counters

Device Control

Differences

Master Database

Firmware

Request All

Clear All

Request

Clear

Trigger:

Request

Clear

Good Read:

Request

Clear

No Read:

Request

Clear

Mismatch:

Trigger, Good Read, No Read, and Mismatch counters can be requested and cleared simultaneously (**Request All** and **Clear All** buttons) or individually (**Request** and **Clear** buttons to the left of each item).

## Counters by Serial Command

### No Read Counter

Sending <N> displays the total number of No Reads that have occurred since the last reset.

### No Read Counter Reset

Sending <O> sets the No Read Counter to 00000.

### Trigger Counter

Sending <T> displays the total number of triggers since the last reset.

### Trigger Counter Reset

Sending <U> sets the trigger counter to 00000.

### Good Read/Match Counter (or Good Read Counter)

Sending <V> displays the total number of good reads matching the master symbol, or, if **Master Symbol** is not enabled, the number of good reads since the last reset. This counter is always enabled, but will only work as a match count when **Master Symbol** is enabled. If **Master Symbol** is not enabled, this counter records the number of good reads. This count can be requested at any time.

### Good Read/Match Counter Reset

Sending <W> sets the Match Counter to 00000.

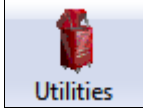
### Mismatch Counter

Sending <X> displays the number of decoded symbols since the last reset that do not match the master symbol.

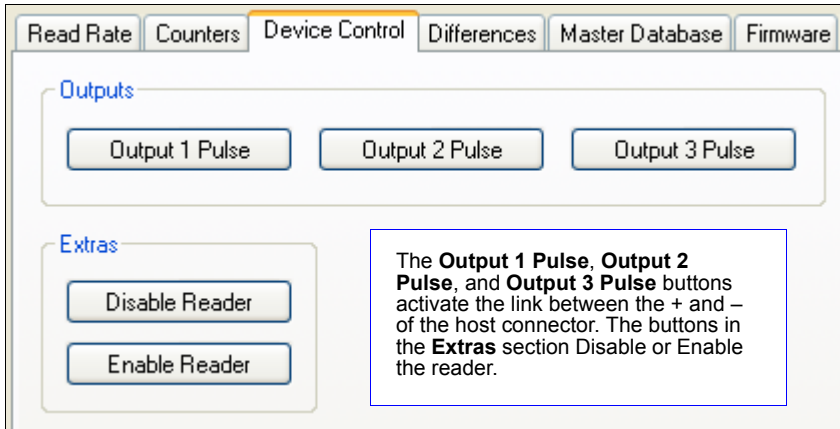
### Mismatch Counter Reset

Sending <Y> sets the Mismatch Counter to zero.

# Device Control



Click the **Utilities** button and then the **Device Control** tab to display the Device Control view.



## Device Control by Serial Command

### Output 1 Pulse

Sending **<L1>** activates the link between Output 1 (+) and Output 1 (–) of the host connector (regardless of Master Symbol or Output 1 status).

### Output 2 Pulse

Sending **<L2>** activates the link between Output 2 (+) and Output 2 (–) of the host connector (regardless of Master Symbol or Output 2 status).

### Output 3 Pulse

Sending **<L3>** activates the link between Output 3 (+) and Output 3 (–) of the host connector (regardless of Master Symbol or Output 3 status).

### Disable Reader

Sending **<I>** will turn the reader OFF, end the current read cycle, and will not allow the reader to enter another read cycle until turned ON. This feature is useful during extended periods of time when no symbols are being decoded, or the reader is being configured. Disabling the reader will not affect any commands that have already been downloaded.

### Enable Reader

Sending **<H>** will turn the reader ON and allow it to enter read cycles.

## Differences from Default



Click the **Utilities** button and then the **Differences** tab to display the Differences from Default view.

Clicking the **Differences from Default** button will cause **ESP** to check all stored configuration settings and compare them to default settings. All settings that are different than default will appear in the left column (shown below), and descriptions of those settings will appear in the right column.

Click this button for a list of **ESP** configuration settings that are different than default settings.

Click **Generate Barcode** to display the **Bar Code Dialog**. Then create configuration symbols containing the required configuration commands.

Command	Description
<K200,3>	Mode (External Edge)
<K220,,50>	Read Cycle Timeout (50)
<K474h,2>	Code 128 (Edge to Edge)
<K810,3>	Output On (No Read)
<K811,1>	Output On (Good Read/Match)
<K812,2>	Output On (Mismatch)

Send configuration settings to the reader without saving by clicking **Send to Reader**.

Click **Save As** to save the report as plain text or a tab-delimited text file.

Send configuration settings to the reader and save in ESP by clicking **Send and Save**.

- To create a symbol containing any of the command settings in the table, click **Generate Barcode**.
- To save the **Differences from Default** report, either as plain text or as a tab-delimited text file, click **Save As**.
- Click **Send and Save** to send the settings to the reader and save them, or **Send to Reader** to send the settings without saving them.

**Important:** To use **Differences from Default**, connect to the reader and **Receive Reader Settings** via the **Send/Recv** button on the toolbar.

# Master Database



Click the **Utilities** button and then the **Master Database** tab to display the Master Database view.

**Important:** The **Master Database** is used for all **Matchcode** modes except **Sequential** and **Wild Card**, both of which use **Master Database Index # 1**.

## Master Database Overview

Used where more than one master symbol is required, as in a **Multisymbol** setup, for matching and other **Matchcode** operations.

Allows the user to define up to 10 master symbols as the master symbol database, which can be entered by keyboard, scanned in, displayed, or deleted by serial or **ESP** commands.

1. Click the **Master Database** tab.
2. Enable **Matchcode Type**.
3. Set the **Master Symbol Database Size**.
4. Select the database index in which the master symbol will be entered.
5. Do one of the following to enter master symbol data:
  - a. Double-click the index row to type data directly into the index.
  - b. Click the **Read Symbol into Selected Index** to enter the next decoded symbol.

Set **Master Symbol Database Size** here.

Loads the next symbol decoded into the selected index.

Loads the reader's saved master symbols into **ESP**.

Saves the database to the reader.

Double-click a row to enter data into popup text box.

Caution: Since the total number of characters available for the master symbol database is limited, changes to the Master Symbol Database Size will re-allocate the number of characters available for each master symbol and could cause existing master symbols to be deleted. See manual for details.

## Master Symbol Database Size

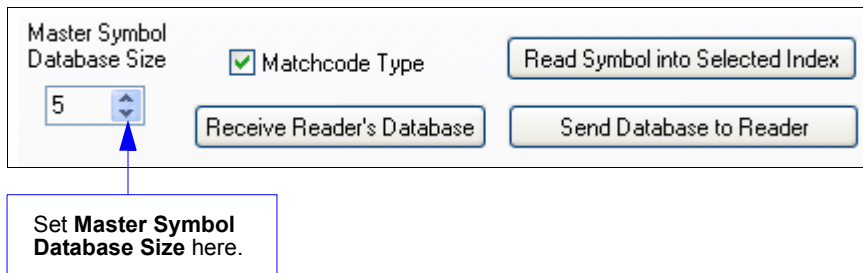
**Definition:** **Number of Master Symbols** allows you to select **1 to 10** master symbols for the master symbol database.

**Serial Cmd:** **<K231,master symbol database size>**

**Note:** You must follow this command with a save command **<A>** or **<Z>**.

**Default:** **1**

**Options:** 1 to 10



Master Symbol Database Size

☒ Matchcode Type

Read Symbol into Selected Index

5

Receive Reader's Database

Send Database to Reader

Set Master Symbol Database Size here.

**Caution:** Since the total number of characters available for the master symbol database is **3000**, changes to the **Master Symbol Database Size** will re-allocate the number of characters available for each master symbol and could cause existing master symbols to be deleted (except master symbol #1, unless it also exceeds the size limitation).

The table below specifies the maximum number of characters available to each symbol according the number of master symbols defined, from 1 to 10.

<b>Master Symbol Number</b>	<b>Maximum Characters</b>	<b>Master Symbol Number</b>	<b>Maximum Characters</b>
# 1	3000	# 6	500
# 2	1500	# 7	428
# 3	1000	# 8	375
# 4	750	# 9	333
# 5	600	# 10	300

## Enter Master Symbol Data

**Usage:** Allows you to enter master symbol data for any enabled master symbol index number (1 to 10), provided the total number of characters does not exceed the maximum allowed.

**Serial Cmd:** <K231, master symbol number, master symbol data>

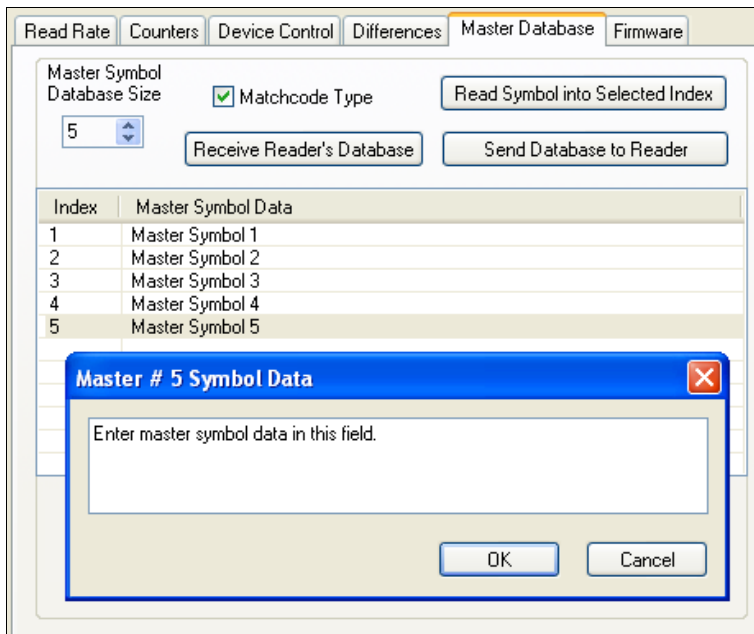
**Options:** Enter data for 1 to 10 symbols (any combination of ASCII text up to the maximum allowed).

For example, to enter data for master symbol 9, after making certain that master symbol database size is enabled for 9 or more symbols, send <K231,9,data>.

**Important:** The ASCII characters <, >, and , can only be entered as hex pairs.

**Caution:** If no data is entered, the existing data will be deleted.

- ESP:**
1. Open the **Utilities** menu.
  2. Set the number of master symbols you want to create in **Master Symbol Database Size**.
  3. Double-click on each symbol number you want to set up and copy, or type your data in the popup dialog and click **OK**.



4. When all your data has been entered, click the **Send Database to the Reader** button.



## Request Master Symbol Data

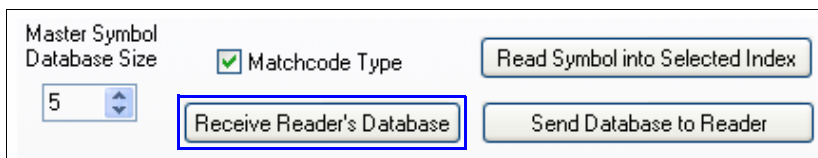
**Definition:** Returns master symbol data for any enabled master symbols from 1 to 10. For example, to request master symbol # 5, enter **<K231?,5>**. The reader transmits master symbol # 5 data in brackets in the following format: **<5/>**. If no master symbol data is available, the output will be: **<5/>**.

**Serial Cmd:** **<K231?,master symbol number>**

**Caution:** Be sure to add the **?** or you will delete the master symbol.

**Note:** This command returns the number of master symbols if no number is included.

- ESP:**
1. Click the **Utilities** button and the **Master Database** tab.
  2. Click the **Receive Reader's Database** button.



Master Symbol Database Size

☒ Matchcode Type

5

Receive Reader's Database

Read Symbol into Selected Index

Send Database to Reader

## Request All Master Symbol Data

**Definition:** This command will return master symbol data for all symbols enabled (up to 10).

**Serial Cmd:** **<K231?>**

---

## Read Next Symbol as Master Symbol

**Definition:** After you've set the size in the database, you can order the reader to read the next symbol as the master symbol for any given master symbol number.

**Serial Cmd:** **<Gmaster symbol number>**

To store the next symbol decoded as master symbol # 1, send:

**<G>** or **<G1>**.

To store the next symbol decoded as the master symbol for any other master symbol database number, send:

**<Gmaster symbol number [1-10]>**.

For example, **<G5>** will cause the next symbol read to be entered as master symbol # 5.

**ESP:** In the **Master Database** tab under the **Output Format** menu,

1. Select the master symbol index number in which you want to store the symbol data.
2. Click the **Read Symbol into Selected Index** button.

**Caution:** If you've selected an index which already has existing data, that data will be copied over by new decoded data when you use this command.

## Request New Master Status

**Usage:** Informs the user when a new master symbol is pending and which position it is in.

**Definition:** Returns the position in the master symbol database that will be loaded on the next read.

**Serial Cmd:** **<NEWM>**

The reader returns: **<NEWM/next master to load>**

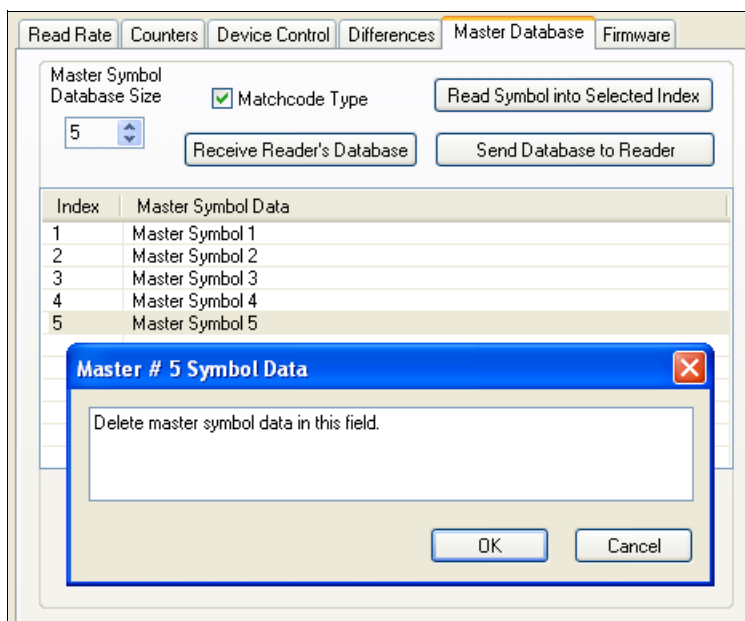
Once a symbol has been read and loaded, the status will be cleared and the response will be **<NEWM/0>**.

(See also **New Master Pin** in **Matchcode**.)

## Delete Master Symbol Data

**Definition:** You can directly delete the master symbol data by serial command or by **ESP**.

- ESP:**
1. Click the **Utilities** button to access the master symbol.
  2. Click the **Master Database** tab and double-click the symbol number you want to delete.
  3. Delete text and click **OK**.



**Serial Cmd:** **<K231, master symbol number,>**

To delete a master symbol, enter the database number and a comma, but leave the data field empty. For example, to delete master symbol # 5, send the following: **<K231,5,>**. The command is entered with a blank master symbol data field, which tells the reader to delete the selected master symbol from the database.

# Firmware



Click the **Utilities** button and then the **Firmware** tab to display the Firmware view.

## Firmware Update

Application code versions are specific to your reader. Consult with your sales representative before downloading application code. If needed, an application code will be sent to you in the form of a **\*.mot** file.

### To download application code:

1. First, be sure that your reader is connected to the host.
2. Apply power to the reader.
3. Before updating, you should verify the current firmware.
4. Click in the Firmware Update text box and select the file type you want to download. This will open a file locator box.
5. Navigate to the appropriate file (it will be a **\*.mot** file), open the file and click **Start**.

**Caution:** Do not interrupt power or disconnect the host cable while download is in progress. Be sure that each download is complete before moving on to the next.

**Important:** When updating firmware, be sure that the application code and boot code versions are mutually compatible.

## Firmware Verification

### ***Request Part Number by ESP***

You can send a request to the reader for part numbers, checksums, boot code, and application code.

1. Click the **Firmware** tab.
2. From the dropdown selection box to the left of the **Request Part No.** button, make your selection.
3. Select the code type to see its part number displayed in the text field to the right of the **Request Part No.** button.

The screenshot shows a window titled "Firmware Verification". Inside, there are two rows of controls. The top row consists of a dropdown menu with "App Code" selected, a button labeled "Request Part No." (which is highlighted with a blue border), and an empty text input field. The bottom row consists of a dropdown menu with "App Code" selected, a button labeled "Request Checksum", and another empty text input field.

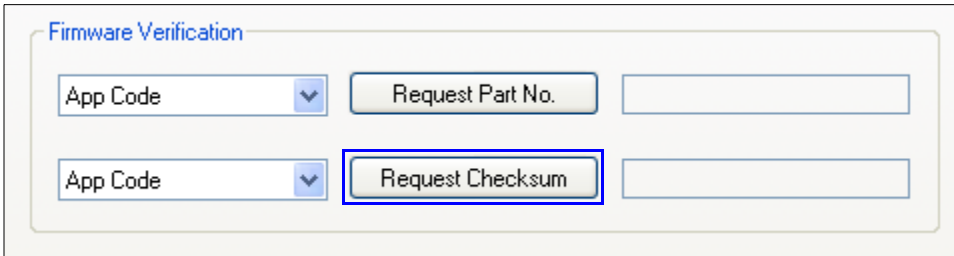
### ***Request Part Number by Serial Command***

- When you send **<#>** (a request for all product part numbers), the reader returns:  
**<#b/BOOT\_P/N><#a/APP\_P/N><#p/PROFILE\_P/N>**.
- When you send **<#a>** (a request for the application code part number), the reader returns:  
**<#a/APP\_P/N>**.
- When you send **<#b>** (a request for the boot code part number), the reader returns:  
**<#b/BOOT\_P/N>**.
- When you send **<#p>** (a request for profile module part numbers), the reader returns:  
**<#p/PROFILE\_P/N>**.

### **Request Checksum by ESP**

You can send a request to the reader for part numbers, checksums, boot code, and application code.

1. Click the **Firmware** tab.
2. From the dropdown selection box to the left of the **Request Checksum** button, make your selection.
3. Select the code type to see its checksum displayed in the text field to the right of the **Request Checksum** button.



The screenshot shows a 'Firmware Verification' window. It contains two rows of controls. The top row has a dropdown menu labeled 'App Code', a button labeled 'Request Part No.', and an empty text field. The bottom row has a dropdown menu labeled 'App Code', a button labeled 'Request Checksum' (which is highlighted with a blue border), and an empty text field.

### **Request Checksum by Serial Command**

- When you send `<!>` (a request for all available firmware checksums), the reader returns:  
`<!b/BOOT_CHECKSUM><!a/APP_CHECKSUM><!p/PROFILE_CHECKSUM>`
- When you send `<!a>` (a request for the application code checksum), the reader returns:  
`<!a/APP_CHECKSUM>`
- When you send `<!b>` (a request for the boot code checksum), the reader returns:  
`<!b/BOOT_CHECKSUM>`
- When you send `<!p>` (a request for profile module checksum), the reader returns:  
`<!p/PROFILE_CHECKSUM>`

# Default/Reset/Save

Understanding and controlling your reader's active, saved, and default settings is critical to its successful operation.

	<i><b>Function</b></i>	<i><b>Serial Cmd</b></i>	<i><b>ESP (first, right-click in any menu tree)<sup>a</sup></b></i>
<b>Resets</b> <i>(not saved for power-on)</i>	Reset	<A>	Save to Reader, Send No Save
	Reset and Recall Microscan Defaults	<Ard>	No
	Reset and Recall Power-On Parameters	<Arp>	No
	Reset and Recall Customer Default Parameters	<Arc>	No
<b>Saved for Power-on</b>	Save Current Settings for Power-On	<Z>	Save to Reader, Send and Save
	Save Current Settings as Customer Default Parameters for Power-On	<Zc>	Save to Reader, Send and Save Customer Defaults for Power-On <sup>b</sup>
	Recall Microscan Default Parameters and Save for Power-On	<Zrd>	No
	Recall Customer Default Parameters and Save for Power-On	<Zrc>	No

a. When you right-click in a menu tree and select **Default Current Menu Settings** or **Default All ESP Settings**, it is important to note that only **ESP** settings are defaulted. To save these defaults to the reader itself, you need to follow up with a **Save to Reader, Send and Save** command.

b. Only available in ESP if enabled under the Options dropdown menu.

## Resets

Resets (“A” commands) affect only the current settings (active memory) and are not saved for power-on.

## Saved for Power-on

Power-on parameters (“Z” commands) are saved to NOVRAM and recalled and loaded into current parameters when power is cycled to the reader or the <Arp> command is issued.

## Defaults

Defaults are Microscan firmware settings or saved customer settings that can be recalled, either by software or hardware reset.

## **Customer Default Parameters**

Customer default parameters (saved by **<Zc>**) are the same set of parameters as power-on parameters but are saved in a different, isolated section of NOVRAM. This allows a user essentially to create a backup set of parameters that can be recalled in the event that the current parameters or power-on parameters have been accidentally changed or are no longer desired.

It is important to note that a hardware default does not affect customer default parameters. For example, a user that has inadvertently changed communication settings and saved them with a **<Z>** command, may not know the correct settings or have the capability to communicate within those settings. By first doing a hardware default to restore the known Microscan defaults, the user can then recall the previous customer saved settings with an **<Arc>** or **<Zrc>** command.

## **Microscan Default Parameters**

Microscan default parameters are contained in the firmware and cannot be changed.

### **Software Defaults**

Microscan default parameters can be recalled (loaded into current settings) with **<Ard>** command or recalled and saved for power-on with the **<Zrd>** command.

### **Hardware Default**

If a software default and reset is not possible, it may be necessary to reset the reader by shorting (connecting) specific pins. This procedure has the same effect as the **<Zrd>** software command.

**Important:** For this reset to occur, this command must be executed within 60 seconds of a power-on or a reset.

1. Apply power to the reader.
2. Locate Pin 3 (Default) and Pin 7 (Ground) on the connector. (Access depends on wiring configuration. May require an accessory cable).

**Caution:** Be certain that the correct pins are located. Connecting the wrong pins could cause serious damage to the unit.

3. Momentarily connect these wires (or pins) and listen for a series of short beeps.
4. Within 3 seconds, connect them again. A longer beep should be heard. If not, repeat the process.



# Reader Status Requests

## <?> Status Byte

The reader responds to a status request <?> with a two character hex value, such as <?/22>. To determine status:

1. Look up the binary conversion in the table below.

For example, the first hex **2** would be **0 0 1 0** in binary, as read from binary digits **3** through **0**; the second hex **2** is also **0 0 1 0** as read from binary digits **7** through **4**.

2. Next, enter your binary values from the table below in the "Binary" column next to the appropriate bit.

Bit	Binary	Reader Status
0	0	Command error detected
1	1	Command received
2	0	Communication error detected
3	0	Flash sector unprotect failure
4	0	Host port buffer overflow
5	1	Reader is in a read cycle

3. Under "Binary," interpret **1s** as true and **0s** as not true. For example, bit 1 has a **1** in the "Binary" column, indicating "Command Received". Bit 5 is also a **1**, indicating that the "Reader is in a read cycle".

Hex Value	Binary Bit Digits			
	7	6	5	4
	3	2	1	0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
A	1	0	1	0
B	1	0	1	1
C	1	1	0	0
D	1	1	0	1
E	1	1	1	0
F	1	1	1	1

## <K?> All Status Request

This is the fastest way to learn the reader's current configuration. Sending this request will return the current settings of all commands, starting with the lowest K command value and ending with the highest K command value.

## <K??> All Descriptor Status Request

This request will return all current descriptors for every K command, starting with the lowest K command value and ending with the highest K command value.

## <K?#> All Range Status Request

This request will return the current settings of all commands within the user-defined range, starting with the lowest user-defined K command value and ending with the highest user-defined K command value.

### ***<Knnn?> Single Status Request***

This request will return the value of the variables associated with the requested K command. The request of a single entry of a database command cannot exceed the number of database slots for the specific command.

### ***<Knnn??> Single Descriptor Status Request***

This request returns the basic functional description of all fields in the requested K command.

### ***<Knnn?#> Single Range Status Request***

This request will return the value range and storage type description of all fields in the requested K command.

### ***<Knnn?\*> Display Command Wildcard***

This request will return the individual K command status, description, and range for each parameter.

# Other Operational Serial Commands

The following serial utility commands can be entered from **ESP's Terminal** or a PLC:

## *Train and Optimize*

### Train Symbol

Format: **<TRAIN>**: Initiates a Train operation.

This command will put the product into a mode of operation that will cause it to “train” the next symbol decoded. This mode of operation will remain active until either a symbol is decoded or the call is made to disable the mode and revert back to normal operation. Upon decoding a symbol, the image processing will save pertinent information regarding the target symbol to allow higher readability for the similar symbols.

The data collected by the Train operation can be saved for a power-on condition, using the **<Z>** command.

### Un-Train Symbol

Format: **<UNTRAIN>**: Initiates an Un-Train operation.

This command will cause the product to discard any information acquired during a Train operation. The Untrained state can be saved for a power-on condition, using the **<Z>** command.

### Train Status

Format: **<TRAIN?>**: Responds with Train Status

This command will return the current status of the TRAIN operation.

Response: **< TRAIN,0>**: Default, not trained

**<TRAIN,1>**: Train operation in process

**<TRAIN,2>**: Symbol trained

### Train Persistence

The train state and parameters persist the same as ordinary parameters. Examples:

- A unit in the trained state has not been saved. A power cycle will remove any trained state information and the unit will come up in its configured state.
- A unit is saved in a training state. The unit will come from power up in the training state and train the first symbol read.
- A unit is saved in a trained state. The unit will come from power up in the trained state and only read the trained symbol type.
- A trained unit is issued a reset default command **<Ard>**. The unit will return to an untrained state but if power is cycled will return to its saved state.

## Optimize Symbol

Format: **<OPT>** Initiates an Optimize operation.

This command will put the product into a mode of operation that will cause it to “optimize” the next symbol decoded. This mode of operation will remain active until either a symbol is decoded or the call is made to disable the mode and revert back to normal operation.

Upon decoding a symbol, the image processing will save pertinent information regarding the target symbol to allow it to be processed quicker and more consistently.

The data collected by the Optimize operation can be saved for a power-on condition, using the **<Z>** command.

## Un-Optimize Symbol

Format: **<UNOPT>**: Initiates an Un-Optimize operation.

This command will cause the product to discard any information acquired during a Optimize operation. The Un-Optimized state can be saved for a power-on condition, using the **<Z>** command.

## Optimize Status

Format: **<OPT?>**: Responds with Optimize Status

This command will return the current status of the OPT operation.

Response: **<OPT,0>**: Default, not optimized

**<OPT,1>**: Optimize operation in-process

**<OPT,2>**: Symbol optimized

## Optimize Persistence

The Optimize state and parameters persist the same as ordinary parameters. Examples:

- A unit in the optimized state has not been saved. A power cycle will remove any optimized state information and the unit will come up in its configured state.
- A unit is saved in a optimizing state. The unit will come from power up in the optimizing state and optimize the first symbol read.
- A unit is saved in a optimized state. The unit will come from power up in the optimized state and only read the optimized symbol type.
- A optimized unit is issued a reset default command **<Ard>**. The unit will return to an un-optimized state but if power is cycled will return to its saved state.

## ***Y-Modem Upload Transfer Options***

**<uy,path,filename>**

## ***Y-Modem Download Transfer Options***

**<dy,path,filename>**

## ***Image Library Request***

**<op,9>** Manages files in a selected directory.

<b><i>File Source</i></b>	<b><i>Explanation</i></b>
<b>(Nothing)</b>	All files in “root” directory
<b>/</b>	All files in “root” directory
<b>/saved</b>	All files in “saved” directory
<b>*.*</b>	All files in all directories
<b>/del</b>	Deletes all files in the root director
<b>/saved/del</b>	Deletes all files in the saved directory
<b>del*.*</b>	Deletes files in all directories

## Bar Code Configuration Mode

**Definition:** **Bar Code Configuration Mode** is a way of programming the reader by using ECC 200 Data Matrix symbols.

**Serial Cmd:** **<BCCFG>**

**Bar Code Configuration Mode** can be entered in two different ways:

1. By forcing the reader into **Bar Code Configuration Mode** by serial command **<BCCFG>**.
2. By reading a Data Matrix symbol with a special code word used by ISO/IEC 16022 to signify reader programming. This can be either in a regular read cycle or during a read rate test. Reading this symbol in the calibration routine will have no effect.<sup>1</sup>

Once **Bar Code Configuration Mode** has been entered, the Data Matrix symbols can be thought of as serial data. You can configure the reader by printing labels in Microscan's serial command format. Commands are processed as if the data were streamed in through the serial port. The reader will acknowledge the symbol with a beep, green flash, and echo the serial data to the host. If the command causes the reader to produce more serial output, such as serial verification or counter requests, the data will be routed to the host port.

**Bar Code Configuration Mode** can be exited by any reset **<A>**, **<Z>** or **<J>** command.

The command to exit **Bar Code Configuration Mode** can be included as part of the Data Matrix symbol. For example, try encoding **<K200,4><K220,1><J>** into a Data Matrix symbol. This configures the reader to enable **Serial Trigger Mode**, to program a new trigger to end the read cycle, and to exit **Bar Code Configuration Mode** with **<J>**.

**Note:** When the system is in barcode configuration mode, 2D symbols are allowed to pass through even if the reader only has a 1D license. The system outputs a **Config Code Received** message when the configuration symbol is decoded instead of showing the configuration symbol's actual symbol data. A **>** character is output to the Terminal to indicate that the system is ready for bar code configuration mode.

---

1. In normal reading modes, it is required to read a special Data Matrix symbol with a special codeword used by ISO/IEC 16022 to signify reader programming.

# **14 Output Format**

**Contents**

Output Format Serial Commands..... 14-2

Output Format Status..... 14-3

Format Assign ..... 14-4

Format Extract..... 14-5

Format Insert ..... 14-7

Output Filter Configuration ..... 14-9

Number of Filters..... 14-14

This section explains how to control the formatting and filtering of decoded symbol data for output.

# Output Format Serial Commands

Format Extract	<K740,output index,start location,length>
Format Insert	<K741,output index,length,hex string>
Format Assign	<K742,symbol number,status>
Format Status	<K743,output format status>
Output Filter Configuration	<K744,filter number,symbology,length,wildcard,placeholder, data, unused,database index>
Ordered Output Filter	<K745,number of filters>



# Output Format Status

**Definition:** This is a global enable/disable parameter. In order to use formatting you must set up the format using the insert and extract commands, and you must also assign a symbol to format using the **Format Assign** command.

**Serial Cmd:** <K743,output format status>

**Default:** Disabled

**Options:** 0 = Disabled 1 = Enabled

## Output Format Status Disabled

When **Output Format Status** is set to **Disabled**, output formatting is globally disabled.

## Output Format Status Enabled

When **Output Format Status** is set to **Enabled**, output formatting is enabled. However, **Format Assign**, **Format Insert**, and **Format Extract** must be properly set up as well.

## Output Format Status by ESP

On the **Output Format** tab, check the **Enable Output Format** box.

The screenshot shows the 'Output Format' tab in a software interface. A blue callout box with an arrow points to the 'Enable Output Format' checkbox, which is checked. Other visible elements include the 'Auto Sync with Reader' checkbox (checked), 'Set Number of Symbols' set to 1, and an 'Output Phrase' section containing fields for Preamble (CR), Symbol #1 (a QR code), and Postamble (CR LF). Below these are checkboxes for 'Enable', 'Parse', and 'Enable', all of which are checked. At the bottom are buttons for 'Parse Symbols...', 'Send and Save', 'Receive', and 'Quality Parameters...', along with a 'Show Parse Table' link.

# Format Assign

## Symbol Number

**Definition:** **Symbol Number** refers to the number of the symbol to which output formatting will apply. For example, if you wish to enable user-defined formatting to symbol # 2 in a multisymbol read cycle, you would send the command **<K742,2,1>**.

Note that the number of symbols may exceed the format capabilities.

**Serial Cmd:** **<K742,symbol number,status>**

**Options:** 1 to 10

1 = Formatted output status for symbol # 1.

2 = Formatted output status for symbol # 2.

...

10 = Formatted output status for symbol # 10.

## Status

**Definition:** **Status** refers to the user-defined formatting of a selected symbol position in the read cycle result. Note that there is also a global formatting “enable” command that must be set for the formatting to be applied.

**Serial Cmd:** **<K742,symbol number,status>**

**Default:** **Disabled**

**Options:** **0 = Disabled**                      1 = Enabled (Assign parameters to specified symbol.)

## Format Assign by ESP

The screenshot shows the 'Format Assign by ESP' interface. At the top, there is a checkbox labeled 'Enable Output Format' which is checked. Below it is a 'Set Number of Symbols' spin box set to '5'. A blue arrow points from a text box to this spin box. The text box says: 'Use the **Set Number of Symbols** spin box to determine the number of symbols to be included in the output phrase.'

Below the spin box is the 'Output Phrase' section. It contains a 'Preamble' field with 'CR', a 'Symbol Separator' field with a comma, and a 'Postamble' field with 'CR LF'. There are five symbol slots, each with a QR code and a 'Parse' checkbox. The first three symbols (Symbol #1, Symbol #2, Symbol #3) have their 'Parse' checkboxes checked. The last two (Symbol #4, Symbol #5) have theirs unchecked. A blue arrow points from a text box to the first three 'Parse' checkboxes. The text box says: 'Then check the **Parse** boxes beneath the symbols that you wish to format for user-defined output.'

Preamble:	Symbol #1	Symbol Separator:	Symbol #2	Symbol #3	Symbol #4	Symbol #5	Postamble:
CR		,					CR LF
<input checked="" type="checkbox"/> Enable	<input checked="" type="checkbox"/> Parse		<input checked="" type="checkbox"/> Parse	<input checked="" type="checkbox"/> Parse	<input type="checkbox"/> Parse	<input type="checkbox"/> Parse	<input checked="" type="checkbox"/> Enable

# Format Extract

## Output Index

**Definition:** **Output Index** refers to the database entry you wish to modify with this command. A formatted output is built by *extracting data from a symbol's original data output* and/or inserting user-defined characters.

It may be helpful to think of individual indexes as positions in the final formatted output you wish to build. Starting with index # 1, enter either an extract or insert command to begin building your desired output string. Then, with the next index number, enter either an extract or insert command to continue building the output string. Continue this process until you are finished building the string.

**Serial Cmd:** <**K740,output index,start location,length**>

**Options:** 1 to 100

## Start Location

**Definition:** Defines the location within the symbol data where the character extraction will begin. The first character extracted will also be the first character in the sequence displayed in user-defined output.

**Serial Cmd:** <**K740,output index,start location,length**>

**Default:** 0

**Options:** 1 to *n* (maximum number of characters in the symbol data).

## Length

**Definition:** Defines the length (in consecutive characters) that will be extracted and placed in user-defined output.

**Serial Cmd:** <**K740,output index,start location,length**>

**Default:** 0 (disabled; end of format cell array)

**Options:** 1 to *n* (maximum number of characters in the symbol data).

## Format Extract by ESP

The screenshot shows the 'Symbol Parse' dialog box with the 'Extract Range' set to '2 - 4'. A blue arrow points from the 'Extract Range' column header to the '2 - 4' value. Another blue arrow points from the 'Output: BCD' text in the 'Sample Symbol' section to the 'Output: BCD' text in the 'Symbol Parse' dialog box. A third blue arrow points from the 'Output: BCD' text in the 'Symbol Parse' dialog box to the 'Output: BCD' text in the 'Sample Symbol' section.

**Symbol Parse**

Step	Insert Data	Extract Range
Extract		2 - 4

Buttons: Add Insert Step, Add Extract Step, Remove Step, Clear All Parsing

Sample Symbol:  
Original: ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789  
Output: BCD

Buttons: OK, Cancel

The **Extract Range** function corresponds to the **Start Location** and **Length** parameters in the **Format Extract** serial command.

Multiple character sequences can be extracted and inserted using **Symbol Parse**. In this example, the selected extraction range is characters 2-4. The "Sample Symbol" example on the **Symbol Parse** dialog shows the selected character positions extracted and output as desired. Simultaneously, the data string from the selected symbol is displayed at the bottom left of the **Parse Table**, followed by the user-defined extracted output.

# Format Insert

## Output Index

**Definition:** **Output Index** refers to the database entry you wish to modify with this command. A formatted output is built by extracting data from a symbol's original data output and/or *inserting user-defined characters*.

It may be helpful to think of individual indexes as positions in the final formatted output you wish to build. Starting with index # 1, enter either an extract or insert command to begin building your desired output string. Then, with the next index number, enter either an extract or insert command to continue building the output string. Continue this process until you are finished building the string.

**Serial Cmd:** <**K741**,*output index,length,hex string*>

**Options:** 1 to 100

## Length

**Definition:** Specifies the length of the user-defined character string that will be inserted. This function is limited to 4 characters per output index, so multiple indexes must be entered in order to insert longer character sequences.

For example, if you wish to insert a 10 character sequence in user-defined output, you would need three commands with consecutive index numbers, where the first two character sequence lengths were 4 and the third was 2.

**Serial Cmd:** <**K741**,*output index,length,hex string*>

**Default:** 0 (disabled; end of format cell array)

**Options:** 1 to 4

## Hex String

**Definition:** Specifies a character string that represents ASCII characters to be inserted in the database entry. Two hex characters are required for every ASCII character to be inserted in the user-defined output string. These two characters comprise the hex (base 16) value of the ASCII character.

For example, if you wanted to enter the three-character sequence "Hi!" you would enter **3** for the length of the string, and a hex sequence of **486921** for the ASCII sequence to be inserted. (48 = H; 69 = i; 21 = !)

**Important:** Each pair of hex characters represents one ASCII character. Hex character pairs range from 00 to FF. Since you are limited to 4 ASCII characters per insertion per database entry, you are likewise limited to 8 hex characters per insertion per database entry.

**Serial Cmd:** <**K741**,*output index,length,hex string*>

**Default:** **NUL** (00 in hex)

**Options:** 00 to FF (As many as 4 bytes, or hex pairs.)

# Format Insert by ESP

Symbol Parse

Step	Insert Data	Extract Range
Insert		

Add Insert Step

Add Extract Step

Remove Step

Clear All Parsing

Sample Symbol:  
Original: ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789  
Output:  

OK

Cancel

SP SP

SOH	STX	ETX	EOT	ENQ
ACK	BEL	BS	HT	LF
VT	FF	CR	SO	SI
SI	DC1	DC2	DC3	DC4
NAK	SYN	ETB	CAN	EM
SUB	ESC	FS	GS	RS
US	SP			

Click 'Delete' to remove characters.

OK

Cancel

The **Insert** process is very similar to the **Extract** process, except that **Insert** allows the user to enter characters using the **Insert** calculator (shown above).

Parse Symbols...

Send and Save

Receive

Quality Parameters...

▼ Hide Parse Table

Build Sequence:

Test

Step	Insert Data	Extract Range
Insert	SP SP	
Extract		3 - 5

Notice that **Extract** and **Insert** share the same **Parse Table**.

Sample Symbol:  
Original: ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789  
Output: CDE

Symbol Parse

Step	Insert Data	Extract Range
Insert	SP SP	
Extract		3 - 5

Add Insert Step

Add Extract Step

Remove Step

Clear All Parsing

Sample Symbol:  
Original: ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789  
Output: CDE

OK

Cancel

14-8

MicroHAWK ID-20, ID-30, ID-40 User Manual

# Output Filter Configuration

*Definition:* Output filtering is a method of providing a set of good read qualifiers and also providing ordered output. There is a filter for up to the first 10 positions in a multisymbol output. The first filter corresponds to the first symbol output at the end of the read cycle. Each filter has settings for the following four parameters: **Symbology**, **Symbol Length**, **Data**, and **Configuration Database Number**.

*Serial Cmd:* <**K744**,filter number,symbology,length,wildcard,placeholder,data,unused,database index>

## Rules for Output Filter Configuration

### Output Filter Configuration Rule # 1

Each symbol that is decoded must match one of the filters before it can be saved to a read cycle record. There is an exception to this rule, however, when the number of symbols required for a read cycle exceeds the number of active filters. In such a case, unfiltered symbols can be placed into unfiltered output positions.

For example, if the number of symbols required is 6 but there are only 4 active filters, the last 2 positions can be filled by any (unfiltered) qualified symbol.

### Output Filter Configuration Rule # 2

The same filter setup can be used multiple times.

For example, filters 1, 2, and 3 can be set up to filter Data Matrix symbols, and the output will occur in the order the symbols are decoded.

### Output Filter Configuration Rule # 3

All qualified symbols will be sorted and output in the matching filter position. If a symbol matches filter 3, it will be output as the third symbol. If a filter does not have a matching qualified symbol, a No Read message will be output in place of the symbol (assuming the No Read message is enabled).

For example, if there is not a symbol that meets filter 3's requirements, then a No Read message will be output in the third output position.

## Filter Number

**Definition:** This is the filter index number that represents the position of the symbol in the data output at the end of the read cycle. This index number should be entered along with the following filter settings for the predetermined symbol position.

**Serial Cmd:** <**K744**,**filter number**,**symbology**,**length**,**wildcard**,**placeholder**,**data**,  
**unused**,**database index**>

**Options:** 1 to 10

## Symbology

**Definition:** Specifies the symbology allowed to occupy this location in multisymbol output.  
**Note:** To filter or order a symbol, the symbol must meet all the requirements of the selected filter index.

**Serial Cmd:** <**K744**,**filter number**,**symbology**,**length**,**wildcard**,**placeholder**,**data**,  
**unused**,**database index**>

**Default:** **0** (any symbology)

**Options:**

- 0 = Any type
- 1 = Interleaved 2 of 5
- 2 = Code 39
- 3 = Code 128
- 4 = Codabar
- 5 = UPC
- 6 = PDF417
- 7 = EAN 128
- 8 = Code 93
- 9 = Pharmacode
- 10 = GS1 DataBar
- 11 = MicroPDF417
- 12 = Composite
- 13 = BC412
- 14 = Data Matrix
- 15 = QR Code
- 16 = Micro QR Code
- 17 = Aztec
- 18 = Postal Symbologies



## Length

**Definition:** Specifies the length of the decoded symbol allowed to occupy this location in multisymbol output.

**Note:** To filter or order a symbol, the symbol must meet all requirements of the selected filter index.

**Serial Cmd:** <**K744**,filter number,symbology,**length**,wildcard,placeholder,data,unused,database index>

**Default:** 0

**Options:** 0 to 64

## Wildcard

**Definition:** This is the character to be used in the data output field when performing a data filter comparison. The wildcard character represents the end of matching, and allows for variable lengths of symbol output.

**Serial Cmd:** <**K744**,filter number,symbology,length,**wildcard**,placeholder,data,unused,database index>

**Default:** “ \* ” = **2A** (hex)

**Options:** Any ASCII input in the form of a pair of hex characters.

Example:

2A = \*

00 = disabled

## Placeholder

**Definition:** The placeholder character requires a character to be present, but does not compare the data value.

**Serial Cmd:** <**K744**,filter number,symbology,length,wildcard,**placeholder**,data,unused,database index>

**Default:** “ ? ” = **3F** (hex)

**Options:** Any ASCII input in the form of a pair of hex characters.

Example:

3F = ?

00 = disabled

## Data

**Definition:** This is the data string to be used when comparing symbol data for output filtering and ordering. This data string may also contain wildcard and placeholder characters to facilitate matching. Remember that in order to filter or order symbol data, it must meet all the requirements of the selected filter index.

**Examples:**

- Filter data = "123\*". This will match data strings of "123", "123456", and "123ABC", but not "12".
- Filter data = "123\*AB?C". This will be interpreted as "123\*".
- Filter data = "123?". This will match "1234" and "123A", but not "123", "12345", or "1234C".
- Filter data = "123?A". This will match "1234A" and "123BA", but not "123", "1234C", or "1234ABCD".
- Filter data = "123?A?". This will match "1234AB" and "123BAT", but not "1234A" or "123BATS".
- Filter data = "12??\*". This will match "1234", "123456", and "123ABC", but not "12" or "123".
- Filter data = "123?A\*". This will match "1234A", "123BA", and "123BATS", but not "1234" or "1234C".

**Serial Cmd:** <**K744**,filter number,symbology,length,wildcard,placeholder,data,unused,database index>

**Default:** 00 (NUL)

**Options:** Any ASCII input in the form of a pair of hex characters.  
Maximum length: 63 bytes defined.

**Examples:**

41422A = AB\*

Data [0] = NUL represents string matching disabled.

## Database Index

**Definition:** The index of the database entry that decodes a given symbol must equal this setting for filtering to occur. A setting of 0 allows any database index for this filter entry.

**Serial Cmd:** <**K744**,filter number,symbology,length,wildcard,placeholder,data,unused,database index>

**Default:** 0 (any index)

**Options:** 0 to 10

## Ordered Output Filter Configuration by ESP

Double-clicking on a row in the Ordered Output table brings up the **Ordered Output Filter Settings** dialog.

Use these settings to determine **Symbology Type**, **Length** of the symbol, a user-defined **Matching String**, **ASCII Lookup**, **Wildcard Character**, **Placeholder Character**, and **Database Number**. Click **Apply** to save the settings to the corresponding filter.

The screenshot shows a dialog box titled "Ordered Output Filter Settings" with a close button (X) in the top right corner. The dialog is divided into several sections:

- Filter # 1 Criteria**:
  - Symbology Type**: A dropdown menu set to "Any Type".
  - Length**: A text input field containing the number "0".
- Enter Match String**:
  - Matching String**: A text input field containing an asterisk (\*).
  - ASCII Lookup**: A dropdown menu set to "CR".
  - Add to Match String**: A button.
- Match String Definables**:
  - Wildcard Character**: A dropdown menu set to "\*".
  - Placeholder Character**: A dropdown menu set to "?".
- Database Number**: A dropdown menu set to "Any".

At the bottom of the dialog are two buttons: "Apply" (highlighted in blue) and "Cancel".

## Number of Filters

**Definition:** **Number of Filters** refers to the number of active output filters. **0** disables all output filters. Any non-zero numeral will enable filtering to be performed using the filter indexes covered by this value.

For example, if the number of filters is **1**, then only filter index # 1 will be applied. If the number of filters is **2**, then only filter index # 1 and filter index # 2 will be applied, etc.

**Serial Cmd:** <**K745,number of filters**>

**Default:** **0**

**Options:** 0 to 10

# Appendices

**Contents**

Appendix A General Specifications ..... A-2

Appendix B Electrical Specifications ..... A-6

Appendix C Serial Commands ..... A-8

Appendix D Communications Protocol..... A-14

Appendix E ASCII Table..... A-15

Appendix F Glossary of Terms..... A-16

## Appendix A — General Specifications

### MicroHAWK ID-20

#### Mechanical

**Height:** 0.95" (24.1 mm)

**Width:** 1.32" (33.5 mm)

**Depth:** 1.53" (38.8 mm)

**Weight:** 0.92 oz. (26 g)

#### Environmental

**Enclosure:** IP40

**Humidity:** 5% to 95% (non-condensing)

**Operating Temperature:** 0° to 40°C (32° to 104°F)

**Storage Temperature:** -50° to 75° C (-58 to 167°F)

#### Illumination

**LEDs:** 4 Red (625nm), 4 White, 2 Blue (Target Pattern), 2 Green (Green Flash on Good Read)

#### Light Collection Options

Progressive scan, square pixel; Software-adjustable exposure; electronic shutter; fixed focus, factory-configurable, 50 mm to 300 mm; Standard Density or High Density

**WVGA:** 752 by 480 pixels (Global Shutter)

**SXGA:** 1280 by 960 pixels (Global Shutter)

**QSXGA:** 2592 by 1944 pixels (Rolling Shutter)

#### Symbologies

**2D:** Data Matrix (ECC 0-200), QR Code, Micro QR Code, Aztec, DotCode

**Stacked:** PDF417, MicroPDF417, GS1 DataBar (Composite and Stacked)

**Linear:** Code 39, Code 128, BC412, Interleaved 2 of 5, UPC/EAN, Codabar, Code 93, Pharmacode, PLANET, POSTNET, Japanese Post, Australian Post, Royal Mail, Intelligent Mail, KIX

#### Read Parameters

**Pitch:** ±30° **Skew:** ±30° **Tilt:** 360°

**Frames Per Second:** 60 (WVGA); 42 (SXGA); 5 (QSXGA)

**Decoder:** Standard (High-Contrast 1D); Plus (High-Contrast 1D/2D); X-Mode (All 1D/2D and DPM)

**Speed:** Standard Speed or High Speed

#### Connector

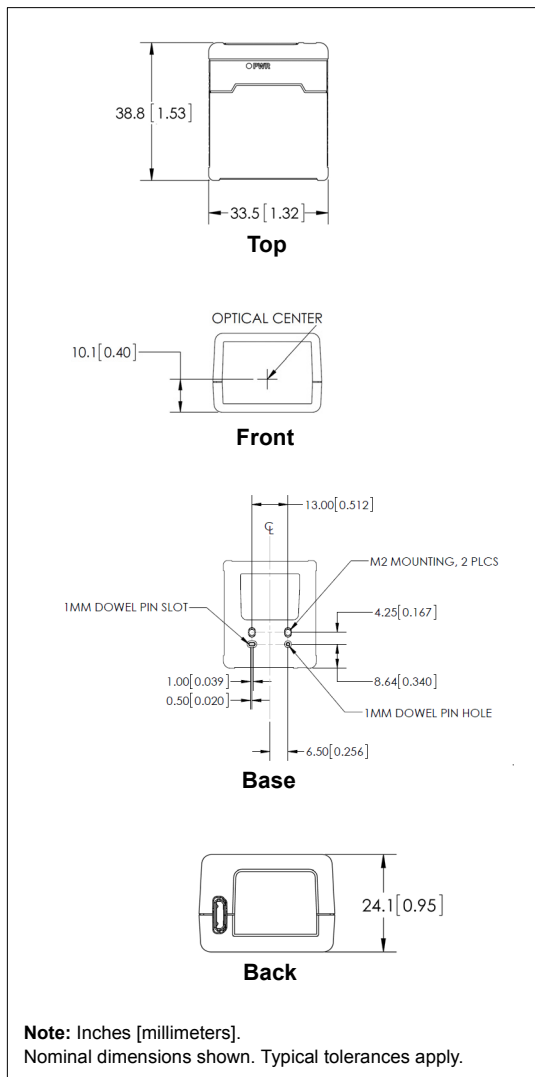
Micro-USB Type B

#### Status Indicators

Power-On LED, Green LED flash on Good Read

#### Communication Interface

USB 2.0



**MicroHAWK ID-20 Dimensions**

# MicroHAWK ID-30

## Mechanical

**Height:** 1.0" (25.4 mm)

**Width:** 1.75" (44.5 mm)

**Depth:** 1.5" (38.1 mm)

**Weight with Cable:** 4.02 oz. (114 g)

**Weight without Cable:** 1.62 oz. (46 g)

## Environmental

**Enclosure:** IP54

**Humidity:** 5% to 95% (non-condensing)

**Operating Temperature:** 0° to 40°C (32° to 104°F)

**Storage Temperature:** -50° to 75° C (-58 to 167°F)

## Illumination

**Inner LEDs:** 4 Red (625nm), 4 White, 2 Blue (Target Pattern), 2 Green (Green Flash on Good Read)

**Outer LEDs:** Optional Red (617nm) or White LED Expansion – 8 High-Intensity LEDs

## Light Collection Options

Progressive scan, square pixel; Software-adjustable exposure; electronic shutter; fixed focus, factory-configurable, 50 mm to 300 mm; Standard Density or High Density

**WVGA:** 752 by 480 pixels (Global Shutter)

**SXGA:** 1280 by 960 pixels (Global Shutter)

**QSXGA:** 2592 by 1944 pixels (Rolling Shutter)

## Symbolologies

**2D:** Data Matrix (ECC 0-200), QR Code, Micro QR Code, Aztec, DotCode

**Stacked:** PDF417, MicroPDF417, GS1 DataBar (Composite and Stacked)

**Linear:** Code 39, Code 128, BC412, Interleaved 2 of 5, UPC/EAN, Codabar, Code 93, Pharmacode, PLANET, POSTNET, Japanese Post, Australian Post, Royal Mail, Intelligent Mail, KIX

## Read Parameters

**Pitch:** ±30° **Skew:** ±30° **Tilt:** 360°

**Frames Per Second:** 60 (WVGA); 42 (SXGA); 5 (QSXGA)

**Decoder:** Standard (High-Contrast 1D); Plus (High-Contrast 1D/2D); X-Mode (All 1D/2D and DPM)

**Speed:** Standard Speed or High Speed

## Connector

High-Density 15-Pin Dsub

## Status Indicators

**LEDs:** Power, Link (Ethernet), Trigger, Mode, Pass, Fail, Green LED flash on Good Read

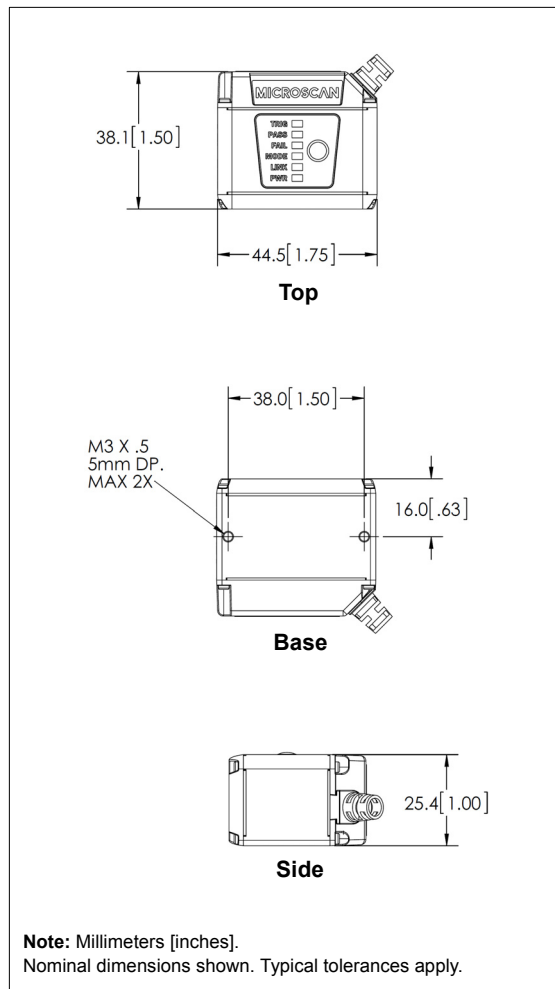
## Communication Interface

**Standard Interfaces:** USB 2.0 / Serial / I/O / Power

## Discrete I/O

**Inputs:** Trigger Input: 5-28V rated (0.16mA @ 5VDC); New Master: 5 to 28V rated (0.16mA @ 5VDC); Default: 3.3V rated (0mA @ 3.3V)

**Outputs (1, 2, 3):** 5V TTL-compatible, can sink 10mA and source 10mA



**MicroHAWK ID-30 Dimensions**

## General Specifications

### MicroHAWK ID-40

#### Mechanical

**Height:** 1.0" (25.4 mm)

**Width:** 1.75" (44.5 mm)

**Depth:** 1.75" (44.5 mm)

**Weight:** 2.40 oz. (68 g)

#### Environmental

**Enclosure:** IP65/67

**Humidity:** 5% to 95% (non-condensing)

**Operating Temperature:** 0° to 45°C (32° to 113°F)

**Storage Temperature:** -50° to 75° C (-58 to 167°F)

#### Illumination

**Inner LEDs:** 4 Red (625nm), 4 White, 2 Blue (Target Pattern), 2 Green (Green Flash on Good Read)

**Outer LEDs:** Optional Red (617nm) or White LED Expansion – 8 High-Intensity LEDs

#### Light Collection Options

Progressive scan, square pixel; Software-adjustable exposure; electronic shutter; fixed focus, factory-configurable, 50 mm to 300 mm; Standard Density or High Density

**WVGA:** 752 by 480 pixels (Global Shutter)

**SXGA:** 1280 by 960 pixels (Global Shutter)

**QSXGA:** 2592 by 1944 pixels (Rolling Shutter)

#### Symbologies

**2D:** Data Matrix (ECC 0-200), QR Code, Micro QR Code, Aztec, DotCode

**Stacked:** PDF417, MicroPDF417, GS1 DataBar (Composite and Stacked)

**Linear:** Code 39, Code 128, BC412, Interleaved 2 of 5, UPC/EAN, Codabar, Code 93, Pharmacode, PLANET, POSTNET, Japanese Post, Australian Post, Royal Mail, Intelligent Mail, KIX

#### Read Parameters

**Pitch:** ±30° **Skew:** ±30° **Tilt:** 360°

**Frames Per Second:** 60 (WVGA); 42 (SXGA); 5 (QSXGA)

**Decoder:** Standard (High-Contrast 1D); Plus (High-Contrast 1D/2D); X-Mode (All 1D/2D and DPM)

**Speed:** Standard Speed or High Speed

#### Connector

Dual M12 12-Pin (A = Power) and M12 8-Pin (B = Ethernet)

#### Status Indicators

**LEDs:** Power, Link (Ethernet), Trigger, Mode, Pass, Fail, Green LED flash on Good Read

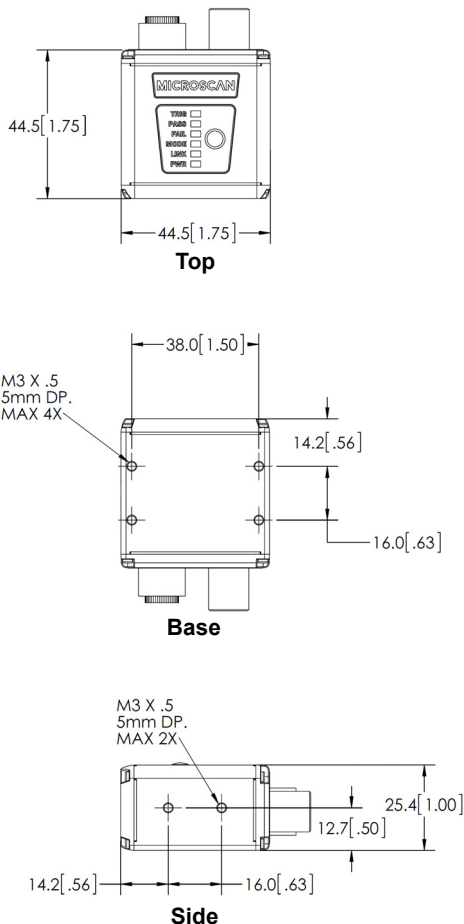
#### Communication Interface

**Standard Interfaces:** Serial / Ethernet 10/100 / I/O / Power

#### Discrete I/O

**Inputs:** Trigger Input; New Master: Bi-directional, optoisolated, 4.5-28V rated (10mA @ 28VDC)

**Outputs (1, 2, 3):** Bi-directional, optoisolated, 1-28V rated, ( $I_{CE} < 100$  mA at 24VDC, current limited by user)



**Note:** Inches [millimeters].  
Nominal dimensions shown. Typical tolerances apply.

### MicroHAWK ID-40 Dimensions



### **Safety Certifications**

PENDING: FCC, UL/cUL, CE, CB Class A, RoHS/WEEE



©2015 Microscan Systems, Inc.

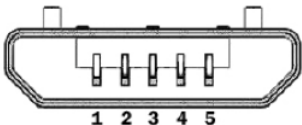
All rights reserved. Specifications subject to change.

Product specifications are given for typical performance at 25°C (77°F) using grade A labels. Performance characteristics may vary at high temperatures or other environmental extremes. Warranty—One year limited warranty on parts and labor. Extended warranty available.

# Appendix B — Electrical Specifications

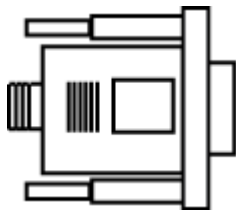
- MicroHAWK ID-20: 5VDC ± 5%; 350mA at 5VDC (typ.)
- MicroHAWK ID-30: 5V ± 5%; 600mA at 5VDC (typ.)
- MicroHAWK ID-40: 4.75V – 30V; 150mA at 24VDC (typ.)

## MicroHAWK ID-20 Micro-USB Type B Socket



Pin	Function
1	Vbus (5V)
2	D-
3	D+
4	N/C
5	Ground

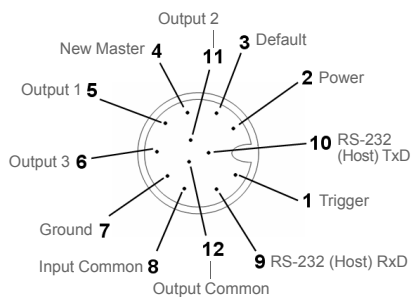
## MicroHAWK ID-30 High-Density 15-Pin Dsub Socket



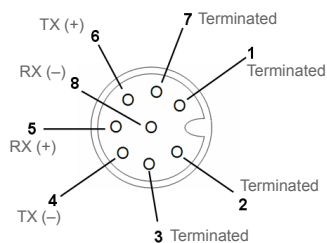
**Note:** Accessory cable required between 15-pin socket and host USB port.

Pin	Function
1	+5VDC
2	TX232
3	RX232
4	GND
5	D+
6	N/C
7	Output 1+
8	Default+
9	Trigger+
10	D-
11	Output 3+
12	New Master+
13	Chassis
14	Output 2+
15	Vbus

## MicroHAWK ID-40 M12 Connectors



### ***M12 12-pin Plug***



### ***M12 8-pin Socket (Ethernet)***

## Appendix C — Serial Commands

**Note:** For a list of utility commands such as read rate requests, device control options, and reader status requests, see **Serial Utility Commands**.

<b>Communications</b>	
Host Port Connections	<K100,baud rate,parity,stop bits,data bits>
Ethernet	<K126,status,IP address,subnet,gateway,IP address mode>
Ethernet TCP Ports	<K127,TCP Port 1,TCP Port 2>
Search and Configure Mode	<K128,status,timed window>
EtherNet/IP	<K129,status>
Host Protocol	<K140,protocol,address>
Preamble	<K141,status,preamble characters>
Postamble	<K142,status,postamble characters>
Response Timeout	<K143,response timeout>
LRC Status	<K145,status>
ACK/NAK Options	<K147,RES,REQ,STX,ETX,ACK,NAK>
Polling Mode Options	<K148,RES,REQ,STX,ETX,ACK,NAK>
<b>Calibration</b>	
Calibration	<K529,gain,exposure,unused,symbol type,WOL framing,WOL margin,linescan height,processing>
<b>Read Cycle</b>	
Trigger Mode/Filter Duration	<K200,trigger mode,leading edge trigger filter,trailing edge trigger filter>
Serial Trigger Character	<K201,serial trigger character>
External Trigger State	<K202,external trigger state>
End of Read Cycle	<K220,end of read cycle mode,read cycle timeout>
Decodes Before Output	<K221,decodes before output>
Multisymbol	<K222,number of symbols,multisymbol separator>
Start Trigger Character	<K229,start character>
Stop Trigger Character	<K230,stop character>
Capture Mode	<K241,capture mode,number of rapid captures,rapid capture mode,number of continuous captures,images per read cycle limit,read cycle history limit>
Capture Time	<K242,time before first capture,time between captures 1 and 2,time between captures 2 and 3,time between captures 3 and 4,time between captures 4 and 5,time between captures 5 and 6,time between captures 6 and 7,time between captures 7 and 8>
Image Storage	<K244,image storage type,image store mode>
Image Processing Timeout	<K245,image processing timeout>
<b>Symbologies</b>	
Composite	<K453,symbology status,separator status,separator character>
Aztec Code	<K458,status>
Micro QR Code	<K459,status>

Postal Symbolologies	<K460,postal symbology type,POSTNET status,PLANET status,USPS4CB status>
Code 39	<K470,status,check character status,check character output status,large intercharacter gap,fixed symbol length status,fixed symbol length,full ASCII set>
Codabar	<K471,status,start/stop match,start/stop output,large intercharacter gap,fixed symbol length status,symbol length,check character type,check character output>
Interleaved 2 of 5	<K472,status,check character status,check character output status,symbol length #1, symbol length #2, guard bar status,range mode status>
UPC/EAN	<K473,UPC status,EAN status,supplemental status,separator status,separator character,supplemental type,format UPC-E as UPC-A>
Code 128/EAN 128	<K474,status,fixed symbol length status,fixed symbol length,EAN 128 status,output format,application record separator status,application record separator character,application record brackets,application record padding>
Code 93	<K475,status,fixed symbol length status,symbol length>
PDF417	<K476,status,unused,fixed symbol length status,symbol length>
Pharmacode	<K477,status,fixed bar count status,fixed bar count,minimum bar count,bar width mode,direction,fixed threshold value>
Data Matrix	<K479,ECC 200 status, ECC 000 status,ECC 050 status, ECC 080 status,ECC 100 status,ECC 140 status,ECC 120 status,ECC 130 status>
QR Code	<K480,status>
BC412	<K481,status,check character output,fixed symbol length status, fixed symbol length>
DataBar-14	<K482,status>
DataBar Limited	<K483,status>
DataBar Expanded	<K484,status,fixed symbol length status,fixed symbol length>
MicroPDF417	<K485,status,[unused],fixed symbol length status,fixed symbol length>
DotCode	<K497,status,rotation mode>

### **I/O Parameters**

Power On/Reset Counts	<K406,power-on,resets,power-on saves,customer default saves>
Time Since Reset (Read-Only)	<K407,hours,minutes>
Service Message	<K409,status,service message,threshold,resolution>
User-Defined Name	<K412,user-defined name>
Serial Verification	<K701,serial command echo status,serial command beep status, control/hex output>
Beeper	<K702,status>
Quality Output Separator	<K704,quality output separator,decodes per trigger status>
Symbol Data Output	<K705,status,when to output,symbology identifier status>
Read Duration Output	<K706,status,separator>
No Read Message	<K714,status,message>
Object Output Information	<K734,output frame number status,output coordinates status>
LED Configuration	<K737,ISO/IEC 16022 grade>

## Serial Commands

Green Flash and Target LEDs	<K750,green flash LED status,target pattern LED status,green flash duration>
Database Identifier Output	<K759,status,separator character>
Trend Analysis Output 1	<K780,trend analysis mode,number of triggers,number to output on,decodes per trigger>
Trend Analysis Output 2	<K781,trend analysis mode,number of triggers,number to output on,decodes per trigger>
Trend Analysis Output 3	<K782,trend analysis mode,number of triggers,number to output on,decodes per trigger>
Diagnostic Output 1	<K790,service unit message status>
Diagnostic Output 2	<K791,service unit message status>
Diagnostic Output 3	<K792,service unit message status>
ISO/IEC 16022 Symbol Quality Output 1	<K800,output on symbol contrast,symbol contrast threshold,output on print growth,print growth threshold,output on axial non-uniformity,axial non-uniformity threshold,output on Unused ECC,Unused ECC threshold>
ISO/IEC 16022 Symbol Quality Output 2	<K801,output on symbol contrast,symbol contrast threshold,output on print growth,print growth threshold,output on axial non-uniformity,axial non-uniformity threshold,output on Unused ECC,Unused ECC threshold>
ISO/IEC 16022 Symbol Quality Output 3	<K802,output on symbol contrast,symbol contrast threshold,output on print growth,print growth threshold,output on axial non-uniformity,axial non-uniformity threshold,output on Unused ECC,Unused ECC threshold>
Output 1 Parameters	<K810,output on,output state,pulse width,output mode>
Output 2 Parameters	<K811,output on,output state,pulse width,output mode>
Output 3 Parameters	<K812,output on,output state,pulse width,output mode>

### Symbol Quality

Symbol Quality Separator/Output Mode	<K708,symbol quality separator,Data Matrix output mode>
ISO/IEC 16022 Symbol Quality Output	<K709,symbol contrast,print growth,axial non-uniformity,unused ECC>
Microscan Symbol Quality Output	<K710,percent cell damage,total read time,capture time,locate time,decode time,pixels per element,ECC level,matrix size,quiet zone,symbol angle>

### Matchcode

Matchcode	<K223,matchcode type,sequential matching,match start position,match length,wildcard character,sequence on No Read,sequence on mismatch>
Number of Master Symbols	<K224,number of master symbols>
New Master Pin	<K225,status>
Sequence Step	<K228,sequence step value>
Master Symbol Data	<K231,index,master symbol data>
Match Replace	<K735,status,match replacement string>
Mismatch Replace	<K736,status,mismatch replacement string>

### Camera and IP Setup

Window of Interest	<K516,row pointer,column pointer,row depth,column width>
Focal Distance	<K525,focal distance,distance units,lens model>

Illumination Brightness	<K536,brightness,light source>
Camera Settings	<K541,exposure,gain>
Pixel Binning	<K542,status>
Morphological Pre-Processing	<K550,status>
Morphological Operation	<K551,morphological operation,operator size>
<b>Configuration Database</b>	
Number of Active Indexes	<K252,number of active indexes,sort database>
Configuration Database Status	<K255,index,exposure,gain,focal distance,pixel binning,row pointer,column pointer,row depth,column width,symbology,morphology operator,morphology size>
Save Current Settings to Database	<K255+,index>
Load Current Settings from Database	<K255-,index>
Request Selected Index Settings	<K255?,index>
Request All Database Settings	<K255?>
Database Mode	<K256,switch mode,frame count/time,image process looping,image dimensions>
<b>Output Format</b>	
Format Extract	<K740,output index,start location,length>
Format Insert	<K741,output index,length,hex string>
Format Assign	<K742,symbol number,status>
Format Status	<K743,output format status>
Output Filter Configuration	<K744,filter number,symbology,length,wildcard,placeholder,data,unused,database index>
Ordered Output Filter	<K745,number of filters>

## Serial Command Format

Serial commands are of two types: utility and configuration.

### Rules that apply to both utility and configuration commands

- A less than < and greater than > character enclose the commands.
- Commands and data are “case sensitive.” That is, characters must be entered as upper or lower case, as specified.

### Serial Utility Commands

These are sent during operations and are not followed by <A> or <Z>.

### Serial Configuration “K” Commands

These begin with a single “K” character followed by a 3-digit numeric character, data fields, and an initializing command, as follows:

**<Knumeric parameter,data,data,...etc.><initializing command>**

An initializing command <A> or <Z> may follow the command. A <Z> initializes the reader’s memory and saves for power-on; an <A> initializes the reader’s memory but does not save for power-on.

For example, to enable **UPC** and save the change for power-on, send **<K473,1><Z>**.

To change **Baud Rate** and reset without saving changes for power-on, send **<K100,3><A>**.

### Serial Configuration Command Conventions

- All data fields (except the last) must be followed by a comma (*without a space*).
- NUL cannot be used. The characters <, >, and , can be used, but only if entered as hex pairs (see **ASCII Character Entry Modifier**).
- All fields preceding a modified field must be included.
- If there is no change in preceding fields, then commas alone can be entered in these fields. For example, if only the last field in the following command is changing, **<K100,4,1,0,0>** can be entered as **<K100,,,0>**.
- All fields *following* a modified field can be omitted. For example, to change **Baud Rate** only, send **<K100,3>**.



## Concatenating Configuration Commands

Commands can be concatenated (added together) in a single string or data block. For example, **<K145,1><K220,1><K451,0><A>** enables **LRC**, sets **End of Read Cycle** mode to **New Trigger**, sets **Background Color** to **White**, and resets the data buffers (without saving the changes for power-on).

## Serial Command Status Request

To ensure that any command was received and accepted, you can send the **Show Reader Status** command: **<?>**.

The status of a specific serial command can be requested by entering the command followed by a question mark. For example, send **<K142?>** to request the status of **Postamble**.

## Entering Control Characters in Serial Commands

To enter control characters within a serial command, hold down the **Ctrl** key while typing the desired character.

*Example:* To enter a carriage return and line feed (**^M^J**), enter **<K141,1,CNTL-m CNTL-j>**.

## **Appendix D — Communications Protocol**

### **Communications Protocol Command Table**

<b>Protocol Command (Mnemonic displayed on menu)</b>	<b>Control Characters (Entered in menu or serial command)</b>	<b>Hex Value</b>	<b>Effect of Command</b>
RES	^D	04	Reset
REQ	^E	05	Request
EOT	^D	04	Reset
STX	^B	02	Start of Text
ETX	^C	03	End of Text
ACK	^F	06	Acknowledge
NAK	^U	15	Negative Acknowledge
XON	^Q	11	Begin Transmission
XOFF	^S	13	Stop Transmission

# Appendix E — ASCII Table

Dec	Hex	Mne	Ctrl	Dec	Hex	Ch	Dec	Hex	Ch	Dec	Hex	Ch
00	00	NUL	^@	32	20	SP	64	40	@	96	60	`
01	01	SOH	^A	33	21	!	65	41	A	97	61	a
02	02	STX	^B	34	22	"	66	42	B	98	62	b
03	03	ETX	^C	35	23	#	67	43	C	99	63	c
04	04	EOT	^D	36	24	\$	68	44	D	100	64	d
05	05	ENQ	^E	37	25	%	69	45	E	101	65	e
06	06	ACK	^F	38	26	&	70	46	F	102	66	f
07	07	BEL	^G	39	27	'	71	47	G	103	67	g
08	08	BS	^H	40	28	(	72	48	H	104	68	h
09	09	HT	^I	41	29	)	73	49	I	105	69	i
10	0A	LF	^J	42	2A	*	74	4A	J	106	6A	j
11	0B	VT	^K	43	2B	+	75	4B	K	107	6B	k
12	0C	FF	^L	44	2C	,	76	4C	L	108	6C	l
13	0D	CR	^M	45	2D	-	77	4D	M	109	6D	m
14	0E	SO	^N	46	2E	.	78	4E	N	110	6E	n
15	0F	SI	^O	47	2F	/	79	4F	O	111	6F	o
16	10	DLE	^P	48	30	0	80	50	P	112	70	p
17	11	DC1	^Q	49	31	1	81	51	Q	113	71	q
18	12	DC2	^R	50	32	2	82	52	R	114	72	r
19	13	DC3	^S	51	33	3	83	53	S	115	73	s
20	14	DC4	^T	52	34	4	84	54	T	116	74	t
21	15	NAK	^U	53	35	5	85	55	U	117	75	u
22	16	SYN	^V	54	36	6	86	56	V	118	76	v
23	17	ETB	^W	55	37	7	87	57	W	119	77	w
24	18	CAN	^X	56	38	8	88	58	X	120	78	x
25	19	EM	^Y	57	39	9	89	59	Y	121	79	y
26	1A	SUB	^Z	58	3A	:	90	5A	Z	122	7A	z
27	1B	ESC	^[	59	3B	;	91	5B	[	123	7B	{
28	1C	FS	^\	60	3C	<	92	5C	\	124	7C	
29	1D	GS	^]	61	3D	=	93	5D	]	125	7D	}
30	1E	RS	^^	62	3E	>	94	5E	^	126	7E	~
31	1F	US	^_	63	3F	?	95	5F	_	127	7F	D

## **Appendix F — Glossary of Terms**

**Aberration** — The failure of an optical lens to produce an exact point-to-point correspondence between the object and its resulting image. Various types are chromatic, spherical, coma, astigmatism and distortion.

**Absorption** — The loss of light of certain wavelengths as it passes through a material and is converted to heat or other forms of energy. (–)

**Active Illumination** — Lighting an area with a light source coordinated with the acquisition of an image. Strobed flash tubes and pulsed lasers are examples.

**Ambient Light** — Light which is present in the environment of the imaging front end of a vision system and generated from outside sources. This light, unless used for actual illumination, will be treated as background noise by the vision system.

**Analog** — A smooth, continuous voltage or current signal or function whose magnitude (value) is the information.

**Analog-to-Digital Converter** — A device that converts an analog voltage or current signal to a discrete series of digitally encoded numbers (signal) for computer processing.

**Application-Specific Integrated Circuit (ASIC)** — An integrated circuit that is customized for a particular kind of use, rather than general use. All vision system elements including firmware can be integrated into one ASIC.

**Architecture** — The hardware organization of a vision system designed for high speed image analysis.

**Aspect Ratio** — The ratio between the height and width of a sensor or display. Found by dividing the vertical number of pixels (height) by the horizontal number of pixels (width) leaving it in fractional format.

**Automatic Gain Control (AGC)** — Adjustment to signal strength that seeks to maintain a constant level regardless of the distance between a reader and symbol.

**Auxiliary Port** — RS-232 connection to an auxiliary terminal or device for remote viewing.

**Baud Rate** — The number of discrete signal events per second; bits per second.

**Capture** — The act of acquiring and storing video images in a reader or host computer. Also, the image captured.

**Charge-Coupled Device (CCD)** — A semiconductor device with an array of light-sensitive elements that converts light images into electrical signals.

**Check Character** — A Modulus 43 or Modulus 10 character that is added to encoded symbol data for additional data integrity.

**Complementary Metal Oxide Semiconductor (CMOS)** — Like CCDs, CMOS readers include an array of photo-sensitive diodes, one diode within each pixel. Unlike CCDs, however, each pixel in a CMOS reader has its own individual amplifier integrated inside.

**Connector** — A plug or socket on a device or cable providing in/out connectivity for various circuits and pins.

**Counter** — Memory space allocated to keep track of reader events.

**Dark Field Illumination** — Lighting of objects, surfaces, or particles at very shallow or low angles, so that light does not directly enter a reader's optical hardware.

**Decode** — A good read. The successful interpretation and output of the information encoded in a symbol.

**Default** — Restores ROM or flash settings, initializes serial commands and resets all counters.

**Delimited** — A delimited command or field is bracketed by predefined characters.

**Depth-of-Field** — The in-focus range of an imaging system. Measured from the distance behind an object to the distance in front of the object with all objects appearing in focus.

**Diffused Lighting** — Scattered soft lighting from a wide variety of angles used to eliminate shadows and specular glints from profiled, highly reflective surfaces.

**Digital Imaging** — Conversion of an image into pixels by means of an Analog-to-Digital Converter where the level of each pixel can be stored digitally.

**Digital Signal Processor (DSP)** — A VLSI chip designed for ultra-high-speed arithmetic processing. Often imbedded in a vision engine.

**Digital-to-Analog Converter** — A VLSI circuit used to convert digitally processed images to analog for display on a monitor.

**Direct Memory Access** — A capability provided by some computer bus architectures that allows data to be sent directly to memory from an attached device.

**Discrete I/O** — Inputs and outputs characterized by discrete signal transitions from one voltage level to another so that digital switching can occur.

**Dynamic Range** — The difference between the minimum and maximum thresholds of discernible images; the amount of usable signal.

**Edge Enhancement** — Image processing method to strengthen high-spatial frequencies in the image.

**Embedded Memory** — Onboard memory device such as EPROM or flash.

**End of Read Cycle** — The time or condition at which the reader stops expecting symbol information to decode.

**Erasable Programmable Read-Only Memory (EPROM)** — A memory chip that retains data when its power supply is turned off. Non-volatile memory.

**External Edge** — Allows a read cycle to be initiated by a trigger signal from an object detector when it detects the appearance of an object (rising edge). The read cycle ends with a good read, a timeout, or a new trigger.

**External Level** — Allows a read cycle to be initiated by a trigger signal from an object detector. The read cycle ends when the object moves out of the detector's range.

**Falling Edge** — A change of state (to inactive) associated with a level trigger.

**Field-Programmable Gate Array (FPGA)** — A semiconductor containing programmable interconnects and logic components.

**Firmware** — Software hard-coded in non-volatile memory, and closely tied to specific pieces of hardware.

---

## Glossary of Terms

**Fixed Symbol Length** — Increases data integrity by ensuring that only one symbol length will be accepted.

**Focal Distance** — In camera-based vision, the distance from the front of the camera to the object being viewed. (In optics, the distance from the lens to the focal plane.)

**Focal Plane** — Usually found at the image sensor, it is a plane perpendicular to the lens axis at the point of focus (—).

**Focus** — Any given point in an image at which light converges; the focal point.

**Frame** — The total area captured by an image sensor.

**Frame Grabber** — A device that interfaces with a camera and, on command, samples the video, converts the sample to a digital value and stores that in a computer's memory.

**Full Duplex** — A communications system in which signals can travel simultaneously between devices.

**Gain** — The amount of energy applied to pixel gray scale values prior to output; optimal signal strength.

**Good Read** — A decode. The successful scanning and decoding of the information encoded in a bar code symbol.

**Gradient** — The rate of change of pixel intensity (first derivative).

**Grayscale** — Variations of values from white, through shades of gray, to black in a digitized image with black assigned the value of zero and white the value of one.

**Half Duplex** — A communications system in which signals can travel between devices in both directions, but not simultaneously.

**Histogram** — A graphical representation of the frequency of occurrence of each intensity or range of intensities (gray levels) of pixels in an image. The height represents the number of observations occurring in each interval.

**Host** — A computer, PLC, or other device that is used to execute commands and process data and discrete signals.

**Image** — Projection of an object or scene onto a plane (i.e. screen or image sensor).

**Image Processing (IP)** — Transformation of an input image into an output image with desired properties.

**Image Resolution** — The number of rows and columns of pixels in an image. A higher resolution means that more pixels are available per element of the symbol being read. Examples: VGA (640 x 480); WVGA (768 x 480); SXGA (1280 x 1024); QXGA (2048 x 1536).

**Image Sensor** — A device that converts a visual image to an electrical signal; a CCD or CMOS array.

**Initialize** — Implement serial configuration commands into the reader's active memory.

**Input** — A channel or communications line. Decoded data or a discrete signal that is received by a device.

**Integration** — Exposure of pixels on a CMOS sensor.

**Ladder Orientation** — A linear symbol orientation in which the bars are parallel to the symbol's direction of travel.

**Light-Emitting Diode (LED)** — A semiconductor device that emits light when conducting current.

**Lens** — A transparent piece of material with curved surfaces which either converge or diverge light rays.

**Noise** — The same as static in a phone line or “snow” in a television picture, noise is any unwanted electrical signal that interferes with the image being read and transferred by the reader.

**Normally Closed** — A discrete output state that is only active when open.

**Normally Open** — A discrete output state that is only active when closed.

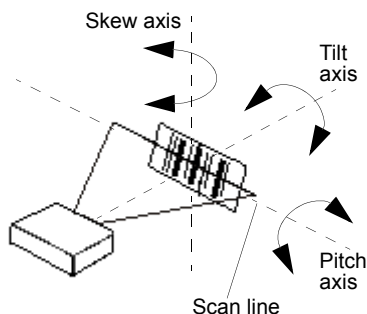
**Object Plane** — An imaginary plane in the field of view, focused by an reader’s optical system at the corresponding image plane on the sensor.

**Output** — A channel or communications line. Data or discrete signals that are transmitted or displayed by a device.

**Parity** — An error detection routine in which one data bit in each character is set to 1 or 0 so that the total number of bits in the data field is even or odd.

**Picket Fence Orientation** — A linear symbol orientation in which the bars are perpendicular to the symbol’s direction of travel.

**Pitch** — Rotation of a linear or 2D symbol around an axis parallel to the symbol length on the substrate. See the illustration below.



**Pixel** — An individual element in a digitized image array. Derived from the phrase “picture element”.

**Port** — Logical circuit for data entry and exit. (One or more ports may be included within a single connector.)

**Processing Time** — The time used by a vision system to receive, analyze and interpret image information. Often expressed in parts per minute.

**Programmable Logic Controller (PLC)** — An electronic device used in industrial automation environments such as factory assembly lines and automotive manufacturing facilities.

**Progressive Scan** — A non-interlaced scan that doubles the number of visible picture lines per field by displaying all picture lines at once.

**Protocol** — The rules for communication between devices, providing a means to control the orderly flow of information between linked devices.

**Random-Access Memory (RAM)** — A data storage system used in computers, composed of integrated circuits that allow access to stored data in any sequence without movement of physical parts.

---

## *Glossary of Terms*

**Read Cycle** — A programmed period of time or condition during which a reader will accept symbol input.

**Read-Only Memory (ROM)** — A data storage medium used in computers and other electronics, primarily used to distribute firmware.

**Read Rate** — The number of good reads per second achieved by a reader.

**Real-Time Processing** — In machine vision, the ability of a system to perform a complete analysis and take action on one part before the next one arrives for inspection.

**Region** — Area of an image. Also called a region of interest for image processing operations.

**Saturation** — The degree to which a color is free of white. One of the three properties of color perception, along with hue and value.

**Scattering** — Redirection of light reflecting off a surface or through an object.

**Skew** — Rotation of a linear or 2D symbol around an axis parallel to the symbol height on the substrate. See the illustration under the definition of pitch.

**Substrate** — The surface upon which a linear or 2D symbol is printed, stamped, or etched.

**Symbology** — A symbol type, such as Code 39 or Code 128, with special rules to define the widths and positions of bars and spaces to represent specific numeric or alphanumeric information.

**Symbol Transitions** — The transition of bars and spaces on a symbol, used to detect the presence of a symbol on an object.

**Tilt** — Rotation of a linear or 2D symbol around an axis perpendicular to the substrate. See the illustration under the definition of pitch.

**Trigger** — A signal, transition, or character string that initiates a read cycle.

**Very Large-Scale Integration (VLSI)** — The creation of integrated circuits by combining thousands of transistor-based circuits on a single chip.